A Hybrid Machine Translation Framework for an Improved Translation Workflow

Dissertation

zur Erlangung des akademischen Grades eines

Doktors der Philosophie

der Philosophischen Fakultäten

der Universität des Saarlandes

vorgelegt von

Santanu Pal

aus

Sonamukhi, West Bengal, Indien

Saarbrücken, 2018

Der Dekan: Prof. Dr. Roland Marti

Erstberichterstatter: Prof. Dr. Josef van Genabith

Zweitberichterstatter: Prof. Dr. Dietrich Klakow

Tag der letzten Prüfungsleistung: 27.11.2017

"A scientist in his laboratory is not a mere technician: he is also a child confronting natural phenomena that impress him as though they were fairy tales ...."

Marie Curie

Abstract

A Hybrid Machine Translation Framework for an Improved Translation Workflow

by Santanu Pal

Doctor of Philosophy

Computerlinguistik, Sprachwissenschaft und Sprachtechnologie

Universität des Saarlandes

Over the past few decades, due to a continuing surge in the amount of content being translated and ever increasing pressure to deliver high quality and high throughput translation, translation industries are focusing their interest on adopting advanced technologies such as machine translation (MT), and automatic post-editing (APE) in their translation workflows. Despite the progress of the technology, the roles of humans and machines essentially remain intact as MT/APE are moving from the peripheries of the translation field closer towards collaborative human-machine based MT/APE in modern translation workflows. Professional translators increasingly become post-editors correcting raw MT/APE output instead of translating from scratch which in turn increases productivity in terms of translation speed. The last decade has seen substantial growth in research and development activities on improving MT; usually concentrating on selected aspects of workflows starting from training data pre-processing techniques to core MT processes to post-editing methods. To date, however, complete MT workflows are less investigated than the core MT processes. In the research presented in this thesis, we investigate avenues towards achieving improved MT workflows. We study how different MT paradigms can be utilized and integrated to best effect. We also investigate how different upstream and downstream component technologies can be hybridized to achieve overall improved MT. Finally we include an investigation into human-machine collaborative MT by taking humans in the loop. In many of (but not all) the experiments presented in this thesis we focus on data scenarios provided by low resource language settings.

German Summary

(Zusammenfassung)

Aufgrund des stetig ansteigenden Übersetzungsvolumens in den letzten Jahrzehnten und gleichzeitig wachsendem Druck hohe Qualität innerhalb von kürzester Zeit liefern zu müssen sind Übersetzungsdienstleister darauf angewiesen, moderne Technologien wie Maschinelle Übersetzung (MT) und automatisches Post-Editing (APE) in den Übersetzungsworkflow einzubinden. Trotz erheblicher Fortschritte dieser Technologien haben sich die Rollen von Mensch und Maschine kaum verändert. MT/APE ist jedoch nunmehr nicht mehr nur eine Randerscheinung, sondern wird im modernen Übersetzungsworkflow zunehmend in Zusammenarbeit von Mensch und Maschine eingesetzt. Fachübersetzer werden immer mehr zu Post-Editoren und korrigieren den MT/APE-Output, statt wie bisher Übersetzungen komplett neu anzufertigen. So kann die Produktivität bezüglich der Übersetzungsgeschwindigkeit gesteigert werden. Im letzten Jahrzehnt hat sich in den Bereichen Forschung und Entwicklung zur Verbesserung von MT sehr viel getan: Einbindung des vollständigen Übersetzungsworkflows von der Vorbereitung der Trainingsdaten über den eigentlichen MT-Prozess bis hin zu Post-Editing-Methoden. Der vollständige Übersetzungsworkflow wird jedoch aus Datenperspektive weit weniger berücksichtigt als der eigentliche MT-Prozess. In dieser Dissertation werden Wege hin zum idealen oder zumindest verbesserten MT-Workflow untersucht. In den Experimenten wird dabei besondere Aufmertsamfit auf die speziellen Belange von sprachen mit geringen ressourcen gelegt. Es wird untersucht wie unterschiedliche MT-Paradigmen verwendet und optimal integriert werden können. Des Weiteren wird dargestellt wie unterschiedliche vor- und nachgelagerte Technologiekomponenten angepasst werden können, um insgesamt einen besseren MT-Output zu generieren. Abschließend wird gezeigt wie der Mensch in den MT-Workflow intergriert werden kann. Das Ziel dieser Arbeit ist es verschiedene Technologiekomponenten in den MT-Workflow zu integrieren um so einen verbesserten Gesamtworkflow zu schaffen. Hierfür werden hauptsächlich Hybridisierungsansätze verwendet. In dieser Arbeit werden außerdem Möglichkeiten untersucht, Menschen effektiv als Post-Editoren einzubinden. Die hierbei gewonnenen Übersetzungsprozessdaten

werden automatisch gesammelt und stehen für künftige Forschung zur Verfügung (z.B. zur Unterstützung von inkrementellen Updates einzelner Workflow- und Post-Editing-Komponenten). Das Hauptziel dieser Dissertation ist es, die echten Bedürfnisse und Probleme der Anwender von Übersetzungstechnologie - einschließlich von professionellen übersetzern - zu erfassen. Es soll ein kollaborativer Rahmen für hybride maschinelle Übersetzung geschaffen werden, der den Übersetzungsworkflow verbessert und den Post-Editing-Effort für den Übersetzer reduziert. Auf dieser Grundlage soll die Funktionalität von Übersetzungssystemen in Hinblick auf die Anforderungen der Anwender optimiert werden, statt die Anwender dazu zu zwingen, ihre Arbeitsweise an die Technologie anzupassen. Des Weiteren untersucht diese Arbeit ob und wie bestehende Technologien wie neuronale MT (NMT), statistische MT (SMT), beispielbasierte MT (Example Based MT, EBMT) und Translation- Memory-Systeme (TM) die Anforderungen der Anwender unterstützen können.

Hierfür wird der in der arbeit verfolgte Ansatz auf zwei Arten beschrieben. Normalerweise werden verschiedene Technologiekomponenten im Übersetzungsworkflow kombiniert; im ersten Teil dieser Arbeit liegt der Schwerpunkt auf den (i) Komponenten, im zweiten Teil auf den (ii) Workflows. Im ersten Teil dieser Arbeit liegt der Schwerpunkt insbesondere auf Design und Implementierung von leistungsstarken und benutzerfreundlichen Technologiekomponenten (beschrieben in Kapitel 3, Kapitel 4 und Kapitel 5). In diesem Teil werden diverse Hybridisierungsansätze angewandt. Im zweiten Teil der Arbeit liegt der Schwerpunkt auf der Identifizierung optimierter Workflows durch die Kombination verschiedener Technologiekomponenten. Hier wird eine Plug&Play-Methodologie angewendet, bei der der Optimierungsgrad in Bezug auf besseren Übersetzungsoutput gemessen wird. Des Weiteren wird in diesem Teil der Arbeit auch der Mensch in den Übersetzungsprozess zur Bewertung der Übersetzungsqualität sowie zur langfristigen schrittweisen Verbesserung der Technologiekomponenten durch Feedback eingebunden. Zusätzlich werden gleichzeitig wertvolle Ressourcen für Übersetzungsprozessforschung geschaffen (vgl. Kapitel 6). In Kapitel 6 liegt der Fokus auf Technologien der die Sammlung von Ressourcen für den vorgeschlagenen Rahmen zur schrittweisen Verbesserung von MT/APE-Komponenten für der Workflow sowie für die Übersetzungsprozessforschung

unterstützen. Nachdem Ressourcen in beträchtlichem Umfang durch den kollaborativen Rahmen für hybride human-maschinelle Übersetzung geschaffen wurden, soll diese Forschungsarbeit künftig durch die Einbeziehung weiterer Komponenten fortgeführt werden. APE stellt jedoch auch eine beträchtliche Verbesserung unseres übersetzungsrahmen dar.

MT ist per Definition ein computergestützter Prozess, der Text einer menschlichen Sprache in eine andere umwandelt. Dies geschieht entweder voll- oder halbautomatisch (dies ist der Fall bei von Menschen unterstützen MT, bei der der Mensch in den Übersetzungsprozess eingebunden wird). In den letzten Jahren wurden verschiedene Ansätze zur MT untersucht, z.B. regelbasierte, beispielbasierte, wissensbasierte, statistische und neuronale Ansätze. Hinsichtlich weitreichender Entwicklungen und Anwendungen war SMT und insbesondere phrasenbasierte SMT (phrase-based SMT, PB-SMT) aus all diesen Ansätzen bis vor kurzem weitgehende dominierend[1]. Die Qualität von PB-SMT hängt sehr stark von vorgelagerten Prozessen wie Wordalignment und Bewertung von Phrasenpaaren ab. Beides kann durch die Verwendung großer satzalignierter Parallelkorpora erreicht werden. Jedoch kann die Verfügbarkeit von Daten eine Herausforderung sein. PB-SMT für Sprachpaare mit knappen Datenressourcen liefert schlechtere Übersetzungsqualität, da nicht genügend Trainingsdaten aus Parallelkorpora verfübar sind. Hieraus resultiert die erste Forschungsfrage (RQ).

RQ1: *Wie kann MT für Sprachen mit geringen Ressourcen verbessert werden?*

In Kapitel 3 wird eine mögliche Lösung für das Problem der Datenknappheit dargestellt. In diesem Kapitel wird eine Methodologie zur Extraktion paralleler Textfragmente aus vergleichbaren Korpora beschrieben, die als zusätzliche Trainingsdaten in der SMT für das Sprachenpaar Englisch-Bengali genutzt werden können. Für dieses Sprachenpaar stehen nur geringe Ressourcen zur Verfügung. Zur Extraktion von Paralleltexten aus vergleichbaren Korpora wird im Rahmen dieser Arbeit Textual Entailment Techniken (TE) angewendet. Der wichtigste Teil dieser Forschung, der in diesem Kapitel vorgestellt wird, wurde auch in (Pal et al., 2014b, 2015b) veröffentlicht.

[1]WMT 2016 war die erste große Shared Task, bei der NMT besser abschnitt als SMT-basierte Ansätze. Diese Entwicklung wird in Kapitel 5 zu neuronaler APE betrachtet.

Um bereits bestehende zweisprachige Daten bestens zu nutzen, ist außerdem die Datenvorverarbeitung entscheidend. Wie schon erwähnt, baut SMT sehr stark auf gute Qualität von Wort- und Phrasenalignment als Darstellung von Übersetzungswissen durch SMT-Systemen aus einem bilingualen Korpus. Eine der Kernkomponenten der SMT ist das statistische Wortalignment, oft basierend auf IBM-Modellen (Brown et al., 1993). Diese IBM-Modelle können jedoch schlecht mit komplexen Ausdrücken (z.B. Multi-Word Expressions (MWE), Named Entities (NE)) umgehen, da diese Modelle viele-zu-viele Alignments nicht ohne weiteres abdecken können. Des Weiteren ist unterschiedliche Wortstellung in verschiedenen Sprachen ein bekanntes Phänomen und stellt für MT (insbesondere SMT) eine besondere Herausforderung dar. Hieraus resultiert die zweite Forschungsfrage:

RQ2: *Wie kann SMT bereits bestehende Trainingsdaten besser nutzen?*

In Kapitel 4 wird das Problem der Alignierung von vielen-zu-vielen Ausdrücken behandelt und beschrieben wie mit unterschiedlicher Wortstellung in weit entfernten Sprachpaaren effizient umgegangen werden kann. Es wird außerdem eine Hybridmethode zur Kombination verschiedener Wordalignments unterschiedlicher Wordaligners vorgestellt. Diese Methode wird anschließend in den Rahmen eines hybriden Multienginesystems zur Verbesserung der MT-Performance eingebunden.

In Kapitel 4 sollen Modellfehler reduziert werden (z.B. weist das Modell nicht den höchsten Score dem besten Übersetzungskandidaten zu). Dies geschieht durch (i) die systematische Kombination verschiedener MT-Komponenten, (ii) die Kombination verschiedener Systeme und (iii) die Neubewertung des Outputs verschiedener MT-Systeme. Hieraus resultiert die dritte Forschungsfrage:

RQ3: *Wie könnte eine verbesserte hybride MT-Implementierung aussehen?*

In Kapitel 4 und (Pal et al., 2014c) wird ein hybrides SMT-System beschrieben, mit dem die Baseline-SMT-Performance verbessert wird. Dies erfolgt durch die Einbindung zusätzlicher Wissensquellen wie beispielsweise extrahierter zweisprachige Named Entities, Translation Memories und Phasenpaare, die durch beispielbasierte Methoden und

Standard-SMT-Ressourcen gewonnen wurden. Die Performance verschiedener hybrider Systeme und auch Ergebnisse einer Systemkombination basierend auf einem Confusion Network, bei dem die beste Performance jedes einzelnen Systems in der Multi-Engine Pipeline kombiniert wird, werden beschrieben. Wichtige Teile der in diesem Kapitel vorgestellten Forschungsarbeit wurden in (Pal et al., 2014c,a, 2015a, 2016a) publiziert.

MT-Systeme haben das Fernziel vollautomatisch eine Übersetzung zu generieren, die ohne Nachbearbeitung veröffentlicht werden kann. Bestehende MT-Systeme erreichen dieses Qualitätsziel jedoch meist nicht, so dass der rohe MT-Output von Menschen post-editiert werden muss (vgl. Abbildung 1). Zur Verbesserung der Übersetzungsqualität ohne eine Veränderung des ursprünglichen MT-Systems selbst kann ein zusätzliches Plug-in-Modul zur Nachbearbeitung verwendet werden. Dies kann beispielsweise ein nachgelagertes monolinguales MT-System wie ein automatisches Post-Editing-System (APE) sein, dass auf Output des eigentlichen MT-Systems und den vom Menschen durchgeführten Korrekturen trainiert wurde (cf. Abbildung 2). So kann eine vernünftigere und machbare Lösung erreicht werden, ohne das gesamte eigentliche MT-System zu erneuern. Somit kann eine vierte Forschungsfrage gestellt und untersucht werden:

RQ4: *Wie kann ein effektives automatisches Post-Editing-System erstellt werden, dass die Übersetzungsqualität des eigentlichen MT-Systems verbessert?*

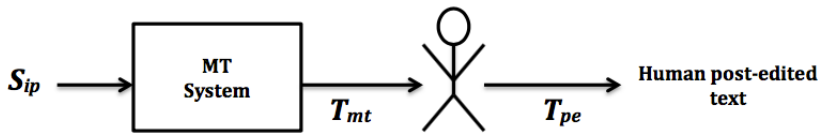


FIGURE 1: Post-editing on MT; S_{ip}: source texts, T_{mt}: corresponding MT output texts and T_{pe}: the human post-edited version of T_{mt}.

Der Vorteil von APE liegt in seiner Anpassungsfähigkeit an jegliche black-box MT-Engine; d.h. sind post-editierte Daten verfügbar, so ist kein inkrementelles oder volles Training des eigentlichen MT-Systems, das bei der Sammlung der Post-Editing-Daten verwendet wurde, notwendig. APE setzt Verfügbarkeit der Quelltexte voraus (S_{ip}), sowie den

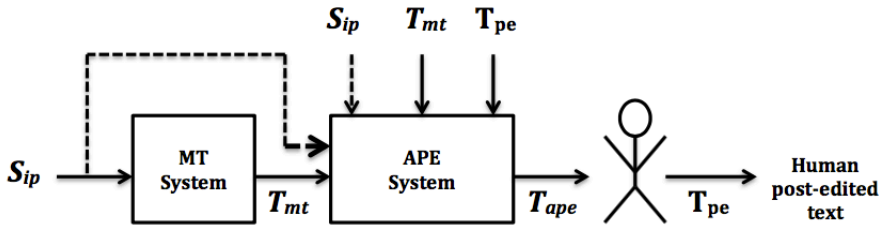


FIGURE 2: Automatic Post-editing; here "- - →" means S_{ip} may be or may not be included in the APE system between $S_{ip}\_T_{mt}$–T_{pe}. Our APE system currently uses only T_{mt}–T_{pe}.

entsprechenden MT-Output der Texte (T_{mt}) und die vom Menschen post-editierte Version (T_{pe}) of T_{mt}. APE-Systeme können als MT-Systeme zwischen $S_{ip}\_T_{mt}$ und T_{pe} modelliert werden. Statistische APE-Systeme (SAPE) oder neuronale APE-Systeme (NNAPE) können auch ohne S_{ip} gebaut werden. Hierfür werden genügend "monolinguale" Paralleltexte der Zielsprache T_{mt}–T_{pe} verwendet. APE-Tasks konzentrieren sich gewöhnlich auf systematische Fehler des MT-Systems – meist Lexik- oder Wortstellungsfehler sowie falsches Löschen oder Hinzufügen eines Wortes. Hieraus resultiert die fünfte Forschungsfrage.

RQ5: *In wie weit ist ein APE-System in der Lage, den letztlichen Post-Editing-Effort zu verringern und die Produktivität zu steigern?*

Zur Beantwortung von Forschungsfragen RQ4 und RQ5 wird in Kapitel 5 und (Pal et al., 2016c) ein APE-System auf Grundlage von neuronalen Netzen zur Verbesserung des rohen MT-Outputs vorgestellt. Das neuronale APE-Modell (NNAPE) basiert auf einem bidirektionalen rekurrenten neuronalen Netz (RNN) und besteht aus einem Encoder, der MT-Output in einen Vektor mit festgelegter Länge kodiert. Ein Decoder nutzt den Vektor zur Erstellung der post-editierten Übersetzung (PE).

In Kapitel 5 werden zudem zwei Stränge der MT-Forschung kombiniert: MT basiert auf APE (statistsch (Pal et al., 2016f) und neuronalen Netzen (Pal et al., 2016c)) sowie Multi-Engines (Systemkombinationen) (Pal et al., 2016b). APE-Systeme nutzen ein nachgelagertes MT-System, das auf Zielsprachenseite durch die Korrekturen des Human-Post-Editors lernt und verbessern so den Output des eigentlichen MT-Systems. Dies ist eigentlich eine *sequentielle* MT-Systemarchitektur. Gleichzeitig gibt es außerdem sehr

viel Literatur zu *parallelen* MT-Systemkombinationen, bei denen der gleiche Input in verschiedene Engines gegeben wird. Der beste Output wird dann ausgewählt oder es werden kleinere Teile verschiedener Outputs kombiniert um einen besseren Übersetzungsoutput zu erhalten. In einem Experiment werden sequentielle und parallele Systemkombinationen integriert und somit eine signifikante Produktivitätssteigerung bezüglich Post-Editing durch professionelle Übersetzer erreicht. Wichtige Teile der in diesem Kapitel vorgestellten Forschungsarbeit wurden in (Pal et al., 2015c, 2016b,c,f) publiziert.

Die Übersetzungen von MT/APE-Systemen müssen oft noch durch menschliche Übersetzer korrigiert werden, so dass sie veröffentlicht werden können. Fallstudien (O'Brien et al., 2009; TAUS Report, 2010) zeigen, dass der Einsatz einer MT/APE-Engine für alle Seiten bezüglich Kosten- und Zeiteinsparnis von Nutzen sein kann: für Kunden, Übersetzer und Language Service Providers (LSP). Das Gesamtbild bleibt jedoch weiterhin gemischt: MT/APE ist einerseits oft günstig und einfach anwendbar; andererseits ist die Übersetzungsqualität oft nicht zufriedenstellend. Einige professionelle Übersetzer übersetzen lieber von Grund auf selbst. Somit stellt sich die folgende Forschungsfrage:

RQ6: *Wie können bestehende MT-Workflows bei der Arbeit des Menschen mit CAT-Tools optimiert werden?*

In Kapitel 6 wird ein neues webbasiertes Post-Editing Tool vorgestellt: *CATaLog Online* wurde mit einigen neuen Features verbessert. Das Tool kann als reines CAT-Tool genauso verwendet werden wie für das Post-Editing von TM-Segmenten oder von MT-Output. Das Tool erfasst umfassende Informationen im Aktivitätslog, eine Funktion, die die meisten CAT-Tools nicht bieten. Das Tool ist für die Übersetzungsprozessforschung ausgelegt und bietet die folgenden Vorteile: (i) farblich markierte TM-Übersetzungsvorschläge (die markierte Quelle TM sowie das entsprechende Zielfragment werden auf der selben Oberfläche angezeigt), (ii) eine Vielzahl an Editinglogs, (iii) Alignment von Quelle, TM/MT/APE-Output mit dem Ergebnis von humanem PE, (iv) eine verbesserte TM-ähnlichkeitsmessung und Suche (Pal et al., 2016e) und (v) die zusätzliche Übersetzungsoption APE. Wichtige Teile der in diesem Kapitel vorgestellten Forschungsarbeit wurden in (Nayek et al., 2015; Nayak et al., 2016; Pal et al., 2016e,d) publiziert.

Acknowledgements

I express my sincere gratitude to my advisor **Prof. Josef van Genabith**, for the continuous support of my Ph.D study and related research, for his patience, motivation, and immense knowledge. I could not have imagined having a better advisor and mentor for my Ph.D study.

Most of the work reported in this thesis would be impossible without the collaboration and guidance of my co-advisor **Dr. Sudip Kumar Naskar** for his affectionate and valuable guidance; without his help, the present work could not have been successful. His valuable suggestions and thoughts were very fruitful for shaping my ideas in research. He provided much encouragement and moral support especially during the final year of my doctoral thesis.

Besides my advisors, I would like to thank the rest of my thesis committee: Prof. Ingo Reich, Prof. Dietrich Klakow, Prof. Erich Steiner, and Dr. Cristina España-Bonet, for their insightful comments and encouragement.

My sincere thanks also go to other researchers and colleagues including Prof. Elke Teich, Liling Tan, Marcos Zampieri, Mihaela Vela, Jon Dehdari, Raphael Rubino, Katrin Manzel, Jörg Knappen, Daniele Moretti, José Manuel Martínez, Ekaterina Lapshinova-Koltunski, Stefania Degaetano-Ortlieb, Marc Summkeller, Peggy Daut, Ulrike Konz, Anne Weber, Andrea Wurm and Stefan Fischer who provided me with opportunities to do collaborative research.

I thank my fellow EXPERT labmates, my friends in the following institutions: Jadavpur University (India), Dublin City University (Ireland) and other EXPERT's partner universities, and industry partners: Pangeanic (Spain) and Translated SRL (Italy). In particular, I would like to thank the People Programme (Marie Curie Actions) of the European Union's Framework Programme (FP7/2007-2013) under REA grant agreement no. 317471 for providing the financial support to undertake the research described in this thesis.

I would also like to thank Heike Przybyl who helped me writing the German thesis summary and Gareth Dwyer who proofread this thesis as a native speaker.

Last but not the least, the encouragement given by my grandmother Puspa Rani Dutta, my grandfather Late Nabani Kumar Dutta, my mother Alpana Pal, my father Amar Nath Pal, my maternal uncle Swarup Dutta, Arup Kumar Dutta and Ashish Karmakar who have always been a constant source of inspiration and whose encouragement is beyond linguistic expression for me.

Finally, my deepest gratitude goes to my wife Sandipta Pal for supporting me spiritually throughout writing this thesis and my life in general. Without her continuous support and immense patience, it would have been impossible for me to complete this work.

Contents

List of Figures

List of Tables

Abbreviations

APE	**A**utomatic **P**ost-**E**diting
BLEU	**Bi**-**L**ingual **E**valuation **U**nderstudy
CAT	**C**omputer **A**ided **T**ranslation
CN	**C**onfusion **N**etwork
CBMT	**C**orpus **B**ased **M**achine **T**ranslation
DNN	**D**eep **N**eural **N**etwork
EBMT	**E**xample **B**ased **M**achine **T**ranslation
FSBSMT	**F**orest to **S**tring **B**ased **S**tatistical **M**achine **T**ranslation
HPE	**H**uman **P**ost-**E**diting
HWA	**H**ybrid **W**ord **A**lignment
MT	**M**achine **T**ranslation
METEOR	**M**etric for **E**valuation of **T**ranslation with **E**xplicit **OR**dering
MERT	**M**inimum **E**rror **R**ate **T**raining
MWE	**M**ulti-**W**ord **E**xpressions
NE	**N**amed **E**ntities
NMT	**N**eural **M**achine **T**ranslation
NN	**N**eural **N**etwork
NNAPE	**N**eural **N**etwork based **A**utomatic **P**ost-**E**diting
OSM	**O**peration **S**equence **M**odel
PB-SAPE	**P**hrase **B**ased **S**tatistical **A**utomatic **P**ost-**E**diting
PB-SMT	**P**hrase **B**ased **S**tatistical **M**achine **T**ranslation
PE	**P**ost-**E**diting
RNN	**R**ecurrent **N**eural **N**etwork
RBMT	**R**ule **B**ased **M**achine **T**ranslation
STS	**S**emantic **T**extual **S**imilarity

SL	**S**ource **L**anguage
SAPE	**S**tatistical **A**utomatic **P**ost-**E**diting
SMT	**S**tatistical **M**achine **T**ranslation
SC-APE	**S**ystem **C**ombination based **A**utomatic **P**ost-**E**diting
TL	**T**arget **L**anguage
TE	**T**extual **E**ntailment
TM	**T**ranslation **M**emory
TPR	**T**ranslation **P**rocess **R**esearch
TER	**T**ranslation Edit Rate

To my Family...

Chapter 1

Introduction

The work presented in this thesis aims at improving and integrating different component technologies of Machine Translation (MT) in an ideal or at least improved workflow based on hybridization approaches. The thesis also investigates effective ways of involving humans as post-editors and supporting this by automatically collecting translation process data for future research (e.g., to support incremental updates of individual workflow and post-editing components). The main aim of the research presented in this thesis is to examine real-life needs and problems which confront translation technology users, including professional translators, and provide a collaborative hybrid machine translation framework towards an improved translation workflow, and reduce post-editing effort of the translators. Based on this, the translation system functionality should be optimized in terms of user requirements rather than forcing the users to change how they work with the technology. The research also investigates whether and how combinations of existing technologies such as Neural MT (NMT), Statistical MT (SMT), Example Based MT (EBMT) and Translation Memory (TM) systems can be used and integrated to support the users' requirements. Throughout, in this thesis has a special focus on the data settings provided by low resource languages.

In order to achieve our objective, we describe our approach in two complementary ways. Technology components are usually combined into translation workflows; part of our research will therefore concentrate on the (i) components, while the other part of our research will focus on (ii) workflows. Specifically, in the first part of our research, we focus on the design and implementation of component technologies (described in Chapter

3, Chapter 4 and Chapter 5) that are effective and user friendly. Here we make extensive use of hybridization approaches. In the second part of our research, we concentrate on identifying better workflows based on a combination of different components of the component technologies in the workflow using a plug and play methodology, where better is initially measured in terms of translation output quality. Furthermore, our work also involves humans in the translation process for the purpose of assessing translation quality and as a long term goal for providing activity-log based feedback to incrementally enhance the different components of the component technology on the one hand and simultaneously generating valuable resources for translation process research on the other hand (cf., Chapter 6).

By definition, MT is a computer based process which transforms text in one human language text into another in either a fully automatic or a semi-automatic (human assisted MT, i.e., involving a human in the translation process) manner. Many approaches to MT have been explored, for instance rule-based, example-based, knowledge-based, statistical and neural approaches to MT. Out of these, in terms of large-scale evaluations and use, SMT in particular phrase-based SMT (PB-SMT) has until recently been the most successful MT paradigm[1]. The quality of PB-SMT mainly relies on good quality upstream word alignment as well as good phrase pair estimation, both of which can be achieved by using large amounts of sentence aligned parallel corpora. However, data scarcity can be a challenge and PB-SMT for low resource language pairs usually produces inferior quality translations due to insufficient amounts of parallel training data. This leads us to our first research question (RQ).

RQ1: *How can MT for low resource languages be improved?*

Comparable corpora provide a possible solution to this data scarcity problem to some extent. Comparable documents are not strictly parallel. Comparable corpora consist of bilingual documents. However these documents are not sentence by sentence translations; but they are on the same topic and convey similar information and hence it is likely that there exists some sentential or sub-sentential level of parallelism. Recently, comparable corpora are being considered as a valuable resource for acquiring parallel data, which can play an important role in improving the quality of SMT (Smith et al., 2010). The parallel

[1]WMT 2016 has been the first large scale shared task in which NMT has outperformed SMT based approaches and we reflect these developments in our research presented in Chapter 5 on neural APE.

segments extracted from comparable corpora are typically added to an existing training corpus as additional training material which is expected to improve performance of SMT systems, specifically for low-resource language pairs (e.g. English–Bengali). However, large fully parallel fragments of text are rarely found in comparable document pairs. The bigger the size of the fragment, the less probable it is that its parallel version will be found in the target. Nevertheless, there is always a chance of obtaining parallel phrases, tokens or even sentences in comparable documents. The challenge is to discover those parallel fragments which can be useful in increasing SMT performance. In Chapter 3, we describe a methodology for extracting English–Bengali parallel text fragments from comparable corpora. To extract parallel text from comparable corpora, in our research we apply textual entailment (TE) and then utilize the additional training data in SMT. In Pal et al. (2015b) we show that the additional training data extracted from comparable corpora provides significant improvements of 3.06 absolute points and 32.97% relative over the baseline PB-SMT system as measured by BLEU (Papineni et al., 2002) for English–Bengali translations.

Furthermore, to obtain optimal benefits from the existing bilingual data, data preprocessing plays a crucial role. As mentioned earlier, SMT relies heavily on the quality of word alignment and phrase alignment which essentially represent the translation knowledge acquired by an SMT system from a bilingual corpus. One of the core components of SMT is statistical word alignment which is often based on IBM models (Brown et al., 1993). These IBM Models do not work well with complex expressions (e.g., multi-word expressions (MWEs), Named Entities (NEs)), due to their inability to handle many-to-many alignments. The IBM Models only allow one-to-many alignments from source language to target language (Koehn et al., 2003; Marcu, 2001)[2]. In another well-known statistical word alignment approach, Hidden Markov Models (HMM: Vogel et al. (1996)), the alignment probabilities depend on the alignment position of the previous word. HMM alignment does not explicitly consider many-to-many alignments either. Furthermore, reordering is a well-known cross-lingual phenomena which poses a significant challenge in MT (especially in SMT). This leads us to the second research question:

RQ2: *How can SMT better profit from the existing training data?*

[2]Alignment in both directions (i.e., alignment symmetrization) are used to partially address this problem.

In Chapter 4, we address this many-to-many alignment problem indirectly. Our objective focuses on how to best handle the NEs and MWEs in SMT. In our research, MWEs, NEs and compound verbs are automatically identified on both sides of the parallel corpus. Then, source and target language NEs are aligned using a statistical transliteration method. We rely on these automatically aligned NEs and treat them as translation examples. We modify the parallel corpus by converting the MWEs into single tokens and adding the aligned NEs to the parallel corpus in a bid to improve the word alignment, and hence the phrase alignment, quality. This preprocessing results in improved MT quality in terms of automatic MT evaluation as we show in Pal et al. (2013b) for English–Bengali, Tan and Pal (2014) for English–Hindi, and Pal et al. (2015a) for English–German. Chapter 4 also presents how reordering between distant language pairs can be handled efficiently in phrase-based SMT. The problem of reordering between distant languages has been approached with prior reordering of the source text at chunk level to simulate the target language ordering (Pal et al., 2014a). In this chapter, we report experiments on prior reordering of the source chunks following the target word order suggested by word alignment. We reorder the test set using a monolingual MT trained on source and reordered source. Our approach of prior reordering of the source chunks is compared with pre-ordering of source words based on word alignments (Holmqvist et al., 2012) and the traditional approach of prior source reordering based on language-pair specific reordering rules. The effects of these reordering approaches is studied on an English–Bengali translation task, a language pair with substantially different word order. Our experimental results show that in our data setting word alignment based reordering of the source chunks is more effective than the other reordering approaches and that it produces statistically significant improvements over the baseline system on BLEU (Pal et al., 2014a). Manual inspection confirms significant improvements in terms of word alignments.

Over the last 10-15 years, MT has made considerable progress, in large part due to research and development in statistical and hybrid (and recently neural) approaches to MT. In many cases, MT services provide a convenient support not only for professional translators but also for general users. Free online MT engines and commercial engines are available, such as Google Translator[3], Systran[4], Microsoft Bing translator[5], etc. Furthermore, commercial as well as free version of translation support tools (well-known as

[3] `https://translate.google.com/`
[4] `http://www.systransoft.com/`
[5] `https://www.bing.com/translator`

Computer Aided Translation – CAT tools) are also available such as MateCat[6], CasmaCat[7], Trados[8] Translation tools and are widely used in the translation industry. The core component of the vast majority of CAT tools are translation memories (TM). TMs work under the assumption that previously translated segments can serve as good models for new translations, especially when translating technical or domain specific texts, where some amount of repetition exists. Translators input new texts to be translated into the CAT tool and these texts are divided into shorter segments. The TM engine then checks whether there are segments (and their translation) in the memory which are similar to those from the input text. Every time the software finds a similar segment in the memory, the tool shows it together with its translation to the translator as a suitable suggestion usually through a graphical interface. In this scenario, translators work as post-editors by adapting retrieved segments suggested by the CAT tool or in case no suitable segments are found translating new segments from scratch. This process is done iteratively and every post-edited segment or new translation increases the size of the translation memory making it more useful for future translations.

TM systems store source and target language translation pairs for effectively reusing the previous translations originally created by human translators. Conceptually, EBMT is closely related to TM. The difference between the two approaches is that EBMT extracts translations of fragments of an input sentence to be translated from the translation model and combines fragments to produce translations for a segment in question whereas TMs are not translation systems as such but rather act like search engines which provide closely matching translation pairs for a complete segment to effectively reduce the translation workload of translators.

Despite continued and significant progress, fully automatic MT is often not yet able to always provide desirable performance in terms of output quality. Each approach to MT has its own method of acquiring and using translation knowledge from the parallel bilingual translation examples, along with its own advantages and disadvantages. The knowledge representation processes in both EBMT and SMT use very different techniques in order to extract translation knowledge. SMT phrases are n–grams (i.e., contiguous sequence of words), rather than grammatical phrases (as in a grammatical theory such as noun

[6] `https://www.matecat.com/`
[7] `http://www.casmacat.eu/`
[8] `http://www.translationzone.com/products/trados-studio/` etc.

phrases (NPs), prepositional phrases (PPs) or Verb phrases (VPs)) as in EBMT. Many researchers have investigated combinations of different MT approaches (Hybrid MT) to achieve better performance. The decoding task for SMT models is an NP-hard (Knight, 1999) problem with exponential complexity of the search space which implies that a decoder can only perform a non-exhaustive search (using heuristic search methods) to find the best possible translation for a given input. This may lead to a number of system errors such as – **model error:** the model fails to assign the highest score to the best translation candidate, **search error:** the search process fails to find the best translation hypothesis in the search space and **induction error:** when the optimal translation is absent in the search space owing to various pruning strategies (Fancellu and Webber, 2015). Our research aims to reduce the model error (i) by systemically combining various MT components, (ii) by combining different systems and (iii) by re-ranking different MT systems' outputs. This leads to our third research question.

RQ3: *What could improved hybrid implementations of MT be like?*

Our Hybrid MT system described in Chapter 4 and in (Pal et al., 2014c) improves over the baseline SMT performance by incorporating additional knowledge sources such as the extracted bilingual named entities, translation memories, and phrase pairs induced from example-based methods together with the standard SMT resources. We report performance on different hybrid systems as well as results of a confusion network based system combination that combines the best performance of each individual system within a multi-engine pipeline. Our best system (Pal et al., 2014c) achieved an overall BLEU score of 24.61 averaged over all language pairs and all domains in the shared task on SMT in Indian languages which encompassed translating from five languages (Bengali, English, Marathi, Tamil and Telugu) into Hindi in three different domains (Health, Tourism and General).

We also performed a similar experiment on English–German (Pal et al., 2015a) using the WMT-2015[9] news-2015 test set, where our confusion-network-based system combination model outperforms all our individual MT systems. Our hybrid system achieved an improvement of 5.9 absolute BLEU points i.e., a 35.3% relative improvement over the English–German baseline PB-SMT system.

[9] `http://www.statmt.org/wmt15/translation-task.html`

In Chapter 4, we also present a similar hybrid technique which combines different word alignment methods and integrates them into a forest-to-string based SMT (FSBSMT) system (Pal et al., 2016a). We show that hybrid word alignment integrated into various experimental settings of FSBSMT provides considerable improvements over state-of-the-art Hierarchical Phrase based SMT (HPBSMT). The research also demonstrates that additional integration of Named Entities (NEs), their translations and Example Based Machine Translation (EBMT) phrases (all extracted from the bilingual parallel training data) into the system brings about further improvements over the hybrid FSBSMT system. Our best system achieves 78.5% relative (9.84 BLEU points absolute) improvement over the baseline HPBSMT on an English–Bengali data set.

The ultimate goal of MT systems is to provide fully automatic publishable quality translations. However, existing MT systems often fail to deliver publishable quality translation output requiring human post-editing of raw MT output. To achieve translations of sufficient quality, translations often need to be corrected or post-edited by human translators (cf. Figure 1.1). Nonetheless, translations produced by MT systems have improved substantially and consistently over the last two decades. Translations produced by MT systems are now widely used in the translation and localization industry. To enhance the quality of translation without changing the original MT system itself, an additional plug-in automatic post-processing module, e.g. a second stage monolingual MT system such as an automatic post-editing (APE) system trained on previous output of the first-stage MT system and its human corrections (PEs/HPEs), can be introduced (cf. Figure 1.2). This may lead to a more reasonable and feasible solution compared to rebuilding the entire existing first-stage MT system. This motivated us to pose and explore for our fourth research question.

RQ4: *How can we build an effective automatic post-editing system which can improve the translation quality of the first-stage MT system?*

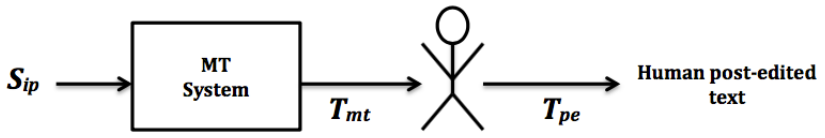


FIGURE 1.1: Post-editing on MT; S_{ip}: source texts, T_{mt}: corresponding MT output texts and T_{pe}: the human post-edited version of T_{mt}.

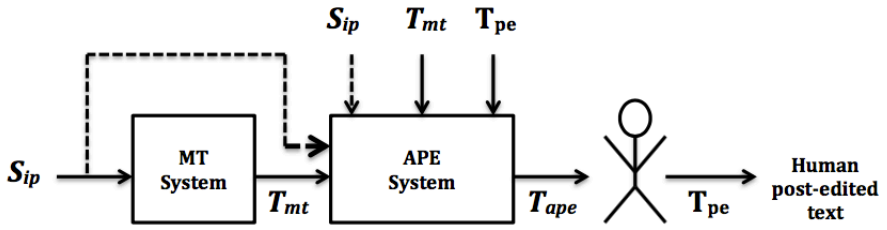


FIGURE 1.2: Automatic Post-editing; here "- - →" means S_{ip} may be or may not be included in the APE system between S_{ip}_T_{mt}–T_{pe}. Our APE system currently uses only T_{mt}–T_{pe}.

The term "Post-Editing" (PE) is defined as the corrections performed by humans over the translations produced by an MT system (Veale and Way, 1997). It is often understood as the process of improving a translation provided by an MT system with the minimum amount of manual effort (TAUS Report, 2010). While MT is often not perfect, post-editing MT output can yield productivity gains as it may require less effort than translating the same input manually from scratch. MT outputs are often post-edited by professional translators and the use of MT has become an important part of the translation workflow. A number of studies confirm that post-editing MT output can improve translators' performance in terms of productivity and it may positively impact on translation quality and consistency (Guerberof, 2009; Plitt and Masselot, 2010; Zampieri and Vela, 2014). The wide use of MT in modern translation workflows in the localization industry, in turn, has resulted in substantial quantities of human PE data (MT output and its human correction) which can be used to develop APE systems. APE (Knight and Chander, 1994) has been proposed as an automatic method for improving raw MT output, before (as it may not be a perfect) performing final human post-editing on it (cf. Figure 1.2). The approach is based on automatic corrections of errors made by the MT system. The automatic corrections of errors are learned from the human-corrected output of a first stage MT system possibly resulting in a productivity increase in the translation process. The advantage of APE relies on its capability to adapt to any black-box MT engine; i.e., upon availability of post-edited data, no incremental training or full re-training of the first stage MT system is required to improve the overall translation quality of the first stage MT system that was involved in the post-editing data collection. APE assumes the availability of source texts (S_{ip}), corresponding MT output texts (T_{mt}) and the human post-edited version (T_{pe}) of T_{mt}, and APE systems can be modelled as an MT system

between $S_{ip}\_T_{mt}$ and T_{pe}. However, statistical APE (SAPE) or neural APE (NNAPE) systems can also be built without the availability of S_{ip} using only sufficient amounts of target side "mono-lingual" parallel T_{mt}–T_{pe} text within the SMT framework. Usually APE tasks focus on systematic errors made by MT systems – the most frequent ones often being incorrect lexical choices, incorrect word ordering, and incorrect insertion or deletion of a word. This leads to our fifth research question.

RQ5: *To what extent is an APE system able to reduce final post-editing effort in terms of increasing productivity?*

To find answers to research questions RQ4 and RQ5, in Chapter 5 and (Pal et al., 2016c), we present a neural network based APE system to improve raw MT output quality. Our neural model of APE (NNAPE) is based on a bidirectional recurrent neural network (RNN) model and consists of an encoder that encodes an MT output into a fixed-length vector from which a decoder provides a post-edited (PE) translation. APE translations produced by NNAPE show statistically significant improvements of 3.96, 2.68 and 1.35 BLEU points absolute over the original English–Italian MT, phrase-based APE and hierarchical APE outputs, respectively. Furthermore, human evaluation shows that the NNAPE generated PE translations are much better than the original MT output.

In Chapter 5, we also investigate an APE method to improve the translation quality produced by an English–German SMT system (Pal et al., 2016f). We present an APE system based on the Operation Sequence Model (OSM) combined with a PB-SMT system. The system is trained on "monolingual" data consisting of MT output texts (TL_{mt}) produced by a black-box MT system and their corresponding human post-edited version (TL_{pe}). Our system achieves 64.10 BLEU (1.99 absolute points and 3.2% relative improvement in BLEU over raw MT output) and 24.14 TER (0.66 absolute points and 0.25% relative improvement in TER over raw MT output) on the official WMT 2016[10] APE test set.

Furthermore, in Chapter 5, we combine two strands of MT research: APE and multi-engine (system combination) MT (Pal et al., 2016b). APE systems learn a target-language-side second stage MT system from the data produced by human corrected output of a first stage MT system, to effectively improve the output of the first stage MT in what is essentially a *sequential* MT system combination architecture. At the same time, there is a rich research literature on *parallel* MT system combination where the same input is

[10] `http://www.statmt.org/wmt16/ape-task.html`

fed to multiple engines and the best output is selected or smaller sections of the various outputs are combined to obtain an improved translation output. In this chapter we show that parallel system combination in the APE stage of a sequential MT-APE combination yields substantial translation improvements both in terms of automatic evaluation metrics as well as in terms of productivity improvements measured in a post-editing experiment. In addition, we also show that system combination on the level of APE alignments yields further improvements. Overall our APE system yields statistically significant improvements of 5.9% relative BLEU over a strong baseline (English–Italian Google MT) and 21.76% significant productivity increase in a human post-editing experiment with professional translators.

The translations provided by MT/APE systems often need to be corrected by human translators to make them publishable. In the localization and translation industry, CAT tools are widely used by professional translators in their regular work practice. CAT tools are computer software that facilitates translators' work in terms of ease of use, faster project delivery and saving translators' time and cost due to (partial) automation. However, translation tools are progressively changing due to technological advances. Automatic translations or translation suggestions produced by these tools may not always be correct. Because of this and partly due to the fear of job loss due to progressive automation, translation tools are not always well accepted by processional human translators in traditional translation workflows. This is a well-known problem for the translation industry (TAUS Report, 2010; TAUS/CNGL Report, 2010).

Case studies (O'Brien et al., 2009; TAUS Report, 2010) have shown that the deployment of an MT/APE engine can be beneficial to all sides, including clients, translators and language service providers (LSP) in terms of cost and time saving. However, the overall picture remains mixed: on the one hand, often MT/APE is cheap and easy to use, while on the other hand, in many cases, the quality of translation is not always satisfactory: sometimes professional translators prefer to translate from scratch. This leads us to investigate the following research question.

RQ6: *How can human interaction with CAT tools be optimized in existing MT workflows?*

Schematically the research and the research questions presented in this thesis can be represented as in Figure 1.3.

For the purposes of this thesis, we divide workflows into three general sections: "upstream MT", "core MT" and "downstream MT". Upstream MT addresses data preprocessing and extraction, core MT addresses hybrid approaches to MT while downstream MT addresses PE, APE and CAT tools with the human in the loop. The research question addressed in this thesis address all 3 stages of this generic workflow (Figure 1.3).

1.1 Publications Resulting from the Research Presented in this Thesis

1.1.1 Chapter 3

- Santanu Pal, Partha Pakray and Sudip Kumar Naskar. 2014. Automatic Building and Using Parallel Resources for SMT from Comparable Corpora, In Hybrid Approaches to Translation (HyTra-2014) Workshop in 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2014), Gothenburg, Sweden, pages 26–30, April 2014.

  Contributions: We present an automatic approach to extracting parallel fragments from comparable corpora for enriching MT training data for low resource language pairs. First author paper. The technological contributions offered in this work is the application of textual entailment (TE) method in MT research. The TE system was developed by Dr. Partha Pakray, a co-author. I developed the core idea of using TE for extracting parallel fragment from comparable corpora.

- Santanu Pal, Partha Pakray, Alexander Gelbukh, Josef van Genabith. 2015. Mining Parallel Resources from Comparable Corpora to improve performance of Machine Translation, Computational Linguistics and Intelligent Text Processing Lecture Notes in Computer Science Volume 9041, 2015, pages 534–544, April 14-20, 2015, Cairo, Egypt.

  Contributions: This is an extension of the (Pal et al., 2014b) above; however, in this case we applied an advanced TE system. First author paper. In this paper, I used a novel TE method and distributional semantics for text similarity. I applied template-based phrase extraction and a TE based monolingual clustering technique to align parallel phrases from comparable sentence pairs. I developed the core idea of

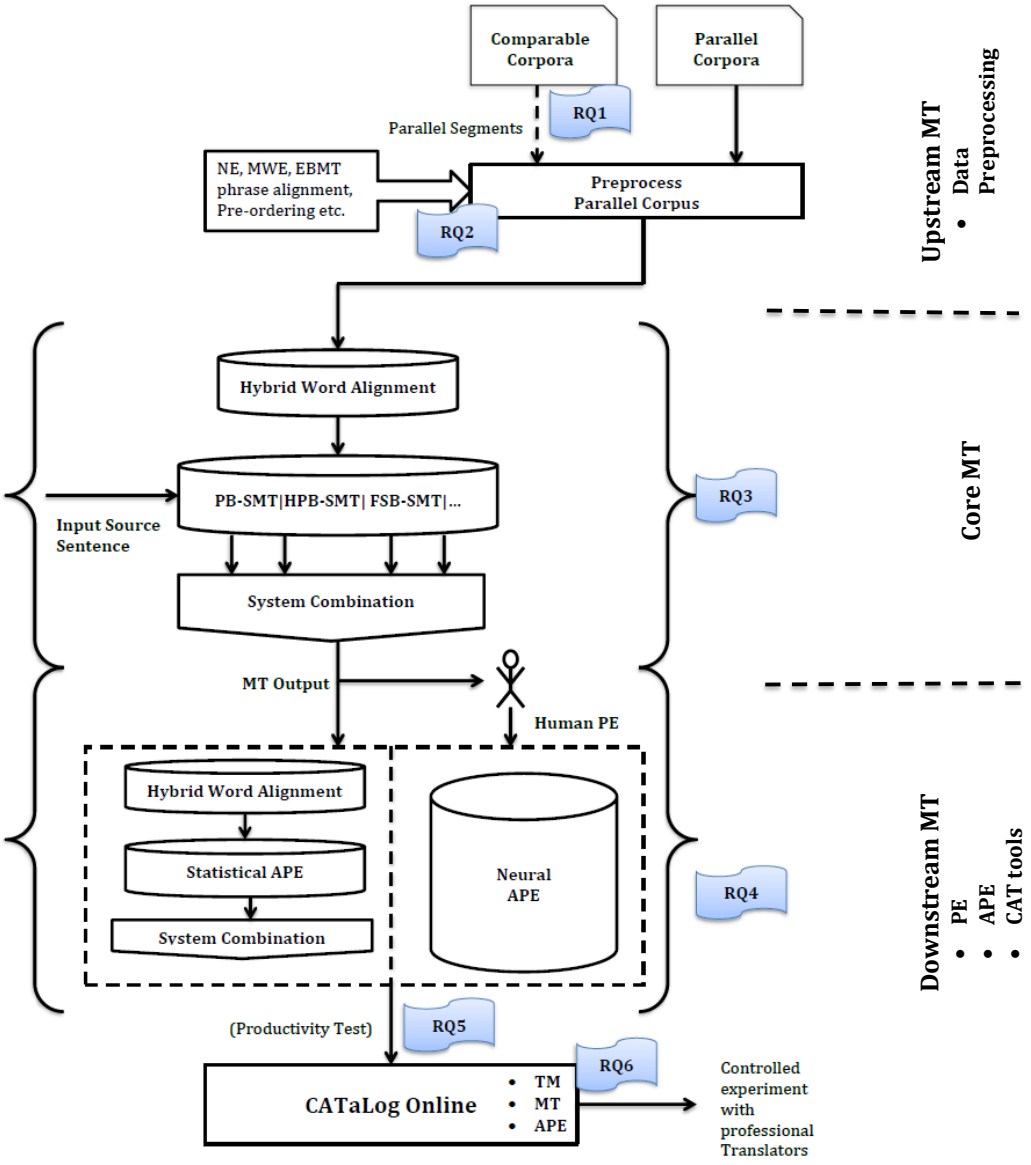


FIGURE 1.3: Schematic design of the research and the research questions presented in this thesis.

using TE and distributional semantic text similarity for extracting parallel fragment from comparable corpora.

1.1.2 Chapter 4

- Santanu Pal, Sudip Kumar Naskar and Sivaji Bandyopadhyay. 2014. Word Alignment-Based Reordering of Source Chunks in PB-SMT. Published in the Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC), Reykjavik, Iceland, pages 3565–3571.

  Contributions: First author paper. In this paper, I show how reordering between distant language pairs can be handled efficiently in PB-SMT. I addressed the problem of reordering between distant languages with prior reordering of the source text at chunk level to simulate the target language ordering. Prior reordering of the source chunks was performed in this work by following the target word order suggested by word alignment. I developed the main idea and performed the central experiments of the work projected in the paper.

- Liling Tan and Santanu Pal. 2014. Manawi: using multi-word expressions and named entities to improve machine translation. In Proceedings of Ninth Workshop on Statistical Machine Translation. WMT 2014. Baltimore, USA, pages 201–206.

  Contributions: This paper describes the English–Hindi MT system submitted to the 2014 WMT translation task. I contributed in alignment of multiwords and named entities and applied this prior alignment to the PB-SMT framework. I developed the core idea of using pseudo-alignment of NEs and MWE as an additional training corpus in PB-SMT framework.

- Santanu Pal, Ankit Srivastava, Sandipan Dandapat, Josef van Genabith, Qun Liu and Andy Way. 2014. USAAR-DCU Hybrid Machine Translation System for ICON 2014. In Proceedings of the 11^{th} International Conference on Natural Language Processing, ICON-2014, Goa, India.

  Contributions: First author paper. This paper presents the USAAR-DCU MT system submitted to the NLP Tools Contest in ICON 2014. I developed the core

idea of using an effective preprocessing method and applying explicitly aligned bilingual terminology, i.e., named entities, to the PB-SMT pipeline and finally developed a simple but effective hybridization technique for using multiple knowledge sources. All the experiments were carried out and research contributions were made by myself; all other co-authors guided me during my secondments in Dublin City University.

- Santanu Pal, Sudip Kumar Naskar, Josef van Genabith. 2015. UdS-Sant: English–German Hybrid Machine Translation System. In the Proceedings of the EMNLP 2015 Tenth Workshop on Statistical Machine Translation (WMT 2015), Lisbon, Portugal, pages 152–157.

  Contributions: First author paper. In this paper, I develop an English–German Hybrid MT system submitted to the Translation Task organized in WMT 2015. I incorporated additional knowledge such as extracted bilingual named entities and bilingual phrase pairs induced from example-based methods into PB-SMT. I developed the core idea and performed the central experiments of the work projected in the paper.

- Santanu Pal and Sudip Kumar Naskar. 2016. Hybrid Word Alignment. In "Hybrid Approaches to Machine Translation". Springer International Publishing Switzerland. M.R. Costa-jussà et al. (eds.), Hybrid Approaches to Machine Translation, Theory and Applications of Natural Language Processing.

  Contributions: First author paper. In this paper, I present a hybrid word alignment model for PB-SMT. This provides most informative alignment links which are offered by both unsupervised and semi-supervised word alignment models. I proposed an algorithm where two unsupervised word alignment models, namely GIZA++ and Berkeley aligner, and a rule based word alignment technique are combined together. The core part of the experiment and the research methodology have been designed by me. Jointly with my co-author we made substantial contributions to conception and design, and/or acquisition of data, and/or analysis and interpretation of data, and experimental outcomes.

- Santanu Pal, Sudip Kumar Naskar, Josef Van Genabith. 2016. Forest to String Based Statistical Machine Translation with Hybrid Word Alignments. Computational Linguistics and Intelligent Text Processing Lecture Notes in Computer Science, 2016, CICLING-2016. Konya, Turkey.

  Contributions: First author paper. In this paper, I show how hybrid word alignment integrated into various experimental settings of Forest to String based SMT can provide considerable improvement over state-of-the-art Hierarchical Phrase based SMT. The research also demonstrates that additional integration of NEs, their translations and EBMT phrases into the Forest to String Based SMT system provides considerable performance improvements.

1.1.3 Chapter 5

- Santanu Pal, Mihaela Vela, Sudip Kumar Naskar, Josef van Genabith. 2015. USAAR-SAPE: An English–Spanish Statistical Automatic Post-Editing System. In the Proceedings of the EMNLP 2015 Tenth Workshop on Statistical Machine Translation (WMT 2015), Lisbon, Portugal, pages 216–221.

  Contributions: First author paper. I developed an APE system submitted to the APE Task organized in WMT 2015. I designed three basic components: corpus preprocessing, hybrid word alignment and a PBSMT system integrated with the hybrid word alignment. The hybrid word alignment consists of a combination of multiple word alignments into a single word alignment table. The PB-SMT based APE system was trained on Spanish MT output and the corresponding manually post-edited output. I developed the core idea and performed the central experiments of the work presented in the paper.

- Santanu Pal. 2015. Statistical Automatic Post Editing. In The Proceedings of the EXPERT Scientific and Technological workshop.

  Contributions: Solo author paper. In this paper, I built a hierarchical phrase based APE system that can automatically handle and estimate word insertion error (by considering one-to-many alignment links between MT–PE aligned data), word deletion error (by considering many-to-one alignment links between MT–PE aligned data), lexical error (by estimating lexical weighting during model estimation) and

word ordering error (using a hierarchical model facilitates word ordering since it uses synchronous context free grammar (SCFG) based hierarchical phrases).

- Santanu Pal, Marcos Zampieri, and Josef van Genabith. 2016. Usaar: An Operation Sequential Model for Automatic Statistical Post-editing. In Proceedings of the First Conference on Machine Translation. WMT 2016. Association for Computational Linguistics, pages 759–763.

  Contributions: First author paper. In this paper, I developed an English–German APE system which is based on Operation Sequence Model combined with PB-SMT system. The system is trained on monolingual settings between MT outputs produced by a black-box MT system and their corresponding post-edited version.

- Santanu Pal, Sudip Kumar Naskar, Mihaela Vela and Josef van Genabith. 2016. A Neural Network based Approach to Automatic Post-Editing. In the Proceedings of the The 54$^{\text{th}}$ Annual Meeting of the Association for Computational Linguistics (ACL 2016), Berlin, Germany, pages 281–286.

  Contributions: First author paper. My contribution in this work is designing the core components, introducing deep neural networks into automatic post-editing pipeline and setting up and carrying out experiments. I collaborated with the industry partner (Translated SRL, Rome, Italy) of the EXPERT project and used their data and translators for the APE experiments. My co-author Dr. Mihaela Vela helped complete the human evaluation.

- Santanu Pal, Sudip Kumar Naskar and Josef van Genabith. Multi-Engine and Multi-Alignment Automatic Post-Editing and its Impact on Translation Productivity. In the 26$^{\text{th}}$ International Conference on Computational Linguistics (COLING 2016), Osaka, Japan, pages 2559–2570.

  Contributions: First author paper. I developed the core idea, designed the core components, introduced alignment combination and multi-engine system combination into the automatic post-editing pipeline, conducted automatic evaluation and human evaluation, applying the model in an industrial setup.

1.1.4 Chapter 6

- Tapas Nayek, Sudip Kumar Naskar, Santanu Pal, Marcos Zampieri, Mihaela Vela and Josef van Genabith. CATaLog: New Approaches to TM and Post Editing Interfaces. In the Proceedings of the 1st Workshop on Natural Language Processing for Translation Memories (NLP4TM), collocated with RANLP 2015, Hissar, Bulgaria, pages 36–42.

  Contributions: In this paper we proposed a novel retrieval technique and post-editing interface for TMs. One of the novel features of CATaLog is a color coding scheme that is based on the similarity between an input segment and the retrieved TM segments, which helps the translators to identify portions of the sentence which are most likely to require post-editing thus demanding minimal effort and increasing productivity. I developed the ideas on color-coding between input text and TM segments (both for source and target).

- Santanu Pal, Marcos Zampieri, Mihaela Vela, Tapas Nayak and Sudip Kumar Naskar, Josef van Genabith. 2016. CATaLog Online: Porting a Post-editing Tool to the Web. In the Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016), pages 599–604.

  Contributions: First author paper. I developed a free-ware software (*CATaLog online*) that can be used through a web browser. *CATaLog online* is the web version of CATaLog with faster TM retrieval and provides a novel and user-friendly online CAT environment to post-editors and translators to reduce post-editing time and effort. It collects post-editing logs which are a fundamental source of information for translation process research. CATaLog online remotely monitors and records user activities generating a wide range of logs. It also provides on-demand MT output that automatically learns from post-editor feedback. The tool provides a complete set of log information currently not available in most commercial CAT tools. Other co-authors helped to reproduce some modules from the *CATaLog* desktop version and also shared their expertise and knowledge about state-of-the-art CAT technology.

- Tapas Nayek, Santanu Pal, Sudip Kumar Naskar, Sivaji Bandyopadhyay and Josef van Genabith. 2016. Beyond Translation Memories: Generating Translation Suggestions based on Parsing and POS Tagging. In the Proceedings of the 2nd Workshop on Natural Language Processing for Translation Memories (NLP4TM 2016), Portoroz, Slovenia, Pages.

  Contributions: Traditionally, TMs do not generate any translation. In this paper we introduced an important functionality in TM, that of proposing a new translation. This improves HCI issues with TM since this new functionality generates a new translation based on the translation template chosen by the user. I developed the core idea on syntactic matching in TM – a step beyond traditional TM, and helped the first-author with implementation.

- Santanu Pal, Sudip Kumar Naskar, Marcos Zampieri, Tapas Nayak and Josef van Genabith. CATaLog Online: A Web-based CAT Tool for Distributed Translation with Data Capture for APE and Translation Process Research. In the 26th International Conference on Computational Linguistics (COLING 2016), Osaka, Japan, pages 98–102.

  Contributions: First author paper. In a bid to reduce post-editing time and effort, improve the post-editing experience and capture data for incremental MT/APE and translation process research, I upgraded *CATaLog Online* (Pal et al., 2016e) to facilitate distributed translation where teams of translators can work simultaneously on different sections of the same text. I also incorporated user activity logging and automatic live word alignment recording features in the online version of the CAT tool. Others co-authors helped with design issues and debugged the Demo version of the tool to find various implementation bugs during quality testing.

Chapter 2

Literature Survey

This chapter provides an overview of previous general basic research related to the research presented in this thesis. Later chapters will review further specific literature relevant to the specific topic of the local chapter in question (see related research sections in Chapter 3, 4, 5 and 6).

2.1 Introduction

The following sections present an overview of research in Machine Translation (MT), Post-Editing (PE), Automatic PE (APE) and interactive translation workflow as an increasingly central practice in the translation field that is relevant to our research presented in the thesis. Section 2.2 describes general approaches to MT. Section 2.3 and 2.4 report recent progress directly related to the MT research we carried out on hybrid MT systems. Section 2.5 reviews relevant research on APE to improve over a first-stage MT system. Finally, Section 2.6 reviews research on translation workflows including human-machine interactive MT systems.

2.2 Machine Translation

MT is an Artificial Intelligence (AI) problem. Conventionally, MT is a computer application which automatically translates texts from one natural language to another.

Architectures of early knowledge-based or rule-based MT systems can be roughly organized into the three classes – Direct, Transfer and Interlingua. They differ on their depth of analysis. The deeper the analysis, the less transfer is needed. The direct approach is the most primitive form of transfer, consisting of word-to-word replacements. The transfer approach consists of three stages: analysis, transfer and synthesis. The interlingual approach has the most "degenerate" form of transfer, i.e., the transfer mapping is essentially non-existent.

In direct MT systems, the source language (SL) words or phrases are directly replaced by target language (TL) words or phrases by means of a bilingual dictionary look-up (Hutchins, 1995). The translations provided by the simplest possible direct MT systems follow the same word order as that provided by the SL. In general, however, TL strings do not follow the same word order that occurs in SL strings. In order to improve translation quality, a direct MT system has to do some local syntactic and morphological analysis before the bilingual dictionary look-up. Translation may need to execute local reordering according to local syntactic analysis. Applying limited syntactic information in a direct MT system increases the readability of the translated text to some extent, but does not always follow a general linguistic theory or a global syntactic analysis necessary for producing good translations. The direct MT system architecture mainly relies on well-developed dictionaries, morphological analysis, and text processing software to produce reasonable translations of the source text into a series of words and phrases in the target language translation. Using a direct MT system, often only simple source sentences can be translated well. A problem in the direct architecture is often the selection of the target language words for the source language words (lexical ambiguity). Direct MT systems often fail to provide a good solution to the lexical selection problem as they are only able to consider a limited local context. As a result, Direct MT systems often produce poor translations, especially if such systems are used to translate between "distant" languages.

The interlingual approach attempts to develop a language independent representation of the source language text that is meant to capture all the linguistic information necessary to generate the appropriate target language translation (Hutchins, 1995). There are many theoretical advantages of an interlingual approach, especially when one thinks of multilingual systems translating between many language pairs. The interlingua approach consists of only two phases: analysis and generation. During analysis, the SL text is converted to

an interlingual representation. In the generation phase, the TL text is generated from this interlingual representation. The interlingual approach to MT generally uses an abstract system of semantic relations to represent events and states of affairs including participant relations, spatial relations and temporal relations. Interlingual representations need to be very rich and tend to be extremely knowledge intensive. One of the theoretical benefits of such a system is that the meaning representation should be language agnostic and therefore (relatively) uniform across multiple source languages, so that in principle it should take fewer components to add a new language in a multilingual system. In particular, no language pair specific transfer component should be required. However, often, the style and emphasis of the original text are lost in the interlingual approach because interlingual representations are highly generic and independent of the particulars of the linguistics of the source and target text. However, and in addition, in practice, it is extremely difficult to create a full blown abstract yet detailed enough "language-independent" representation for human languages, parse the source sentence into such a representation, and from it generate the target sentence (Dorr et al., 2006).

By contrast, the syntactic or semantic transfer approach produces a translation in three different phases: (a) **analysis** of the input into a SL syntactic or semantic representation, (b) **transfer** of that representation into corresponding TL structure, and (c) **synthesis** of the translation output from that structure. This architecture is specialized for a particular pair of languages, and the transfer component converts a source representation into a corresponding target representation. The biggest disadvantage of this approach is that a large set of language-pair specific transfer rules must be constructed for each SL/TL pair. Furthermore, the analysis, transfer and synthesis phases follow each other in sequence; therefore, propagation of errors in each stage can lower translation quality.

With regard to the acquisition of the required knowledge, MT paradigms can be broadly divided into two categories – Rule Based MT (RBMT, knowledge-based MT) and Corpus Based MT (CBMT, data-driven MT). Traditional RBMT relies on hand-built linguistic rules and bilingual dictionaries for each language pair[1]. This requires extensive linguistic and programming skills and is time consuming and expensive to scale. On the other hand, CBMT uses bilingual and target monolingual corpora to (semi-) automatically acquire

[1]If suitably annotated data are available, RBMT system can also be learned from data.

the required translation knowledge. Recently, corpus-based MT has delivered increasingly higher quality translations. There are many approaches that have been proposed in the last few decades such as Example-based Machine Translation (for an overview see e.g., (Carl and Way, 2003)), Statistical Machine Translation (Brown et al., 1993; Koehn, 2010) and Neural Machine Translation (Sutskever et al., 2014; Cho et al., 2014a,b; Bahdanau et al., 2015; Luong et al., 2015a,b; Sennrich et al., 2016a).

Out of these, in terms of large-scale evaluations, until recently SMT has been the most successful and efficient MT paradigm (Koehn, 2010)[2]. The quality of SMT depends on good quality word alignment as well as good phrase pair estimation, both of which can often be achieved by using large amounts of sentence-aligned parallel corpora. However, SMT for low-resource or distant language pairs usually produces inferior quality translation.

2.2.1 Example Based Machine Translation

EBMT was first introduced by Nagao (1984). According to Nagao (1984), EBMT learns and translates like a human i.e., in order to translate new sentence, a human has the tendency to make use of translation examples which they have previously encountered. An EBMT system relies on past translations together with a "divide and conquer" approach to derive the target output for a given input by using a set of bilingual sentence-aligned parallel examples which act as bilingual knowledge source to induce translations of sub-sentential fragments. Traditional EBMT systems perform translation in three steps: matching, alignment and recombination (Somers, 2003) drawing a parallel with analogous phases in traditional transfer-based MT systems (analysis, transfer, and generation). Sentence frames are sequentially compared in a matching step. The alignment step is used to identify which parts of the corresponding translation are to be reused. Recombination is the final step where aligned basic sentence structures are combined with aligned sub-sentential translation pairs. Two main approaches to EBMT can be distinguished: (i) "pure EBMT" (Lepage and Denoual, 2005), where no training or preprocessing stage takes place and the runtime complexity is considerable, and (ii) "compiled approaches" (Cicekli and Guvenir, 2001), where training usually consists of compiling units below the sentence level before runtime.

[2]However, recently SMT has been challenged by neural approach to MT, see Section 2.4 below.

2.2.2 Statistical Machine Translation

More than six decades ago, Weaver (1949) expressed the idea of applying statistical methods to translate a word by taking its context into account. However, researchers abandoned this approach due to the complexity involving implementation at that time. Four decades after the proposal of Weaver (1949), Brown et al. (1993) modelled MT with a probabilistic model, namely the noisy channel model of translation. The noisy channel model of translation (Brown et al., 1993) maximizes the probability $p(e|f)$ of generating a target sentence $e = e_1...e_i...e_I$ given a source sentence $f = f_1...f_j...f_J$. According to the noisy channel model, the translation task can be viewed as a process of finding the $\hat{e}$ that maximizes the probability of $p(e|f)$ as in Equation 2.1:

$$\hat{e} = argmax_e \; p(e|f) = argmax_e \; p(f|e) \times p(e) \tag{2.1}$$

where $p(f|e)$ and $p(e)$ denote the translation model and the target language model, respectively.

2.2.3 Word Based SMT

Word based SMT uses words as their basic translation units and was developed by IBM (Brown et al., 1993). IBM-1 to IBM-5 as well as HMM based models (Vogel et al., 1996) estimate word alignment from a large volume of bilingual parallel corpus employing unsupervised techniques. Once the IBM models are trained, the alignments between source and target words with maximum probability are extracted. These alignments are called "Viterbi alignments". Word based models are the standard starting point for most state-of-the-art alignment and translation models.

2.2.4 Phrase Based SMT

The basic problem of word-based SMT is that it does not capture neighboring contexts well, since the translation unit of this model are the individual words. As a result, word-based models often result in poor lexical selection and they may fail to maintain phrasal cohesion between the phrases of source and target languages. To overcome these

limitations, a phrase-based alignment and translation model (Och et al., 1999) was proposed, which extends the basic translation unit from words to phrases. This model is able to produce alignments which consist of m-to-n non-consecutive word translational correspondences.

Phrase-level alignment starts by segmenting the source sentence into phrases with arbitrary boundaries. Phrase pairs are extracted from the sentence pairs that are consistent with the refined word alignment matrix (Koehn et al., 2003; Koehn, 2009). The approach described in (Och, 2002; Koehn et al., 2003) serves as the basis of state-of-the-art phrase-based SMT model. Instead of the original formulation of the translation problem as a noisy-channel model, phrase-based SMT employs a log-linear interpolation over a set of features as described in the next subsection.

2.2.5 Log-linear Model for SMT

The state-of-the-art phrase-based SMT (PB-SMT) model (Koehn et al., 2007) follows the log-linear model representation (Och, 2002) which can combine together an arbitrary number of features into a single model. A PB-SMT model usually employs the following set of features:

- Phrase translation probability and inverse phrase translation probability
- Lexical translation probability and inverse lexical translation probability
- Word penalty and phrase penalty
- Distance-based or lexicalized phrase reordering models
- N-gram language model

Any additional feature that applies to the source and target phrase pairs can be incorporated into the log-linear model. Each feature of the log-linear model is associated with a weight which is usually estimated using minimum error rate training (MERT) (Och, 2003). In log-linear phrase-based SMT, the posterior probability $p(e|f)$ is modeled as a log-linear combination of features (Och, 2002). This usually consists of M translational features, and the language model, as in Equation 2.2:

$$\log p(e|f) = \sum_{m=1}^{M} \lambda_m h_m(f, e, s_1^k) + \lambda_{LM} \log p(e) \tag{2.2}$$

where $s_1^k = s_1 \ . \ . \ . \ s_k$ denotes a segmentation of the source and target sentences respectively into the sequences of phrases ($\hat{e}_1^k = \hat{e}_1 \ . \ . \ . \ \hat{e}_k$) and ($\hat{f}_1^k = \hat{f}_1 \ . \ . \ . \ \hat{f}_k$) such that $\forall 1 \leq k \leq K$, $s_k = (i_k, b_k, j_k)$, $\hat{e}_k = e_{i_{k-1}+1} ... e_{i_k}$, $\hat{f}_k = f_{b_k} ... f_{j_k}$ (we set $i_0 = 0$) and each feature λ_m and λ_{LM} in Equation 2.2 are estimated using MERT or other tuning methods (e.g., MIRA (Cherry and Foster, 2012)).

2.2.6 Reordering Model

In the PB-SMT framework, reordering is typically handled by two models: a distortion model (Brown et al., 1993) and a lexicalized reordering model (Koehn et al., 2005; Galley and Manning, 2008). IBM models 1 and 2 define the distortion parameters in accordance with the word positions in the sentence pair instead of actual words at those positions. Models 4 and 5 replace absolute word positions with the relative word positions. However all these models are limited to only word movements; they do not consider phrasal movements. Koehn et al. (2005) proposed a relative distortion model in PB-SMT. The model works in terms of the difference between the current phrase position and the previous phrase position in the source sentence. Basic PB-SMT models consider word movements up to a few tokens which could be increased to consider long distance reordering; however, in practice higher distortion limits often result in degraded performance (Koehn et al., 2007). Lexicalized reordering is involved in the movement of words which are moved frequently together. It considers three types of reordering – monotone (M), swap (S), and discontinuous (D) – by considering the orientation of the previous and the next word of each phrase pair. The orientation is called monotonous if the previous word of the source is aligned with the previous word of the target. The orientation of swap occurs when the next word in the source is aligned with the previous word of the target; finally the orientation is discontinuous if neither of the two above mentioned cases are true. The reordering model is built by estimating the probabilities of the phrase pairs associated with the given orientation. Generally the orientation probability is estimated by the count of each orientation type divided by sum of the count of each orientation type.

2.2.7 Language Model

A language model (LM) estimates the likelihood of appearance of a sequence of words in a language. In other words, in SMT, the language model probabilistically measures the linguistic well-formedness of the sentences generated by the MT system. The goal of a statistical LM is to learn the joint probability function of sequences of words in a language. In language models, a word-sequence consisting of n words is referred to as an n-gram. It is impossible to estimate probabilities for large n-grams reliably from data: even very large corpora do not show all possible combinations. To address this problem, estimation is based on the Markov assumption that considers only limited context of $n-1$ previous words. Still, a zero probability is assigned to unseen n-grams. Many smoothing techniques have been introduced to solve this problem. The idea behind smoothing techniques is that some probability mass is subtracted out ('discounting') from seen n-grams and redistributed to unseen n-grams. SMT researchers usually build language models with the interpolated modified Kneser-Ney smoothing (Kneser and Ney, 1995) technique. Interpolation causes the discounted n-gram probability estimates at the specified order n to be interpolated with lower order estimates. The LM order is usually set to 5-gram to 7-gram n-gram to capture a reasonable range of contexts on the target side. Due to the data sparsity issue, n-gram language model probability is difficult to reliably estimate since basic models do not use any information about similarities between words. To address this issue, some approaches involve word clustering techniques (Yuan, 2006; Shi et al., 2013) while Bengio et al. (2003) introduced a feed-forward neural probabilistic LM (NPLM) that operates over distributed representations, real-valued vectors in a high-dimensional feature space. Neural network architectures for language modeling include feed-forward (Bengio et al., 2003), recurrent (Mikolov et al., 2010), sum-product (Cheng et al., 2014) and convolutional (Wang et al., 2015) neural networks.

2.3 Hybrid MT

Each MT approach discussed above has its own advantages and disadvantages. To avoid the limitations and to muster the strengths of all the aforementioned methods, hybrid approaches were proposed to combine the best features of all or a selection of methods.

Much research in MT includes some degree of hybridization based on e.g., incorporating linguistic knowledge in terms of preprocessing or on purely statistical model combination.

Data preprocessing plays a crucial role in NLP, especially with regard to parsing and MT. NEs and MWEs in particular pose difficulties in terms of identification and translation. In parsing, NE and MWE identification is like a chicken and egg problem in the sense that where should it be best fit in the pipeline – before parsing or after parsing; the question is whether MWE is a tokenization problem or a parsing problem. Handling MWEs in SMT deals with two challenging tasks: identification of MWEs and their incorporation into state-of-the-art SMT. Much research has been carried out on both MWE extraction and incorporation within SMT, as described below.

A log likelihood ratio based hierarchical reduction algorithm to automatically extract bilingual MWEs was reported in (Ren et al., 2009). Venkatapathy and Joshi (2006) reported a discriminative approach to use the compositionality information of verb-based MWEs in order to improve the word alignment quality. Carpuat and Diab (2010) replaced the binary feature by a count feature representing the number of MWEs in the source language phrase in SMT. Pal et al. (2013b) and Tan and Pal (2014) used various statistical techniques to extract MWEs from bilingual data and used these bilingual MWEs as additional training material to examine the usefulness of these bilingual MWEs in SMT. Pal et al. (2013b) observed the highest improvement with an additional feature that identifies whether or not a bilingual phrase contains bilingual MWE(s). A hybrid approach to identify MWEs from English–French parallel data was proposed by Bouamor et al. (2012a), who aligned only many-to-many correspondences and dealt with highly correlated MWEs. These MWE are then integrated into the MOSES SMT System (Bouamor et al., 2012b) in three ways: (a) adding the extracted bilingual MWEs as additional parallel training material, (b) integrating bilingual MWE candidates into the phrase table[3], and (c) adding a new feature indicating whether a phrase in the phrase table is an MWE or not. One key difference between Bouamor et al. (2012b) and Pal et al. (2013b) is that, Pal et al. (2013b) considered MWEs as single tokens, which ensures that the phrase extraction module never gets a chance to mark a phrase boundary inside an MWE and MWEs are always treated as a whole. MWEs in SMT was also investigated by Lambert and Banchs (2005) for the Verbmobil corpus. The work related to MWE handling in SMT presented

[3]Bouamor et al. (2012b) use the Jaccard Index to define the two directions translation probabilities and set the lexical probabilities to 1.

in Chapter 4 in this thesis will apply multiword NE and MWE knowledge directly to the SMT word alignment and phrase extraction step. Additionally, and orthogonally, we also investigate how EBMT phrases can provide further improvement in SMT.

A major characteristic of state-of-the-art PB-SMT is that phrase pairs are extracted solely based on the knowledge contained in the word alignment table (plus some additional heuristics). The extracted phrases in PB-SMT do not respect linguistically motivated phrase boundaries and may be fragments of linguistically motivated phrases or contain words from neighboring linguistic phrases. Recent research in SMT has investigated how to incorporate syntactic knowledge into PB-SMT systems to improve translation quality. Syntax based SMT systems have provided promising improvements in recent years. Syntax based SMT can be divided into two categories: formal syntax-based systems where there is no need for using any additional parser with a linguistically motivated grammar (Chiang, 2005), and linguistically motivated syntax-based systems that use PCFG (Liu et al., 2006; Huang, 2006; Mi et al., 2008; Mi and Huang, 2008; Zhang et al., 2009), syntactic word dependency (Ding and Palmer, 2005; Quirk et al., 2005; Shen et al., 2008) or other parsers, e.g., Wu et al. (2011) trained on tree banks. Translation rules can be extracted from aligned string-to-string (Chiang, 2005), tree-to-tree (Ding and Palmer, 2005) or tree/forest-to-string (Galley et al., 2004; Mi et al., 2008; Wu et al., 2011) data structures and their corresponding word alignment tables. The approach described in Chiang (2005) for incorporating syntax[4] into PB-SMT targets mainly phrase reordering. Under this approach, hierarchical phrase translation probabilities are used to handle a range of reordering phenomena. Marcu et al. (2006) present a similar extension of PB-SMT with syntactic structure on the target side. Zollmann and Venugopal (2006) extend the work introduced in Chiang (2005) by augmenting the hierarchical phrase labels with syntactic categories derived from parsing the target side of the parallel corpus. They associate a target parse tree with the corresponding search lattice provided by lexical phrases on the source sentence and assign a syntactic category to phrases which align directly with the parse hierarchy. Similar to Chiang (2005), a chart-based parser with a limited language model was used.

[4]This approach is formally syntax based and uses synchronous context free grammar, it is not necessarily linguistically syntax-based because it induces a grammar from a parallel text without relying on any linguistic annotations or assumptions.

Systems adopting the same (or different) MT framework usually produce different translations for the same input, due to their differences in training data usage, different preprocessing methods, different alignment strategies and adopting various decoding processes, etc. It is therefore beneficial to design a combined framework of multiple systems that combines the output of these MT systems and produces better translations compared to any single system. MT system combination provides an approach to hybrid MT where output from different MT engines belonging to same or different MT paradigms are considered in a bid to either select the best hypothesis from among the candidate hypotheses, or to build a new hypothesis altogether by combining parts of the candidate hypotheses. Many MT system combination approaches have been proposed over the years. These can be roughly grouped into three different categories: (i) hypothesis selection (Rosti et al., 2007a; Hildebrand and Vogel, 2010), (ii) re-decoding (He and Toutanova, 2009; Devlin and Matsoukas, 2012), and (iii) confusion network decoding (Matusov et al., 2006; Rosti et al., 2007b). Further gains can be obtained by the lattice decoding model (Feng et al., 2009; Du et al., 2010) and the paraphrasing model (Ma and McKeown, 2015). Our own hybrid architecture is based on a confusion network based system combination. Confusion Network decoding typically follows four steps:

1. **Backbone selection:** This method selects a backbone/skeleton from all the candidate hypotheses. The backbone defines the word order of the final translation. The backbone selection strategies generally follow Minimum Bayes Risk (MBR) decoding (Rosti et al., 2007b; He et al., 2008). Translation edit rate (TER) or modified BLEU score are often used as the loss function in MBR. The quality of the combination output depends on which hypothesis is chosen as the selected backbone since the backbone determines the word order of the final fusion translation.

2. **Hypothesis alignment:** All words of each hypothesis are aligned against the backbone. To establish alignment between the hypothesis and the backbone, many approaches have been proposed: the edit distance alignment algorithm (Bangalore et al., 2002) which only allows monotonic alignment, a heuristic-based matching algorithm which allows non-monotonic alignments (Jayaraman and Lavie, 2005), GIZA++ (Matusov et al., 2006), TER alignment toolkit (Rosti et al., 2007a,b), the ITG-based method (Karakos et al., 2008), the IHMM-based word alignment method (He et al., 2008) in which the parameters are estimated indirectly from a variety

of sources, and the systematic comparisons method (Chen et al., 2009; Rosti et al., 2012).

3. **Confusion network construction:** A confusion network is prepared based on hypothesis alignments. Hypothesis alignment algorithms produce many-to-one mappings between the hypothesis and backbone. The word alignments need to be normalized to one-to-one word alignments by simply removing duplicate links since the confusion network is built from one-to-one word alignments. The hypothesis words need to be reordered according to the backbone word order.

4. **Confusion network decoding:** This step deals with choosing the best translation path from the confusion network through a beam-search algorithm with a log-linear combination of a set of feature functions. The chosen path achieves the highest confidence in the network. The feature functions include: a language model, word penalty, weights on word arcs and n-gram posterior probabilities. The total weights of feature functions are optimized using MERT (Och, 2003).

2.4 Neural MT

SMT has proved to be the most successful and dominant MT approach in large-scale evaluations until recently. Over the last 3–4 years, a number of researchers proposed the application of neural networks to learn conditional distributions in MT (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Cho et al., 2014a,b; Bahdanau et al., 2015; Luong et al., 2015a,b). These approaches use a recurrent neural network architecture (RNN) with long short term memory (LSTM), typically consisting of two components: an **encoder** that encodes source sentences and a **decoder** which decodes into target sentences. Neural MT (NMT) systems have achieved performance similar to and even better (WMT 2016) than the state-of-the-art PB-SMT system. The advantages of Neural MT (NMT) over SMT are its simplicity and smaller storage requirements. Unlike SMT, NMT has just one overall system to be optimized end-to-end rather than several components estimated individually which are then put into in a pipeline where the only thing that is adjusted in a final learning step is the weight with which the component contributes to the model. NMT also has an advantage in terms of no error propagation between individual components as in the traditional SMT pipeline. Conceptually, SMT is concerned with the

statistical similarity between phrases and may ignore linguistic similarities other than the surface form leading to sparsity issues; NMT able to avoid such problems. NMT directly models the conditional probability $p(t|s)$ of translating a source sentence, $s = s_1...s_n$ to a target sentence, $t = t_1...t_m$ through an encoder-decoder framework. A potential issue with this encoder-decoder approach is that the neural network compresses all the necessary information of a source sentence into a fixed-length vector. The fixed-length vector representation may turn out to be difficult for long sentences in the training corpus. Bahdanau et al. (2015) introduced an extension of the encoder-decoder model which learns to align and translate jointly. The model generates a target word after (soft-)searching for a set of positions in a source sentence where the most relevant information is concentrated and finally predicts the target word based on the associated context vectors of source positions and all the previously generated target words. This method involves an attention mechanism, a form of random access memory for NMT to cope with long input sequences. The attention mechanism was further extended by Luong et al. (2015a) to different scoring functions, used to compare source and target hidden states, as well as different strategies to place the attention. To train an NMT model, every word in the source or target vocabulary can be represented by an one-hot vector of length V where V is the number of top-most frequent words. One-hot vector representations are further transformed into a sequence of n-dimensional word embedding vectors consisting of the embedding weights. The embedding weights are learned during training and are different for the source and the target words. The word embedding vectors are next fed as input to the two RNNs – one for the source language (an encoder) and the other for the target language (a decoder). The RNNs use an LSTM (Hochreiter and Schmidhuber, 1997) which is able to retain information over long sequences. The encoder obtains the feed-forward weights based on the connected hidden units from the previous time-step RNN to the current time-step RNN block. The decoder is fed through an attention layer, which guides the translation by paying attention to relevant parts of the source sentence. Finally, for each target word, the top layer hidden unit of the decoder is transformed into a score vector of length V and the target word associated with the highest score is selected as the output translation.

2.5 Automatic post-editing

Several approaches to automatic post-editing (APE) have been developed, including: statistical APE over RBMT, statistical APE over SMT and rule-based APE over SMT.

2.5.1 Statistical APE over RBMT

APE over RBMT was first reported by Simard et al. (2007b), where the authors successfully improved the translation quality of an RBMT engine using statistical APE. The statistical APE is based on a PB-SMT system trained in a "monolingual" setting utilizing the outputs of the RBMT system as the source and the human-provided reference translations as the target. Simard et al. (2007b) furthermore compared the post-edited RBMT performance to directly using the PB-SMT system in a bilingual setting. They showed that the post-edited RBMT translations were much better than the translations produced by the standalone baseline PB-SMT system. The authors also carried out their experiment using a PB-SMT system instead of RBMT system as the first stage MT system, however, no improvement in translation quality was observed.

2.5.2 Statistical APE over SMT

Improvements using statistical APE (SAPE) of SMT output were first reported by Oflazer and Durgar El-Kahlout (2007) on English–Turkish MT. Béchara et al. (2011) report SAPE of SMT on French–English MT. Both approaches are based on Simard et al. (2007a,b). Béchara et al. (2011) reported a statistically significant improvement of 0.65 BLEU points (Papineni et al., 2002). They achieved further improvements by adding source side information into the post-editing system by concatenating some of the translated words with their source words, eventually reaching an improvement of 2.29 BLEU points. We will provide more discussion regarding this in Chapter 5.

2.5.3 Rule-Based APE over SMT

A rule-based APE of SMT output was reported by Rosa et al. (2012). They developed a rule-based APE tool, called "Depfix", which is an APE system for English–Czech PB-SMT outputs, based on linguistic knowledge. The authors analyzed the types of errors that are typically produced by an SMT system on their data. The tool consists of a set of rules and a statistical component. The APE tool is able to correct systematic errors of first-stage SMT. The tool produced improved quality translations in terms of both automatic and manual evaluations.

2.6 Translation Workflow

2.6.1 Translation Memory

In the localization industry, human translators typically employ translation memory (TM) systems (Kay, 1997). A basic TM stores segments of translated text as a translation unit of source and target pairs. When a new sentence is encountered, the TM fetches previously translated identical or similar sentences using "fuzzy matching" algorithms usually based on a version of edit distance. The TM algorithm locates translations of stored sentences similar to the input source sentence to be translated and presents the corresponding translations as suggestions to the human translator. For matches with less than 100% similarity, the suggested translation(s) may not be a translation of the new sentence that needs to be translated and such translation proposals may need to be post-edited by the human translator.

2.6.2 Beyond Basic TM Functionalities

TM systems are the most popular type of technology widely used in today's translation market. The acceptance of these tools is based on the fact that they have the ability to reduce the translator's effort, increase their productivity and reduce cost. TMs provide support to translators by retrieving segments of text that were already translated. This can be performed by simple string matching and can be improved by using syntactic/semantic information or paraphrasing (Gupta and Orăsan, 2014; Gupta et al., 2015b).

Due to technological advancement, not only has segment matching become more accurate but developers of these tools have added more features and functions to them as well as translation-related resources (e.g., term banks and TM repositories). Therefore, Computer-Aided Translation (CAT) tools often become complex and require considerable time and effort for the translator to learn and use.

Some features such as terminology extractors, corpora compilation tools, automatic translation systems and translation-related resources are really beneficial and are already integrated in some translation software tools (SDL Trados Studio[5], LiveDocs in MemoQ[6], MyMemory[7], Web-based applications MateCAT[8] and Wordfast[9] as an add-on to Microsoft Word through macros etc.). Translators also tend to prefer working with specific translation software. It would be of interest to find out why translators prefer certain translation software over others and what their preferences in terms of translation software requirements[10] are that developers of such translation software should satisfy.

2.6.3 Needs or Problems Encountered by TM Users

According to a survey[10] (Zaretskaya et al., 2015a,b) based on the popularity of various translation technologies, TM systems appear to be the only type of tools that are used regularly by the majority of professional translators. However, the survey shows that there are still a considerable number of translators who have never heard of such technologies at all. One such type of technology, a concordance system, is in fact unknown to the majority of translators. Tools for compiling or managing corpora are less commonly used on a regular basis. A possible reason is that some of the technologies are only recently integrated into CAT tools but translators are completely unaware of them or they believe that they may not provide satisfactory results or that they are slow. Some technologies may not be considered appropriate for everyday use (e.g. compiling and managing corpora). With regard to usage of MT services, some users use them, few are planning to use them in future and some users abandoned MT due to perceived poor quality. In comparison to automatic translation, TMs turned out to be much more popular compared

[5] `http://www.sdl.com/products/sdl-trados-studio/`
[6] `https://www.memoq.com/`
[7] `http://mymemory.translated.net/`
[8] `https://www.matecat.com/`
[9] `http://www.wordfast.net/`
[10] EXPERT deliverables 2.1: User Requirement Analysis

to other translation support systems. However, the survey found that due to continuous enhancement of the integrated MT systems within CAT, recently many translators are showing the tendency to use MT systems for their regular practice. As a final observation the survey established that a majority of the users also feel that terminology management tools integrated into the translation software are also helpful.

2.6.4 Different types of Interactive MT and Learning from Mistakes

As mentioned earlier, current MT technology is still far from perfect, and in order to achieve good translations, manual post-editing and interaction with the translation process or output is often needed. Interactive MT is a collaborative process where the human and the computer collaborate to generate the final translation and the paradigm may work in an iterative manner. Fully interactive MT can be described as an evolution of MT framework, where human translators check and correct the suggested MT translation(s) as and while the automatic translation is produced (from left to right for e.g., European languages). For any human interaction the MT system proposes a new extension, taking the human correction into account and these steps are repeated until the entire sentence/document has been correctly translated. A significant number of interactive MT systems have been built over the past decade: TransType (Macklovitch, 2006), CasmaCat[11], Lilt[12], etc.

In "traditional" professional environments, the translation process typically follows three stages – translation, editing, and proofreading – to ensure high quality results. CAT tools are generally used by professional translators to achieve their goal. In the first stage, the CAT tool provides sentence-level translations for humans to post-edit by using real-time MT systems or a TM. There is a surge in demand for human quality translation that continues to exceed the capacity of the language services industry. To enable human translators to work more efficiently and to provide better assistance for accelerating their work, new technologies are developed which typically follow the translation workflow (i.e., translation, editing, and proofreading). The translation workflow is often supported by sophisticated workflow management software for language service providers (LSPs) to allow better distributed work among teams of translators, outsourcing documents to

[11] http://www.casmacat.eu/
[12] https://lilt.com/

freelance translators, etc. The translation editing software within the translation workflow provides automatic suggestions from TM, terminology banks, bilingual dictionaries and recently from MT and APE.

Ideally, the integrated translation system in CAT tools should be able to learn from the corrections provided by human translators and should avoid making similar mistakes repeatedly. Every time a human translator corrects the translation system output, new bilingual sentence/segment pairs are produced.

Adapting incremental SMT from the newly generated data in CAT environments, Nepveu et al. (2004) experimented with adaptive language and translation models in the context of an interactive CAT environment. They used cache-based grammars and language models that incorporate incrementally translated data. This approach led to improvements in translator productivity (Bertoldi et al., 2013). Levenberg et al. (2010) incorporated post-edited bilingual data into on-demand grammar extraction and introduced suffix array data structures that can be dynamically updated.

In Chapter 6 we present a new web-based post-editing tool, called *CATaLog Online*, enhanced with a number of new features. This tool can be used as a generic CAT tool as well as for post-editing TM segments or MT output. The tool captures and provides a complete set of activity log information currently not available in most CAT tools. This tool is also convenient for translation process research. The tool offers the following advantages: (i) color-coded TM translation suggestions (highlighted TM source and corresponding target fragments are shown in the same interface), (ii) a wide range of editing logs, (iii) alignment between source, TM/MT/APE output and the results of human PE, (iv) an improved TM similarity measure and search technique (Pal et al., 2016e), and (v) additional translation options from APE.

2.7 Conclusions

This chapter presents a literature review of research relevant to the research presented in the thesis. In the following chapters, we present our own research and show (i) how parallel text fragments can be extracted from comparable corpora which can be added to the bilingual training corpus as additional training material to improve the performance

of SMT systems for low-resource language pairs, (ii) the optimal use of existing parallel resources and an improved hybridization method for MT, (iii) different approaches to APE over a first stage MT system, and finally (iv) how human interaction with CAT tools can be optimized in existing MT workflows.

Chapter 3

Mining Parallel Resources from Comparable Corpora

Statistical Machine Translation (SMT) is based on a probabilistic model which is learned from sentence-aligned parallel corpora where each sentence in the source is paired with its translation in the target. Due to the fact that parallel corpora remain a scarce resource for many language pairs (e.g., English–Indian languages) and are often restricted to certain domains, comparable corpora can to some extent provide a possible solution to this data scarcity problem for corpus-based approaches to MT. Many studies and applications in both linguistic and language engineering communities use comparable corpora as resources, and these can play an important role in improving the quality of MT (Smith et al., 2010). Extracting parallel text fragments, paraphrases or sentences from comparable corpora is particularly useful for SMT (Gupta et al., 2013).

In general, comparable documents are not strictly parallel: a comparable corpus consists of documents in two languages, but these are not sentence-by-sentence translations of each other; rather the documents are about the same topic. While the sentences of comparable corpora usually are not (exact) translations, parallel documents convey information on the same topic or event and hence there should exist some sentential or sub-sentential level of parallelism.

Previous studies on comparable corpora mainly focused on: (i) parallel data extraction in the form of bilingual lexicon extraction (BLE) (Fung and McKeown, 1997; Pirkola et al.,

2001; Rapp, 1995), parallel fragment extraction (Quirk et al., 2007) and parallel sentence extraction (Munteanu and Marcu, 2005), (ii) Translation model improvement (Daumé and Jagarlamudi, 2011; Klementiev et al., 2012) and (iii) Language model adaptation (Zhao et al., 2004). The main focus of this chapter is to exploit comparable corpora to address the scarcity of parallel data for less resourced languages. We propose novel approaches to extract parallel fragments from comparable corpora by applying a textual entailment (TE) method and a template based approach.

This chapter addresses **RQ1:** *How can MT for low resource languages be improved?* We extract parallel segments from comparable corpora. The extracted parallel segments are typically added to the training corpus as additional training material that is expected to improve the performance of SMT systems, specifically for low-resource language pairs.

The core part of the research presented in this chapter has been previously published in (Pal et al., 2014b, 2015b).

Figure 3.1 schematically represents the research presented and the research questions addressed in this Chapter.

3.1 Introduction

In this chapter, we describe a methodology for extracting English–Bengali parallel resources from comparable corpora using TE and template based phrase extraction. We collected a document-aligned comparable corpus of English–Bengali document pairs from Wikipedia[1]. Wikipedia is a large collection of documents in many different languages. We first collect an English document from Wikipedia and then follow the inter-language link to find the corresponding document in the Bengali Wikipedia. To extract parallel fragments, we perform three steps. In the first step, we cluster the source side of the bilingual comparable corpus into several small groups using TE and a distributional semantic textual similarity method (Mitchell and Lapata, 2010; Grefenstette and Sadrzadeh, 2011; Socher et al., 2012; Agirre et al., 2014; Bentivogli et al., 2016). In the second step, we produce cross-lingual linked clusters of comparable segments for each comparable document using a probabilistic bilingual lexicon. The bilingual lexicon is prepared from a bilingual

[1] `https://www.wikipedia.org/`

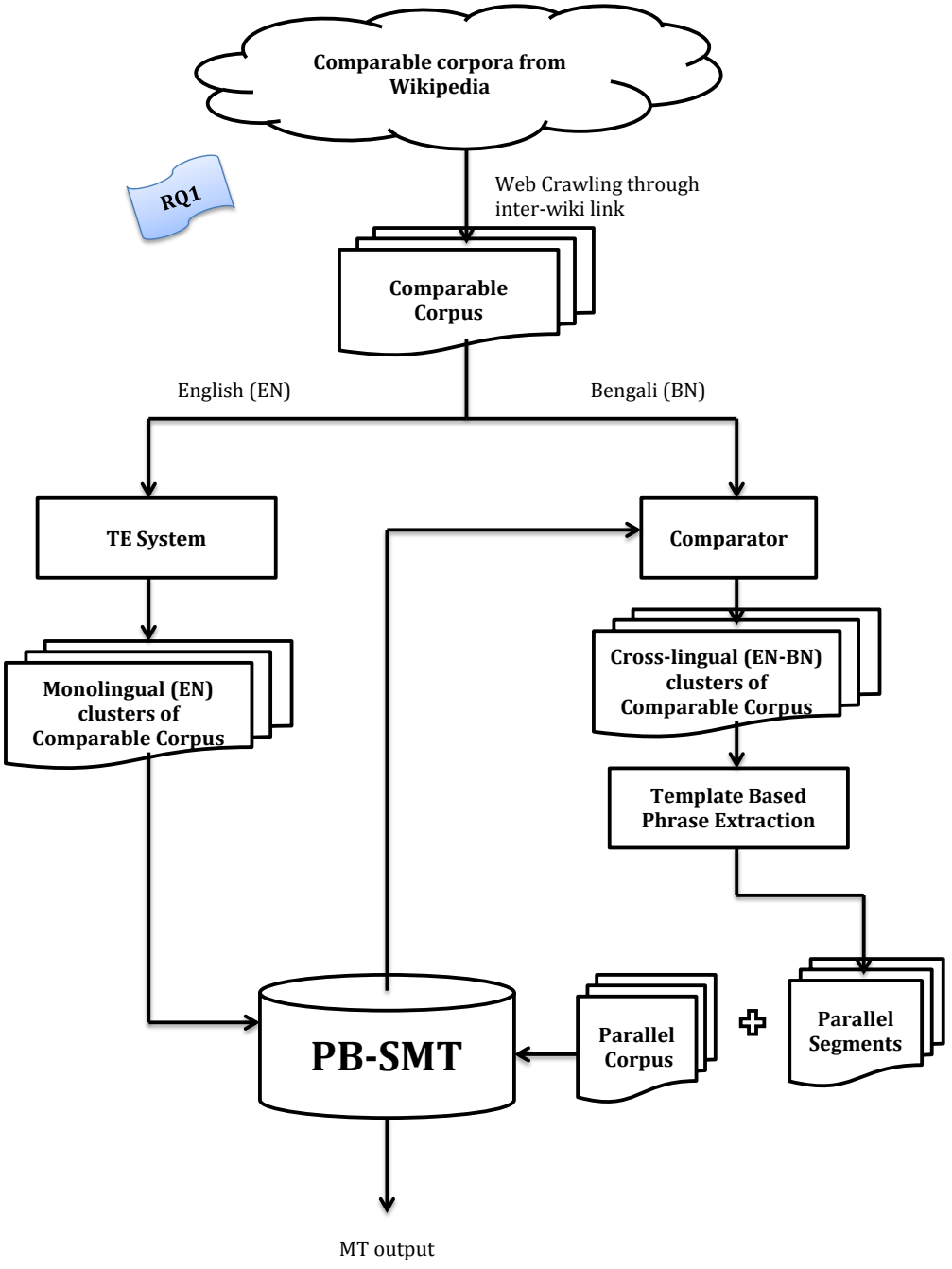


FIGURE 3.1: Schematic design of the research and the research questions presented in this Chapter.

English–Bengali parallel corpus in the tourism domain using a statistical word alignment tool – GIZA++ (Och and Ney, 2003a). In the final step, we use a template-based phrase extraction method (Cicekli and Güvenir, 2001) between each of the aligned groups of comparable segments. The template-based extracted phrases are finally aligned using a baseline phrase-based SMT (PB-SMT) system, which was trained on the English–Bengali tourism parallel corpus.

Typically, there are two approaches that are applied for grouping documents according to their (text) similarity: TE and semantic textual similarity (STS) (Agirre et al., 2014). Given two pieces of text – a text (T) and a hypothesis (H), T is said to entail H if H can be inferred from T (Dagan and Glickman, 2004). The task of TE is to decide whether the meaning of H can be inferred from the meaning of T. For example, let T be: "Eyeing the huge market potential, currently led by Google, Yahoo took over search company Overture Services Inc last year.", and H: "Yahoo bought Overture". For this particular T–H pair, T entails H. STS measures the degree of semantic equivalence between two sentences. This task can be applied in many areas, such as Information Extraction, Question Answering, Summarization, and Information Retrieval, for indexing semantically similar phrases or sentences. STS is related to TE, but differs from TE in that TE is unidirectional while STS is bidirectional. E.g., the two sentences "Yahoo took over search company Overture last year." and "Yahoo acquired Overture. " are highly semantically similar; however, while the first sentence entails the second, it is not true the other way round since the first sentence carries some additional (here temporal) information not contained in the second sentence.

Calculating textual similarity between T and H can be tackled by various techniques at lexical, syntactic, and semantic levels (Šarić et al., 2012; Osman et al., 2012). Lexical techniques are based on word overlap metrics, n-gram matching, or comparing the dependency relations of the two texts. Moreover, some important lexical relationships (e.g., synonyms, hypernyms) can also be applied to measure textual similarities. Other methods, such as syntactic techniques are based on syntactic or dependency trees matching. In addition to STS, another semantic similarity technique was applied based on relations comparison (e.g., logical inference and Semantic Role Labeling).

In distributional semantics approaches (Blei et al., 2003a), similarities between T and H can be computed by measuring their collocation and distributional properties on large

amounts of data in an unsupervised way (Chaney and Blei, 2012) or by using the Gensim framework (Rehurek and Sojka, 2010), in which semantic relationships of words and phrases are computed using the word2vec[2] (Mikolov et al., 2013a) model. Our approach uses Gensim[3] (Řehůřek and Sojka, 2010) to measure distributional semantic similarity. Gensim is a free open source Python library designed to automatically extract semantic topics from documents in an efficient way. Gensim is designed to process raw plain text data (e.g., a corpus). Several popular algorithms such as Latent Semantic Analysis (LSA) (Deerwester et al., 1990), Latent Dirichlet Allocation (LDA) (Blei et al., 2003b) and Random Projections are implemented in Gensim. These algorithms discover semantic structure of documents by examining statistical co-occurrence patterns of the words within a training corpus. LSA and LDA are topic modeling techniques, however LDA is a fully generative model. LSA is also considered as a statistical, corpus-based text comparison method that uses a weighted term-document matrix that is created from a large collection of documents. LSA consists of four steps: (i) preparing a term-document matrix, (ii) a transformation (e.g., tf-idf, log-entropy), (iii) dimensionality reduction using Singular Value Decomposition (SVD) and (iv) retrieval using cosine similarity. LDA assumes that a document is a mixture of latent topics. In contrast to LSA, LDA uses a probabilistic background instead of SVD. In our work, we use a pre-trained Gensim model (cf. Section 3.3) for measuring semantic text similarity.

The rest of the chapter is structured as follows: Section 3.2 discusses previous work relevant to this chapter. Section 3.3 describes the TE system used for our research. Section 3.5 describes comparable text extraction from comparable corpora and Section 3.6 shows how to identify parallel segments from these comparable segments. Section 3.7 and Section 3.8 present the dataset used for our experiments and the baseline experimental setup, respectively. Section 3.9 describes our experiments and presents the evaluation results. Section 3.10 summarizes the outcomes of this research.

3.2 Related Work

Comparable corpora have recently received attention in many research areas in NLP, especially in machine translation. In NLP, there are several applications of comparable

[2] `https://code.google.com/p/word2vec/`
[3] `https://radimrehurek.com/gensim/`

corpora such as the development of bilingual lexicons or terminology databases (Chiao and Zweigenbaum, 2002; Fung and Cheung, 2004), in cross-language information research (Grefenstette, 1998; Chen and Nie, 2000) and in MT (Munteanu and Marcu, 2005; Eisele and Xu, 2010).

Extraction of parallel resources from comparable corpora plays a significant role in MT research. Many approaches have been proposed so far which focus on extracting word or phrase translations from comparable corpora (Fung and McKeown, 1997; Fung and Yee, 1998; Rapp, 1999; Chiao and Zweigenbaum, 2002; Doddington, 2002; Giampiccolo et al., 2007; Saralegui et al., 2008; Gupta et al., 2013). In a majority of these cases, a "seed expressions" list is required to build the context vectors of the words in both the languages. A bilingual dictionary can be used as a "seed word expressions" list. However, most of the strategies follow a standard method such as finding the target words that have the most similar distributions with a given source word based on the context vector similarity measure. Gamallo Otero (2007) prepared a bilingual list of words by using the bilingual correlation method to form a parallel corpus. Instead of a bilingual list, a multilingual thesaurus could also be used for this purpose (Doddington, 2002). Our work shows that comparable corpora containing Wikipedia articles could prove to be beneficial for existing MT systems.

The main objective of the work presented in this chapter is to investigate whether TE can be beneficial to extract parallel text fragments from comparable corpora and whether these parallel text fragments can improve MT system performance. To the best of our knowledge the work presented in this chapter is the first work on the use of textual entailment for parallel segment extraction from comparable corpora. To achieve our goal, we developed a TE system *TESim System* (cf. Section 3.3) which clusters a large comparable document set into smaller groups that accelerate the rest of the processes in the pipeline to extract parallel segments from comparable corpora. Instead of comparing the entire set, the comparison between source–target alignment (cf. Section 3.5.3) and template-based phrase extraction (cf. Section 3.6.1) are performed within the clusters.

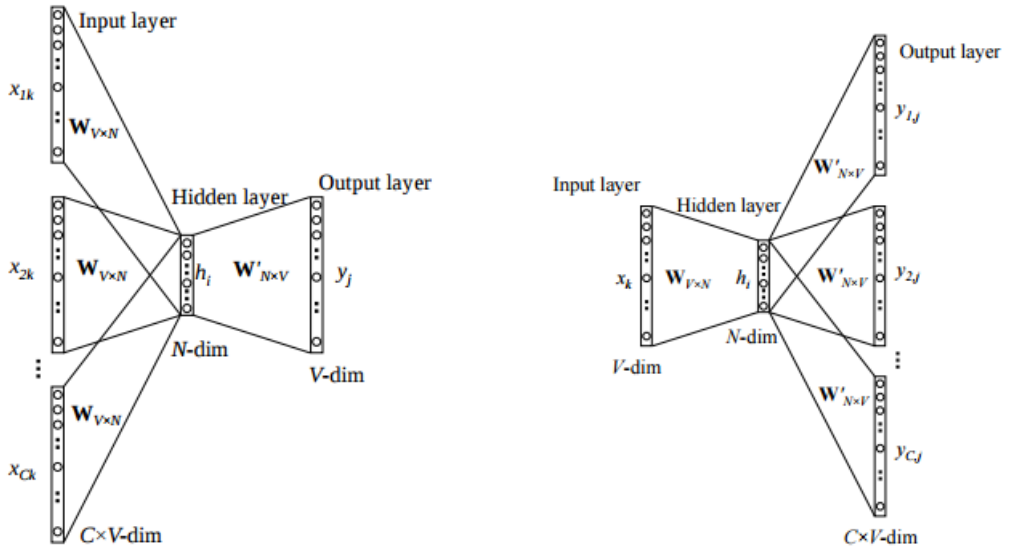
3.3 The TESim System

Our Textual Similarity (TESim) system architecture is shown in Figure 3.3. A detailed description can be found in Pakray (2013). The TESim system contains Semantic Textual Similarity (STS) and TE modules. STS measures the degree of semantic equivalence between two sentences. Our STS model follows monolingual STS approaches. Recently distributed representations of words such as word2vec (Mikolov et al., 2013a) have performed particularly well in STS tasks. word2vec provides state-of-the-art performance in several types of similarity and analogy tasks (Mikolov et al., 2013b; Pennington et al., 2014) and also delivers significant efficiency in training. Word2vec is a computationally efficient predictive model for learning word embeddings from raw text. The word2vec model can be used in 2 ways – to predict a target word given the context (continuous bag of words (CBOW)), or to predict the target context given a word (skip-gram) in an unsupervised way. The CBOW model takes the average of the vectors of the 1-hot encoded vectors of the input context words as shown in Figure 3.2a. Our STS module follows the skip-gram model. The skip-gram model (cf. Figure 3.2b) consists of an input layer of 1-hot encoded vector with V-dimensions and an output layer with $C \times V$-dimensional one-hot encoded word vectors where C is the number of total words predicted by the input word. A weight matrix of $V \times N$-dimension is multiplied with the input vector producing an N-dimensional hidden layer. Each output word is obtained by multiplying the hidden layer and the weight matrix associated with the predicted words.

The word2vec based STS model was trained on the Google News corpus and Wikipedia corpus. The STS module was pre-trained with word and phrase vectors available as part of the Google News dataset (Mikolov et al., 2013a) which consists of about 100 billion words. The STS module used a latent semantic analysis (LSA) word-vector mappings model which contains 300-dimensional vectors for three million words and phrases. Additionally, we built word and phrase vectors from Wikipedia articles for both Bengali and English language data. We generated 300-dimensional word and phrase vectors from Wikipedia articles using the word2vec tool. To build word2vec for learning high-quality word vectors we use Gensim[4] – a Python framework for vector space modeling. Gensim provides an efficient implementation of the word2vec model[5]. The word2vec model computes cosine

[4] `https://radimrehurek.com/gensim/`

[5] `https://radimrehurek.com/gensim/models/word2vec.html`



(A) Continuous bag-of-words model. (B) Skip-gram model.

FIGURE 3.2: Word2vec models: The vocabulary for learning word vectors consists of V-dim words and N is the dimension of the hidden layer. The input to hidden layer connections are represented by matrix $\mathbf{W} \in R^{V \times N}$, where each row represents a vocabulary word. Similarly, hidden layer to output layer connections are described by matrix $\mathbf{W}' \in R^{N \times V}$. C is the number of words in the context. These figures are borrowed from the tutorials in Chris McCormick' Blog (`http://mccormickml.com/tutorials/`) and these models are originally reported in Mikolov et al. (2010)

.

similarity between two vectors. These vectors represent two candidate texts T and H. We evaluated our STS module using SemEval[6] data.

An example for vector representations "in action" could be as follows: vector ("King") − vector ("Man") + vector ("Woman") results in a vector that is closest to the vector representation of the word "Queen."

Our TE recognition system consists of various components: a lexical component, a syntactic component, a semantic component, a Support Vector Machine (SVM) module, and an Entailment Decision module (cf. Section 3.4). The system is a combination of these different components working on various lexical knowledge sources (WordNet, Wikipedia), lexical distance, syntactic similarity, and semantic similarity. The system computes the entailment decision using the outcome from each of these components. Our TE system is trained on Recognition Textual Entailment (RTE)[7] datasets.

[6] `https://en.wikipedia.org/wiki/SemEval`
[7] `http://pascallin.ecs.soton.ac.uk/Challenges/RTE/`

Algorithm 1: Calculation of the score

Input: Text Pairs (t1, t2)

Output: Final Score

CS ← `Calculate` STS Score of (t1, t2) by using Cosine Similarity;

if CS > 0.7 **then**

    Final Score ← `Calculate` TE Score of (t1, t2) using TE system;

else

    Final Score ← CS score of (t1, t2);

Initially, TESim takes text pairs (t_1, t_2) and calculates cosine similarity between the word2vec (vector) representation between two texts. We set a threshold on the cosine similarity value (0.7), and if the similarity measure is greater than the threshold value, a similarity score between t_1 and t_2 is generated by the TE system (cf. Algorithm 1).

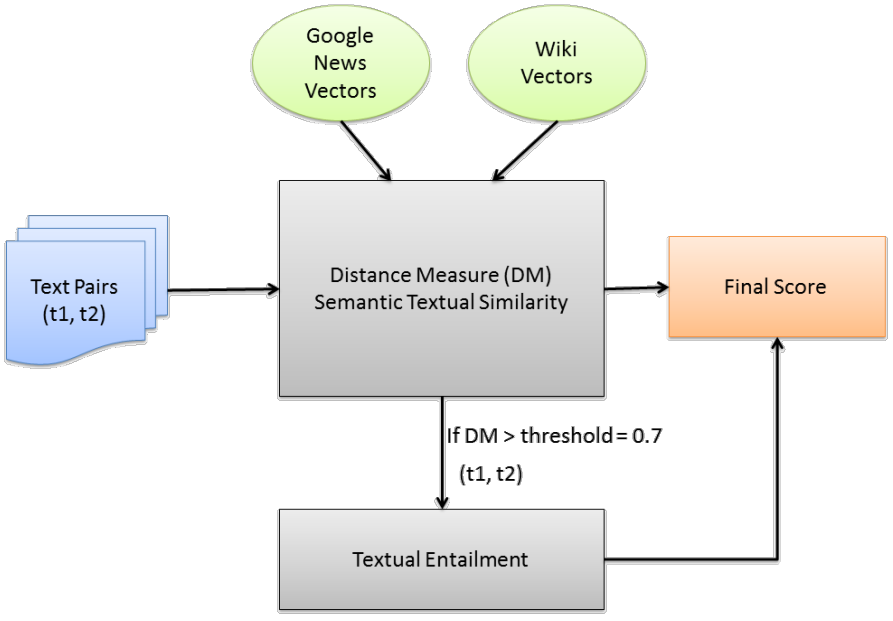


FIGURE 3.3: System Architecture

The reason behind the use of both STS and TE techniques for TESim is that, sometimes STS provides high similarity scores between T and H, even when they are not similar or entailed. Consider the case of Id 1, in Table 3.1. The STS module cannot detect the negation and provides a semantic similarity score of 1.0. However, the entailment score is 0.22776. Therefore, the system can easily conclude that Text 1 and Text 2 (cf. Id 1) are not similar texts. This technique helps to remove negation sentences from the cluster

Id	Text 1 (H)	Text 2 (T)	Semantic similarity score	Entailment
1	Clinton's new book is not a big seller here.	Clinton's book is a big seller.	1.0	0.22776
2	Vodafone's share of net new subscribers in Japan has dwindled in recent months.	There have been many new subscribers to Vodafone in Japan in the past few months.	0.8059	0.2944

TABLE 3.1: Examples of text pairs and entailment results

(cf. Section 3.5.2) which means negative sentences will form a different cluster. This can help MT to better translate negative sentences.

3.4 A Two-way TE System

A two-way automatic TE recognition system is integrated into the TESim system (see Section 3.3). The TE system uses a support vector machine (SVM) which is trained on lexical, syntactic and semantic features between T and H. We use a total of thirty features to train our TE model. The system architecture is shown in Figure 3.4. The entailment engine contains four modules: lexical, syntactic, reVerb and semantic. Thirty features are extracted from T and H after preprocessing (the lexical module produces eighteen features, the syntactic module provides ten features, one feature from reVerb and one feature from the semantic module). We used TE features described in (Pakray, 2013). The work is based on the joint research publication (Pakray et al., 2010a,b).

3.4.1 Lexical Module

This module performs six different types of lexical comparison and twelve types of lexical similarity comparison between T and H. The six lexical comparisons are:

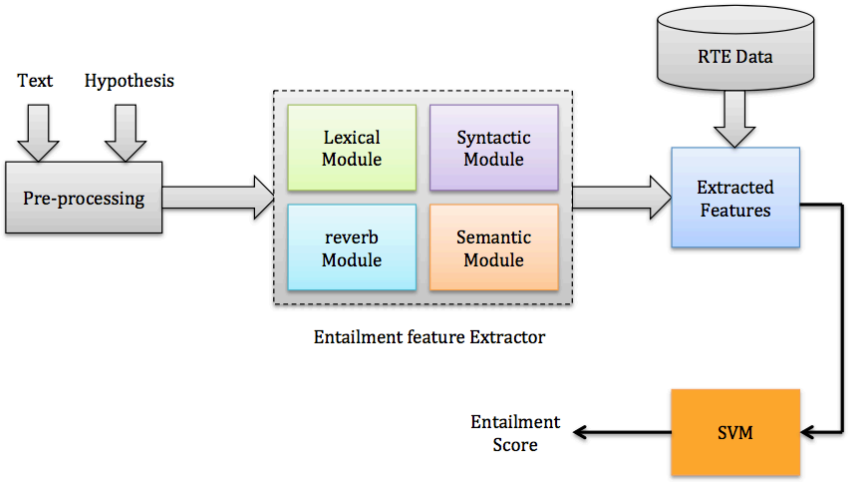


FIGURE 3.4: Two way TE architecture

- **Unigram match:** This features simply measures the fraction of unigrams in H that match with unigrams in T. While matching, we also consider WordNet[8] synonyms as a match.

$$feature\_weight_{uni} = \frac{count(words \in T \cap H)}{count(words \in H)}$$

- **Bigram match:** Bigrams are extracted from T and H. This feature measures the fraction of bigrams in H that match with bigrams in T. Like a unigram matching, we also use WordNet synonym matches for individual words in bigrams.

$$feature\_weight_{bi} = \frac{count(bigrams \in T \cap H)}{count(bigrams \in H)}$$

- **Longest common sub-sequence (LCS):** $LCS(T, H)$ measures the similarity between T and H in terms of the length of the LCS of T and H. Like in case of unigrams and bigrams, we consider WordNet synonyms as matches.

- **skip-gram:** A skip-gram[9] is defined as any combination of n words in the order as they appear in a sentence, allowing arbitrary gaps. We considered only $1\_skip\_bigrams$ i.e., one word gap between two words in a sentence. The skip-gram based feature weight is calculated as

[8] `https://wordnet.princeton.edu/`
[9] These are surface skip-grams, not skip-grams as in the word2vec model.

$$feature\_weight_{skip} = \frac{count(skip\_gram \in T \cap H)}{n}$$

where $skip\_gram(T, H)$ refers to the number of common $1\_skip\_bigrams$ (pair of words in sentence order with one word gap) found in T and H and n is the number of $1\_skip\_bigrams$ in the hypothesis H. Feature weights are calculated on surface form only.

- **Stemming:** Each word in a T–H pair is stemmed and the feature value is calculated as the fraction of stems in H that match with stems in T.

$$feature\_weight_{stem} = \frac{count(stemmed\_unigrams \in T \cap H)}{count\_unigrams(H)}$$

- **Named entity (NE) matching[10]:**

$$feature\_weight_{NE} = \frac{count(NE \in T \cap H)}{count(NE \in H)}$$

Twelve types of lexical similarity comparisons between T and H are measured using Vector Space Measures (Euclidean distance, Block distance, Minkowsky distance, Cosine similarity and Matching Coefficient), Set-based Similarities (Dice, Jaccard, Overlap, Harmonic) and Edit Distance Measures (Levenshtein distance, Smith-Waterman distance, Jaro distance).

3.4.2 The Syntactic Module

The syntactic module compares the dependency relations between both H and T. The system extracts syntactic structures from the T–H pairs using a Combinatory Categorial Grammar (CCG) Parser[11] (Steedman, 2000; Clark et al., 2002) and the Stanford Parser[12] (de Marneffe and Manning, 2008) and compares the corresponding structures. Two different systems have been implemented: one system is based on the Stanford Parser output while the other operates on the CCG Parser. The system accepts pairs of text snippets (T and H) as input and produces a score for every comparison. Some of the important comparisons based on the dependency structures between T and H are:

[10]For NE detection we used LT-TTT2 Toolkit: `http://www.ltg.ed.ac.uk/software/lt-ttt2`
[11]`http://svn.ask.it.usyd.edu.au/trac/candc/`
[12]`http://nlp.stanford.edu/software/lex-parser.shtml`

1. **Subject-Verb Comparison:** Subject and verb are identified from the both H and T by analyzing their word dependency relations. The feature weights are calculated by comparing the subject and verb between T and H.

2. **WordNet Based Subject-Verb Comparison:** In this case, the feature weights are calculated by considering wordnet synonyms during the comparison between T and H.

3. **Object-Verb Comparison:** Objects and verbs that are identified through *dobj* dependency relation. Feature weights are calculated similarity to the feature weight described in 1.

4. **WordNet Based Object-Verb Comparison:** Feature weights are calculated similar to the feature weight described in 2.

Other syntactic similarity features include cross subject-object comparison, number comparison, noun comparison, prepositional phrase comparison, determiner comparison and comparisons for other word dependency relations.

3.4.3 reVerb Module

ReVerb[13] (Fader et al., 2011) is a tool which provides binary relationships of an English sentence. The extraction format is shown in Example 3.1.

Example 3.1. *Extraction Format: A person is playing a guitar*
reVerb Extracts: *arg1= {A person}; rel = {is playing}; arg2 = {a guitar};*

The system parses T and H using the reverb tool. We calculate scores by comparing the relations between T and H.

3.4.4 Semantic Module

The semantic module is based on the Universal Networking Language (UNL) (Uchida et al., 2012). UNL can express information or knowledge in semantic network form with hyper-nodes. UNL is like a natural language for computers to represent and process

[13] `http://reverb.cs.washington.edu/`

human knowledge. There are two modules in the UNL system: En-converter and De-converter. The process of representing natural language sentences in UNL graphs is called En-converting and the process of generating natural language sentences from UNL graphs is called De-converting. An En-converter is a language independent parser, which provides a framework for morphological, syntactic, and semantic analysis synchronously. The En-Converter is based on a word dictionary and a set of en-conversion grammar rules. It analyses sentences according to the en-conversion rules. A De-converter is a language independent generator, which provides a framework for syntactic and morphological generation synchronously. An example UNL relation for a sentence "Pfizer is accused of murdering 11 children" is shown in Example 3.2.

Example 3.2. *Pfizer is accused of murdering 11 children*
{org:en} Pfizer is accused of murdering 11 children{/org}
{unl}
***obj**(accuse(icl>do,equ>charge,cob>abstract_thing,agt>person,obj>person)*
.@entry .@present,pfizer.@topic)
***qua:**01(child(icl>juvenile>thing).@pl,11)*
***obj:**01(murder(icl>kill>do,agt>thing,obj>living_thing)*
.@entry,child(icl>juvenile >thing).@pl)
***cob**(accuse(icl>do,equ>charge,cob>abstract_thing,agt>person,obj>person)*
.@entr y.@present,:01)
{/unl}

3.4.5 Support Vector Machines (SVM)

SVMs[14] (Vapnik, 1995; Cortes and Vapnik, 1995) are supervised learning models used for classification and regression analysis. The basic SVM takes a set of training examples (e.g., feature vectors with binary values) as input data and predicts, for each given input, the possible class from a given set of classes using a classification function. We used the RTE-1, RTE-2, RTE-3 and RTE-4 datasets to build the SVM based TE models. Each of these RTE datasets consists of manually annotated ('YES' and 'NO' TE decision) data for every T–H pairs. The released version of RTE datasets contains development and test

[14] `http://en.wikipedia.org/wiki/Support_vector_machine`

set data. Therefore, to make a single model trained on all available data, we combined all these RTE development as well as test set data together to prepare training data for our SVM based TE system. Table 3.2 shows the statistics of the RTE datasets and overall 5567 T-H pairs were used as the training data for the SVM model.

RTE data		**Entailment decision**	**T-H Pairs**	**Overall**
RTE-1	Development Set 1	Yes	143	287
		No	144	
	Development Set 2	Yes	140	280
		No	140	
	Test Set	Yes	400	800
		No	400	
RTE-2	Development Set	Yes	400	800
		No	400	
	Test Set	Yes	400	800
		No	400	
RTE-3	Development Set	Yes	412	800
		No	388	
	Test Set	Yes	410	800
		No	390	
RTE-4	Test Set	Yes	500	1000
		No	500	
Total				**5567**

TABLE 3.2: RTE-data statistics used for training our SVM based TE system

The SVM classifier deals with the two-way classification (e.g., 'Yes' or 'No' i.e., entailment or not) task. We used LIBSVM[15] (Chang and Lin, 2011) – a library for SVMs for the classifier to learn from these data sets. As shown in Table 3.2, the total size of the training data is small, therefore we set default hyper-parameter settings for these classification task.

[15] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

3.5 Comparable Text Extraction from Comparable Corpora

3.5.1 Comparable Corpora Collection

Wikipedia is a huge collection of articles on large varieties of topics in may domains in a wide range of languages. Wikipedia links articles on the same topic in different languages using the "interwiki" linking facility. Thus, document alignment for multilingual documents on similar topics is already provided in Wikipedia.

To collect comparable corpora for English–Bengali document pairs, a crawler was designed[16]. The crawler operates on an initial "seed keyword list". In our work we focus on the tourism domain. The "seed keyword list" is mainly a named entity (NE) list, collected from the English tourism domain corpus using the Stanford NE Recognizer[17]. The crawler first visits each English page of Wikipedia, saves the raw text (in HTML format), and then follows the cross-lingual link for each English page and collects the corresponding Bengali page. We keep only the textual information and all the other details are discarded. We extract English and Bengali sentences from each document; however, there is not a one-to-one correspondence between the English and Bengali sentences. Moreover, often Bengali documents contain limited information compared to the corresponding English documents.

3.5.2 Monolingual Clustering

The TESim system compares every sentence of a document with every other sentence in the same document and provides an entailment score for each sentence pair. Thus, $n \times (n-1)$ comparisons are made for a document containing n sentences. The TE system operates on the monolingual data. A cut-off (above 0.7) entailment score was considered for grouping entailed sentences into the same cluster. The TE system divides the source side of the complete set of comparable documents into a smaller sets of clusters. Each cluster contains at least two sentences. Since the TESim system operates on monolingual English data, only the English side of the comparable corpora contains sets of clusters. To extract parallel fragments from comparable data, we assign a comparable set of Bengali

[16]The crawl of Wikipedia was made in November, 2013.
[17]`http://nlp.stanford.edu/software/CRF-NER.shtml`

sentences (the Bengali cluster) corresponding to each English cluster which is detailed in Section 3.5.3.

3.5.3 Cross-lingual Linked Clusters

To establish cross-lingual linked clusters between the English and Bengali comparable documents, we use a probabilistic bilingual English-Bengali lexicon (lex_{e2f}). The lexicon was prepared by using a statistical word alignment tool GIZA++[18] (Och and Ney, 2003a; Brown et al., 1993). We established the word alignments on a bilingual parallel English–Bengali corpus (see Section 3.7) in the tourism domain[19] with GIZA++ which produces a probabilistic bilingual word alignment list. We retain only five most probable target words with respect to the source words in the bilingual lexicon. Initially, we consider each English cluster as a bag-of-words (BOW) and translate each word in BOW using lex_{e2f}. The translated BOW clusters are then passed as a query to the indexed Bengali comparable documents and retrieve a set of top ten ranked relevant Bengali sentences. Each Bengali sentence for each document is indexed using Lucene[20]. We use the *OR Boolean retrieval model*[21] to retrieve related sentences from the comparable document. In this way, we established cross-lingual linked clusters of English–Bengali comparable documents.

3.6 Alignment of Parallel Text Fragments

We then extract bilingual phrases from comparable cross-linked clusters using a template based phrase extraction method (cf. Section 3.6.1). Although the template based approach works well in the case of parallel corpora, it can also be applied to comparable corpora. In this case, the template based method can only align *atomic*[22] translations (cf.

[18] `http://code.google.com/p/giza-pp`

[19] The corpus was collected from the consortia-mode project "Development of English to Indian Language Machine Translation (EILMT) System", the EILMT project is funded by the Department of Electronics and Information Technology (DEITY), Ministry of Communications and Information Technology (MCIT), Government of India.

[20] `https://lucene.apache.org/`

[21] The details regarding this method can be found in `http://nlp.stanford.edu/IR-book/pdf/01bool.pdf`

[22] An *atomic* translation template T_{src} and T_{tgt} between languages L_{src} and L_{tgt} means that the strings T_{src} and T_{tgt} correspond to each other. The variables in translation templates T_{src} are $X_1 ... X_n$, and the variables in translation templates T_{tgt} are $Y_1 ... Y_n$, where $n > 1$ i.e., $T_{src} \leftrightarrow T_{tgt}$ if $X_1 \leftrightarrow Y_1$ *and... and* $X_n \leftrightarrow Y_n$.

Section 3.6.1). Other phrases such as corresponding match sequences (CMS) extracted by the template-based method are aligned by a baseline PB-SMT system (see Section 4.8.3) trained on our tourism domain parallel corpus. This is the same machine translation system whose performance we want to improve. We also performed the same task in the other direction, i.e., Bengali–English.

For each of the cross-linked clusters, each translated source phrase (CMS_{src}) (translated from English–Bengali or Bengali–English) is compared with all the target phrases (CMS_{tgt}) extracted from the corresponding target cluster.

3.6.1 Template-based Phrase Extraction

We extract phrase pairs based on the EBMT work described in (Cicekli and Güvenir, 2001). They automatically extract translation templates from sentence-aligned bilingual text by observing the similarities and differences between two example pairs. Their approach produces two types of translation templates, generalized and atomic translation templates. A generalized translation template replaces similar CMS or differing sequences (Corresponding Difference Pair (CDP)) with variables while an atomic translation template does not contain any variables. We extract the atomic translation templates from the cross-linked clusters extracted from our comparable documents and add them as additional phrase pairs to our PB-SMT system. Consider the following two English–Bengali translation pairs from the tourism domain data:

(1) a. visitors feel happiness: *darsakera ananda onuvab kore*
    b. visitors feel restlessness: *darsakera klanti onuvab kore*

These two examples share the word sequence "visitors feel" (CMS) and differ in the word sequence "happiness" and "restlessness" (CDS) on the source side. Similarly, on the target side, the differing fragments are "ananda" and "klanti". Based on these differing fragments, we extract the following sub-sentential phrase pairs as in (2).

(2) a. happiness: *ananda*
    b. restlessness: *klanti*

We apply this process recursively to extract sub-sentential phrase pairs when more than one differing sequence is present between a pair of sentences by looking for further evidence

within the source–target language cluster pairs. The details of the algorithm can be found in (Cicekli and Guvenir, 2001).

This particular approach has a cubic runtime complexity with respect to the number of sentences in the bilingual corpus. It takes a significant amount of time to extract phrase pairs even from a small corpus. Therefore, we used heuristics to reduce the processing time. We grouped the entire comparable corpus into a number of cross-linked clusters based on the technique described in Sections 3.5.2 and 3.5.3 so that similar or entailed sentences belong to the same cluster. We extract atomic translation pairs from each of these clusters. Since the parallel atomic translations are extracted from comparable sentences, they are noisy. To remove noisy atomic translations, we need to validate whether these phrase translations are correct or not. These atomic translations as well as the alignment of CMS which are also extracted from the comparable sentences are validated by using English–Bengali Machine translation. An English–Bengali baseline PB-SMT system was developed which was trained on our tourism domain parallel corpus. We translated the English CMS into Bengali. Each translated source CMS (translated from English to Bengali) is compared with the corresponding target CMS (CMS_{tgt}) extracted from the comparable sentences of the corresponding target-side cross-link cluster. When a translated CMS is considered, we compare each of its tokens to each token in the CMS_{tgt}. We performed the comparison using the Minimum Edit Distance Ratio and Longest Common Subsequence Ratio methods.

3.7 Dataset

In our experiment, we use an English-Bengali parallel corpus containing 23,492 parallel sentences comprising of total 488,026 word tokens from the travel and tourism domain[23]. We randomly selected 500 sentences each for the development set and the test set from the initial parallel corpus. The rest of the sentences were used as the training corpus. The training corpus was filtered with a maximum allowable sentence length of 100 words and sentence length ratio of 1:2 (either way).

[23]The corpus was collected from the "Development of English to Indian Languages Machine Translation (EILMT) System" project funded by the Department of Electronics and Information Technology (DEITY), Ministry of Communications and Information Technology (MCIT), Government of India.

3.8 System Setup

The effectiveness of the parallel phrase pairs extracted from the comparable corpus was tested by using the standard log-linear PB-SMT model as our baseline system. For building the baseline PB-SMT system, we used the Moses toolkit[24] with a maximum phrase length of seven and a 5-gram language model. The other experimental settings were the GIZA++ implementation of IBM word alignment model 4 with grow-diagonal-final-and heuristics for performing word alignment and phrase-extraction (Koehn et al., 2003). The reordering model was msd-bidirectional (i.e., using both forward and backward models) and conditioned on both source and target languages. The reordering model was built by calculating the probabilities of the phrase pairs associated with the given orientations monotone (m), swap (s), and discontinuous (d). We used Minimum Error Rate Training (MERT) (Och, 2003). We also set up a hierarchical (Galley and Manning, 2008) reordering model for our experiment on a held-out development set of 500 sentences, and the target language model was built with Kneser-Ney smoothing (Kneser and Ney, 1995) trained with SRILM (Stolcke, 2002) using the Bengali side of the bilingual tourism data.

3.9 Experiments and Results

Our experiments were carried out in two directions. First, we improved the baseline model using the aligned parallel fragments extracted from our comparable corpora.

	Total (English)		Total (Bengali)	
Comparable corpora	579,037	Sentences	169,978	Sentences
Extracted comparable sentence pairs	4,723	Sentences	4,723	Sentences
Aligned parallel phrases	6,937	Phrases	6,937	Phrases

Table 3.3: Statistics of the Comparable Corpus

The collected comparable corpus consisted of 6,825 English–Bengali document pairs. It is evident from Table 3.3 that English documents are more informative than the Bengali documents, as the sentences in English documents outnumber the sentences in the Bengali documents. The TESim system was able to establish cross-lingual entailment for 4,723 English–Bengali comparable sentence pairs by means of cross-lingual links using

[24] `http://statmt.org/moses/` (Koehn et al., 2007)

GIZA++ (cf. Section 3.5.3). When the Bengali phrases were input to the Bengali–English translation module, some of them could not be translated into English and some of them could be translated only partially. Therefore, some of the tokens were translated while some were not. Untranslated and partially translated phrases were discarded. Manual inspection of the parallel list revealed that most of the aligned texts were of good quality.

The MT evaluation results are reported in Table 3.4. The evaluation was carried out using established automatic MT evaluation metrics: BLEU (Papineni et al., 2002), NIST (Doddington, 2002), METEOR (Lavie and Agarwal, 2007), and TER (Snover et al., 2006a).

Exp.	Experiments	BLEU	NIST	METEOR	TER
1	Baseline (B1) with lexical reordering	10.92	4.16	0.3073	75.34
2	Baseline (B2) with hierarchical reordering	11.04	4.20	0.3101	75.01
3	B1 + Extracted Parallel Phrases	13.98	4.45	0.3401	72.03
4	B2 + Extracted Parallel Phrases	14.12	4.49	0.3492	71.53

TABLE 3.4: Evaluation results; all scores are statistically significant over baseline systems.

Table 3.4 shows the performance of the PB-SMT systems built on the initial parallel training corpus and the larger training corpus containing the parallel phrases extracted from the comparable corpora. There are two different baseline settings based on the reordering models which have been developed: the baseline with lexical reordering model (B1, see Experiment 1 in Table 3.4) and with the hierarchical reordering model (B2, see Experiment 2). Treating the parallel phrases extracted from the comparable corpora as additional training material (cf. Experiment 3) results in a significant improvement in terms of BLEU (3.06 points, 28.02% relative) over the baseline (B1) system. Similar improvements are also obtained for the other metrics. It is to be noted that, when we performed a similar experiment (i.e., Experiment 4 is like Experiment 3) with the hierarchical reordering model, the data augmented system provided further improvement. This experiment reduced TER scores considerably compared to the other experiment. Reducing TER score signifies that the output of the MT system would be more fluent. In terms of BLEU, our system also performs better (3.2 points, 29.30% relative) over the baseline system.

Treating the parallel phrases extracted from the comparable corpora as additional training material results in significant improvement in terms of BLEU (3.06 points, 32.97% relative) over the baseline system. Similar improvements are also obtained for the other metrics.

3.10 Conclusions and Future Work

We presented a methodology that uses a textual entailment technique for extraction of parallel phrases from comparable corpora. For low-resource language pairs, this approach can be useful to improve the quality of MT systems. The overall low evaluation scores (BLEU in the range of 10.92–14.12) obtained in our experiments can be attributed to the fact that Bengali is a morphologically rich language and has a relatively free word order; besides, we had only one set of reference translations for the test set. Manual inspection of a subset of the output generated by our system reveals that the additional training examples extracted from comparable corpora effectively resulted in better lexical choices and fewer out-of-vocabulary cases in comparison with the baseline PB-SMT output. One of the outcomes of this experiment is the improvement over the baseline PB-SMT for a low-resource language pair after adding parallel phrases extracted from comparable corpora, which answers RQ1 i.e., improving MT for low resource languages. Another interesting outcome of this experiment is the usability of the TE method within comparable corpora research which is a novel contribution (Pal et al., 2014b, 2015b).

In future, we will explore parallel phrase extraction techniques from comparable corpora by combining a TE system with hybrid word alignments or hybrid MT methods. We also plan to investigate whether this approach can bring about improvements of comparable magnitude settings where larger parallel training data is available.

Chapter 4

Hybrid Machine Translation

This chapter describes different strategies that we applied to build an efficient hybrid pipeline for state-of-the-art statistical machine translation (SMT) and forest to string based SMT. This chapter mainly addresses **RQ2:** *How can SMT better profit from the existing training data?* and **RQ3:** *What could improved hybrid implementations of MT be like?* by focusing on the following improvements:

1. Pre-ordering
2. Effective preprocessing and use of explicitly aligned bilingual terminology. e.g., named entities (NEs) and multiword expressions (MWEs).
3. Hybrid word alignment
4. A simple but effective hybridization technique to combine multiple knowledge sources

Reordering poses a big challenge in SMT between distant language pairs. This chapter presents how reordering between distant language pairs can be handled efficiently in phrase-based statistical machine translation. We approach the problem of reordering between distant languages with prior reordering of the source text at chunk level to simulate the target language ordering. Prior reordering of source chunks is performed by following the target language word order suggested by word alignment. The test set is reordered using monolingual MT trained on the source and the reordered source. Our approach of prior reordering of the source chunks is compared with pre-ordering of source words based on word alignments and the traditional approach of prior source reordering based

on language-pair specific reordering rules. The effects of these reordering approaches is studied on an English–Bengali translation task, a language pair with different word orders. From the experimental results we find that word alignment based reordering of the source chunks is more effective than the other reordering approaches, and that it produces statistically significant improvements over the baseline system on BLEU. On manual inspection we find that this reordering approach results in significant improvements in terms of word alignments.

Our hybrid system improves over the baseline SMT performance by incorporating additional knowledge sources such as extracted bilingual NEs, MWEs, translation memories and phrase pairs induced from example-based methods. We report the performance of hybrid systems in terms of the results of a confusion network-based system combination that combines the best performance of each individual system within a multi-engine pipeline. Core parts of the research presented in this chapter have been published in (Pal et al., 2014c,a, 2015a, 2016a).

The research presented and the research questions addressed in this Chapter are schematically represented in Figure 4.1.

4.1 Introduction

Recently, corpus-based MT has delivered increasingly better translations. Different corpus-based MT approaches have been proposed in the last few decades such as Translation Memory (TM) (Kay, 1997), Example-based Machine Translation (EBMT) (Carl and Way, 2003) and Statistical Machine Translation (SMT) (Koehn, 2010). Out of these, in terms of large-scale evaluations, SMT has arguably been (until recently) the most successful and efficient MT paradigm[1]. The quality of SMT mainly relies on good quality word alignment as well as good phrase pair estimation, both of which can be achieved by using large amounts of sentence-aligned parallel corpora. However, SMT for low-resource language pairs usually produces inferior quality translation.

One of the dominating approaches in SMT is Phrase-Based SMT (PB-SMT) in which the best translation $e = e_1 \dots e_i \dots e_I$ for a source sentence $f = f_1 \dots . f_j \dots . f_J$ (containing I

[1]WMT 2016 has been the first large scale shared task in which NMT outperformed the SMT based approaches.

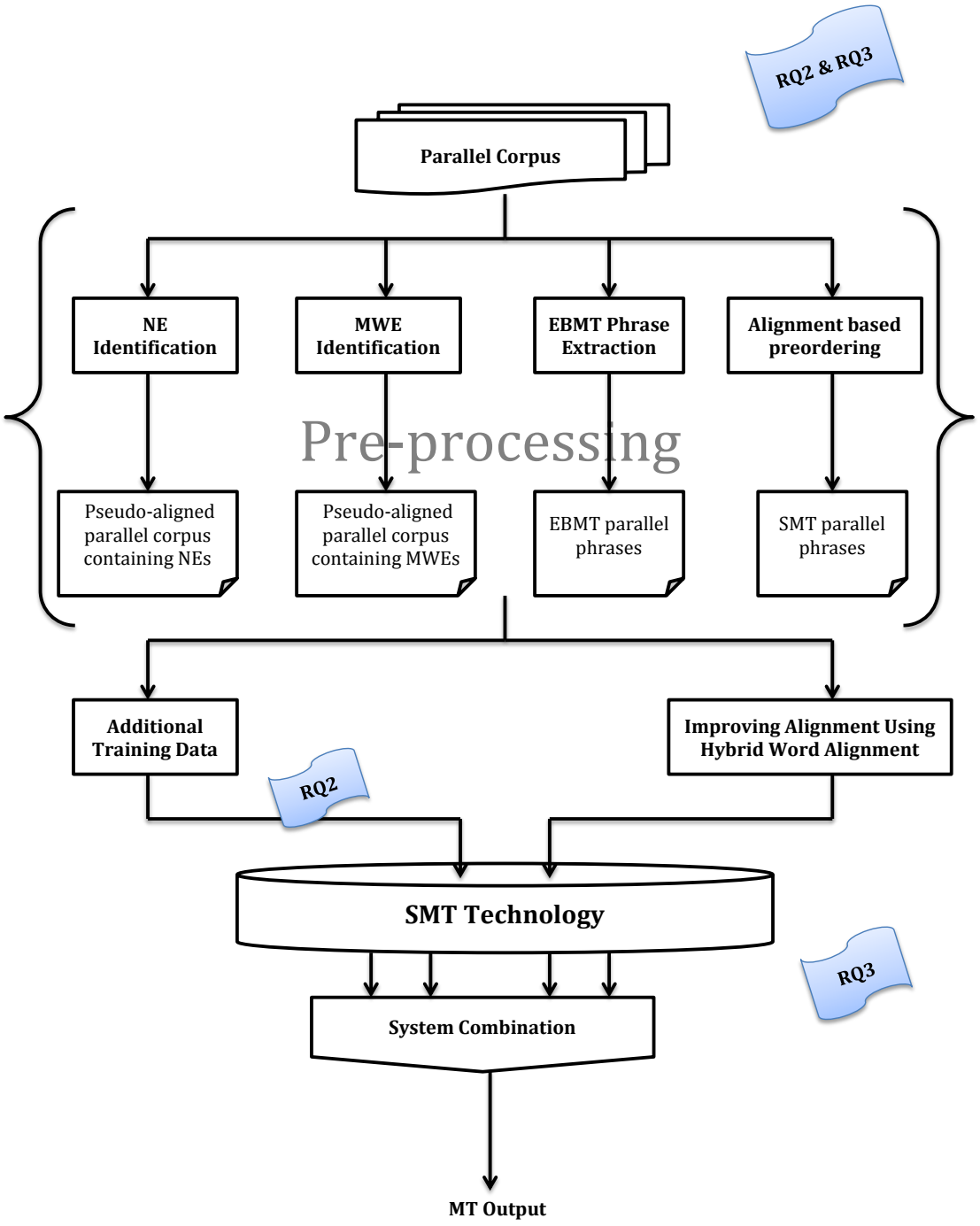


FIGURE 4.1: Schematic design of the research and the research questions presented in this Chapter.

target and J source phrases respectively) is selected to maximize Equation 4.1

$$argmax_{I,e_1^I} P(e_1^I|f_1^J) = argmax_{I,e_1^I} P(f_1^J|e_1^I) \times P(e_1^I) \tag{4.1}$$

where $P(f_1^J|e_1^I)$ and $P(e_1^I)$ denote respectively the translation model and the target language model (Brown et al., 1993). To achieve high quality translation, PB-SMT must ensure two major factors: good quality word alignment and good coverage of the phrase translation candidates in the phrase table. In state-of-the-art PB-SMT systems, these two components are estimated from large sentence aligned parallel corpora. To achieve better estimation, data pre-processing plays a crucial part in any data-driven/corpus-based approach. Effective pre-processing of data in the form of explicit alignment of bilingual terminology (e.g. MWEs and NEs) can provide more productive and functional MT systems. MWEs and NEs offer challenges within a language. The proper way of handling MWEs in the context of estimating phrase translation probabilities, phrase extraction methodologies and even the generation phase of SMT, is extremely challenging because of the idiosyncratic nature of MWEs (Sag et al., 2002). Examples of MWEs include compound nouns (*"building complex"*), phrasal prepositions (*"according to"*), conjunctions (*"as well as"*), idioms (*"kick the bucket"* means *"to die"*), phrasal verbs (*"find out"*), verb-object combinations involving light or support verb constructions (*"make a mistake"*), etc. Named entities on the other hand often consist of more than one word; therefore, they can also be considered as a specific type of MWEs such as noun compounds (Jackendoff, 1997).

Traditional approaches to word alignment following IBM Models (Brown et al., 1993) do not work well with MWEs because the structure and meaning of MWEs can not always be derived from their component words when they appear independently. Most of the South Asian languages, especially, Indian languages like Bengali and Hindi, are morphologically rich. To express predicates these languages often use their morphological inventories in terms of complex predicates. In Bengali, complex predicate (CPs) patterns are made up of verb + verb (**compound verbs:** e.g., বলতে লাগলো (*bolte laglo*) "started saying", মেরে ফেলা (*mere phela)* "kill") or noun/adjective/adverb + verb (**conjunct verbs:** e.g., ভরসা করা (*bharsha kara*) "to depend", ঝকঝক করা (*jhakjhak kara*) "to glow"). The first verb in a compound verb is represented either in conjunctive participial form *"-এ (-e)"* or the infinitive form *"-তে (-te)"* at the surface level which is called a Full Verb. The other

verb bears inflection for Tense, Aspect and Person, and is referred to as the Light Verb. On the other hand, each Bengali conjunct verb consists of an adjective, adverb or noun followed by a light verb. These light verbs (LV) are polysemous, semantically bleached and confined to a limited set of verbs (Paul, 2010; Das et al., 2010). Complex predicates are also considered as MWEs since the conventional meaning of light verbs in complex predicates is usually absent (Baldwin and Kim, 2010). Traditional PB-SMT systems derive phrase pairs directly from the training corpus purely based on statistical methods. Thus, PB-SMT phrase pairs may not follow the MWE constituents of a sentence; they are just n-grams. This we expect is one of the reasons why PB-SMT often produces wrong word translations for MWEs. Our approach is to restrict the phrase extraction module to extract phrase pairs that respect MWE boundaries. This approach ensures that the extracted phrase pairs are not just n-grams; they also contain MWE knowledge to some extent.

Ideally, NEs – particularly multiword NEs – on the source and the target sides of a parallel corpus should be aligned and translated as a whole. This is also true for MWEs and complex predicates in general (Pal et al., 2011). However, in state-of-the-art PB-SMT systems, the constituents of such MWEs are often split and aligned as part of consecutive phrases since PB-SMT (or any other approaches to SMT) does not generally treat MWEs as special tokens. This motivated us to consider NEs for special treatment in this work by converting them into single tokens to make sure that PB-SMT also treats them as a whole. There has been some work to handle MWEs in PB-SMT (Pal et al., 2011, 2013b) using a single tokenization technique and they also proposed various alignment methods for MWEs. We follow these methods for a resource-rich language pair (English–German) and some resource-poor language pairs (Indian languages, English–Hindi, etc.). Additionally, and orthogonally, we also investigate how EBMT phrases can provide further improvement in SMT. However, in this work, instead of only using one-to-one alignment of MWEs and NEs, we also apply a pseudo aligned parallel corpus[2] containing MWEs and NEs.

The first objective of the work described in this chapter is to investigate how single tokenization, prior alignment (pseudo alignment) of NEs and MWEs, and the use of these parallel terminologies as supplementary training data affect the overall MT quality.

[2]These works have been published as shared task papers in WMT 2014 (Tan and Pal, 2014), 2015 (Pal et al., 2015a) and ICON 2014 (Pal et al., 2014c).

The second objective is to investigate whether a hybrid word alignment model combining both unsupervised and semi-supervised techniques can enhance the quality of translation. Our hybrid word alignment combines several word alignment models including explicit alignment of MWEs NEs, and EBMT phrases.

EBMT phrases tend to be more linguistically motivated than SMT phrases which essentially operate on n-grams. The knowledge extraction as well as the representation process, in both EBMT and SMT, use very different techniques in order to extract resources. In our research, we extract EBMT phrases following the work of Cicekli and Güvenir (2001). High frequency EBMT phrases obtained from the training corpus are added to our training corpus as additional training material. Prior (EBMT) phrase alignment helps the statistical aligner operating on the extended training material indirectly in the sense that more evidence is provided to the statistical aligner about highly frequent EBMT phrases. It also narrows down the focus of the alignment at the beginning of IBM Model 1's initially uniform probability estimation. The EBMT phrases facilitate the IBM models to make the alignments more reliable and definite and they also help the IBM models converge faster during the training phrase of SMT. Thus, prior (EBMT) phrase alignment indirectly improves the performance of the statistical word aligner, which in turn results in the extraction of well aligned source–target phrases during the phrase extraction process. Reduction in noisy alignments also reduces the size of the phrase table which is prepared during the SMT training pipeline. Moreover, a smaller translation model also results in faster translation during decoding time by reducing the search space. This motivated us to improve the quality of state-of-the-art word alignment methods by applying a hybrid methodology. We present an improvement of word alignment quality by combining three statistical word alignment tables: (i) GIZA++ alignment (Och and Ney, 2003a), (ii) SymGiza++ alignment (Junczys-Dowmunt and Szał, 2012) and (iii) Berkeley alignment (Liang et al., 2006). Our third objective is to assess the effectiveness of the hybrid word alignment model and to see whether it can enhance the overall translation quality.

Each data-driven approach (in this case, EBMT and SMT) has its own method of acquiring and using translation knowledge from the parallel bilingual translation examples, along with its own advantages and disadvantages. The SMT phrases operate on n–grams, rather than syntactic phrases as in EBMT. Many researchers have investigated combining these different MT approaches (hybrid MT) to achieve better performance (Smith and

Clark, 2009; Dandapat et al., 2010, 2011). Our hybrid MT system described below is one such approach. Additionally, we incorporate aligned MWEs and NEs as additional training material. The end process of the hybrid pipeline is the combination of different SMT based engines developed in different component settings. We investigate the performance of our approach with

1. Resource poor languages (English–Hindi, Bengali–Hindi, Marathi–Hindi, Tamil–Hindi, and Telugu–Hindi) in two different domains: tourism and health
2. Resource-rich languages (English–German)

Reordering is the one of the most difficult problems in SMT; it presents itself differently for different language-pairs. For some language pairs (English–French, Chinese–English, etc.) only local movements are sufficient for translation, while some language-pairs have significant syntactic divergences. Particularly, SMT between SVO–SOV (e.g., English–Hindi, English–Bengali etc., here S, V and O stand for subject, verb, and object, respectively) or SVO–VSO (e.g., English–Arabic) language pairs suffer from long-distance reordering phenomena. Most of the Indian languages are relatively free phrase-order languages; they are generally verb-final, i.e., verb phrases are positioned at the end of the sentence and local movement of words within phrases also takes place (i.e., SOV). In section 4.3.3, we address this issue for English–Bengali language pair using a word-alignment based chunk pre-ordering approach.

In this chapter, we also investigate Forest to String Based SMT (FSBSMT) with hybrid word alignment settings. FSBSMT (Galley et al., 2004; Mi et al., 2008; Wu et al., 2011; Neubig, 2013) is a forest-based tree sequence to string translation model for syntax based SMT. The model automatically learns tree sequence to string translation rules from a given word alignment estimated on a source-side-parsed bilingual parallel corpus. This chapter also presents a hybrid method which combines different word alignment methods and integrates them into an FSBSMT system. The hybrid word alignment provides the most informative alignment links to the FSBSMT system. We show that hybrid word alignment integrated into various experimental settings of FSBSMT provides considerable improvement over state-of-the-art Hierarchical Phrase based SMT (HPBSMT). The research also demonstrates that additional integration of NE alignments and EBMT phrases (all extracted from the bilingual parallel training data) into the system brings

further improvements over the hybrid FSBSMT system. We apply our hybrid model to a distant language pair, English–Bengali.

4.2 Related Research

Like any other approach to data driven MT, phrase alignment in syntax based SMT or FSBSMT relies on word alignment quality and also on data preprocessesing (cf. Chapter 2). Hybrid approaches to word alignment have been able to successfully improve MT translation quality (Tu et al., 2012; Pal et al., 2013a). Previous research demonstrated that compact representations such as alignment combination (Och, 2003; Koehn et al., 2003; Ayan et al., 2005; DeNero and Macherey, 2011), can produce improved results. Inspired by Och (2003) and Koehn et al. (2003), a novel approach to combine multiple alignments for improving MT was proposed by Tu et al. (2012). Instead of combining exactly two bidirectional alignments as in (Och, 2003; Koehn et al., 2003), they used an arbitrary number of alignments. Apart from that they also considered the occurrences of potential links of individual alignments. To combine an arbitrary number of alignments, they constructed weighted alignment matrices over 1-best alignments (Liu et al., 2009; Tu et al., 2011) from multiple alignments generated by different models (including a refined model as well as minimum Bayes risk (MBR) based models). As the alignment probabilities between different alignment models are generally incomparable, they proposed a novel calculation of link probabilities in word alignment models. An alignment refinement model was applied to refine multiple alignments into a new alignment that favors the consensus of various models. The MBR decision is used to find the candidate hypothesis that has the least expected loss under a probability model when the true reference is unknown (Bickel and Doksum, 1977).

To alleviate the problem of reordering, researchers carried out work in two directions: one which tries to directly improve the reordering model inside the SMT system, and the other by prior reordering of the source text so that it resembles the target word order. This section also presents an overview of research that deals with prior reordering of the source text to emulate the target word order.

Prior reordering of the source text affects MT performance in two ways as stated in Holmqvist et al. (2012). Firstly, it lessens the burden of the reordering model since most

of the long-distance reorderings are taken care of during the reordering of the source text prior to training; only minor reorderings are performed during decoding and the translation hypothesis is constructed almost monotonically. Secondly, since statistical word alignment techniques are known to perform better for language pairs with similar word order, prior source reordering essentially should lead to more accurate word alignments and hence better translation model and improved translation quality.

Most of the research on pre-ordering relies either on automatically acquired (Xu et al., 2009; Niehues and Kolss, 2009; Genzel, 2010; Gupta et al., 2007; Habash, 2007) or handcrafted reordering rules (Collins et al., 2005; Popović and Ney, 2006). Reordering rules are usually automatically learned from parsed training data and/or word alignments.

Holmqvist et al. (2012) presented a method where source text is reordered to replicate the target word order based on word alignment. Then word alignment is performed again between reordered source and target training data; the new word alignments are transferred back to the original training data to connect words in their original order which results in the same parallel training data with potentially improved word alignments. Holmqvist et al. (2012) reported improved translation quality for English–German and English–Swedish. They also studied the effect of this preprocessing on the word alignment quality and found that this approach resulted in improved recall but degraded precision.

Andreas et al. (2011) reported improvements in an Arabic–English translation task by using two parse "fuzzification" techniques that allow the translation system to select among a range of possible subject–verb reorderings.

A syntax-driven approach to reordering using association rule mining was proposed by Avinesh (2010) where reordering rules are automatically learned from parsed source side data and word alignment; however it resulted in a drop in BLEU score compared to baseline Moses.

Dan et al. (2012) proposed linguistically motivated head-finalization reordering rules based on HPSG parses in a Chinese-to-Japanese translation task and reported significant improvements in translation quality. Gupta et al. (2007) proposed a POS-based prior reordering model which learns to reorder adjectives, nouns and verbs by observing the distances between the source and target phrases using target-to-source alignments. Their model was employed as an additional feature function at the rescoring stage of PB-SMT

and it resulted in improved BLEU scores in Japanese–English and German–English translation tasks.

Xu et al. (2009) presented a preordering approach where handcrafted precedence rules are applied recursively on dependency trees. They applied this approach on English to five SOV languages and achieved statistically significant improvements over the respective PB-SMT baselines for all the language pairs.

Niehues and Kolss (2009) proposed automatically extracting POS-based discontinuous reordering rules from word-aligned parallel data to model long-range reorderings. This method improves over applying POS-based continuous reordering rules and baseline PB-SMT.

Badr et al. (2009) presented linguistically motivated reordering rules that reorder English text to look like Arabic. To automatically detect and relocate clause-initial verbs in the Arabic side of a word-aligned parallel corpus, Bisazza and Federico (2010) proposed a chunk-based reordering technique that impacts the VSO type sentences in Arabic–English machine translation. Carpuat et al. (2010) proposed a novel approach to improve the SMT quality using a noisy syntactic parser that reorders verb-subject construction to subject-verb construction in Arabic–English SMT.

In this chapter, we propose word alignment-based pre-ordering of source chunks which is inspired by and an extension of Holmqvist et al. (2012). However there are two important distinctions between the work presented here and in Holmqvist et al. (2012). Firstly, the main objective of Holmqvist et al. (2012) was to improve word alignment, not reordering. They do not use the reordered training set to train the final system. Contrary to Holmqvist et al. (2012), in the present work we address both the issues of word alignment and reordering, and reorder the source side of all the datasets accordingly. Secondly, Holmqvist et al. (2012) reordered source words based on word alignment, whereas we suggest reordering source chunks. We also show that chunk-level reordering is much more effective than word-level reordering.

The motivation behind this work stems from the fact that word alignment-based pre-ordering of source words requires neither any reordering rules, nor any language dependent preprocessing. But word alignment-based reordering of source words is dependent on the quality of the word alignment. The objective of the present work is to reorder the

source chunks such that the source and target chunk alignments become monotone. We argue that with imperfect word alignments it might not be possible to produce perfectly monotone word alignments. However, by using these word alignments we can obtain monotone chunk associations which reduces the problem of long-range reordering to only short-range, intra-chunk reordering while preserving some source language syntax. The only language-dependent processing involved is chunking in the source language. The assumption is that human translators perform translation at chunk level rather than at the word level, and given the choices of translating from word- and chunk-reordered source text, human translators would much prefer translating from the latter.

Hybrid MT systems have been explored by many researchers using a combination of different modules, approaches, resources and paradigms. Chapter 2 provides an account of previous research on such hybrid MT systems.

4.3 Preprocessing

Effective preprocessing of data in the form of explicit alignment of bilingual terminology (viz. NEs and MWEs) (cf. Section 4.3.1 and 4.3.2) has been shown to improve the output quality of the baseline PB-SMT system (Pal et al., 2013a; Tan and Pal, 2014). Two kinds of terminologies, viz. NEs and MWEs, are considered in the present work. Intuitively, MWEs should be both aligned in the parallel corpus and translated as a whole. However, as we discussed earlier, state-of-the-art PB-SMT (or any other approaches to SMT) does not generally treat MWEs as special tokens. By converting the MWEs into single tokens, we make sure that PB-SMT also treats them as a whole. Translation correspondences between MWEs are mainly many-to-many. In our approach, once the MWEs are identified, they are converted into single tokens by replacing the spaces with underscores ("_") so that their alignments can be mapped to single tokens. Before decoding, MWEs in the source side of the testset are also single tokenized by looking up the extracted MWE list. For our experiments, we considered Point-wise Mutual Information (PMI), Log-likelihood Ratio (LLR) and Phi-coefficient for identification of MWEs. Finally, a system combination model was developed which provides a normalized score for each of the extracted MWEs. Candidates having scores above a predefined threshold value are considered as MWEs.

4.3.1 Named Entity Alignment

We applied two different methodologies to align bilingual NEs: **Method 1** for English–Bengali and **Method 2** for English–Hindi, Bengali–Hindi, Marathi–Hindi, Tamil–Hindi, Telugu–Hindi, and English–German.

Method 1: We identified NEs on the source (i.e., English) side of the parallel corpus using Stanford NER[3] (Finkel et al., 2005). NEs in the target side (i.e., Bengali) are identified using the NER system of Ekbal and Bandyopadhyay (2010). Next, we try to align the extracted source and target NEs. The alignment is trivial when both sides contain only one NE. We add such NE pairs to populate a parallel NE corpus that contains examples having only one token in both sides. Since Bengali has a different orthography than English, NE alignments are performed using transliteration and edit distance (Pal et al., 2010). However, for language pairs having the same orthography, NE alignments can often be established by making use of edit distance solely. If both the source and target sides contain n number of NEs, and the alignments of $n-1$ NEs can be established through the transliteration method or by means of already existing alignments, then the n^{th} alignment is established between the remaining (i.e., non-aligned) source and target NE. The bilingual NE pairs thus extracted serve as additional training material and they improve the word alignment.

Method 2: We initially identify NEs on both the source and target side of the POS-tagged parallel training corpus[4]. We create an NE parallel corpus (a pseudo-parallel corpus) by extracting the source and target NEs from the NE-tagged (NNP or N_NNP) parallel translations in which both sides contain at least one NE. For example, we extract the NE translation pairs given in (2) from the sentence pair shown in (1), where the NEs are shown as italicized.

(1a) In/IN this/DT *Yamuna*/NNP *Bio*/NNP *Diversity*/NNP *Park*/NNP an/DT effort/NN has/VBZ been/VBN made/VBN to/TO grow/VB and/CC preserve/VB the/DT herbs/NNS produced/VBN in/IN the/DT *Yamuna*/NNP region/NN ./.

[3] `http://nlp.stanford.edu/software/CRF-NER.shtml`

[4] POS tagged parallel corpus was released in the ICON 2014 shared task on translation between Indian languages. In case of English–Hindi WMT 2014 data, we developed our own POS tagger for Hindi (Tan and Pal, 2014).

(1b) इनमें/DM_DMR यमुना/*N_NNP* बायो/*N_NNP* डायवर्सिटी/*N_NNP* पार्क/*N_NNP* में/PSP यमुना/*N_NNP* क्षेत्र/N_NN में/PSP उपजने/V_VM वाली/PSP वनस्पतियों/N_NN को/PSP एक/QT_QTC जगह/N_NN उगाने/V_VM और/CC_CCD संरक्षित/N_NN करने/V_VM की/PSP कोशिश/N_NN की/V_VM गई/V_VAUX है/V_VAUX ।/RD_PUNC

(2a) Yamuna Bio Diversity Park Yamuna

(2b) यमुना बायो डायवर्सिटी पार्क यमुना

The above example (2a–2b) is not an exact one-to-one NE alignment; instead both source and target NEs preserve their respective order as they occur in the parallel corpus. The resultant corpus is a pseudo-parallel corpus containing only NEs. Compared with Method 1, Method 2 is a simple method that is more easily scalable to many language pairs. Although the resulting bilingual NE table does not provide a perfect NE dictionary, it filters out useful NEs from the training sentences and improves word alignments at the start of the MT pipeline due to the additional training data.

4.3.2 Multi-word Expression Alignment

We extracted highly collocated MWEs on both the source and target side. The extraction methods are based on statistical association measurement techniques. In the case of complex predicates identification (extracted only from the Bengali side) we followed the approach described in Das et al. (2010) which uses a rule based approach to identify the lexical patterns of complex predicates based on the information provided by shallow morphology and a seed list of verbs. Finally a fine-grained error analysis through a confusion matrix was performed to highlight the limit of lexical patterns and in addition the impact of different constraints for identifying the complex predicates. For MWE identification other than complex predicates on both source and target side of the bilingual training data, we considered the following association measures.

Point-wise Mutual Information (PMI): This is an information-theoretic measure for discovering interesting collocations (Church and Hanks, 1989). Point-wise mutual information is defined in Equation 4.2,

$$PMI(x, y) = \log\left(\frac{p(x, y)}{p(x)p(y)}\right) \tag{4.2}$$

where, $p(x, y)$ is the probability of the words x and y occurring together, $p(x)$ is the probability of x occurring in the corpus and $p(y)$ is the probability of y occurring in the corpus.

Log-Likelihood Ratio (LLR): This is the ratio between the probability of observing one component (i) of a collocation given the other (j) is present (i.e., $f(i, j)$) and the probability of observing the same component of a collocation in the absence of the other $(f'(i, j))$ (Dunning, 1993).

$$LLR(x, y) = -2 \sum_{i,j} f(i, j) \log \Big(\frac{f(i, j)}{f'(i, j)} \Big) \tag{4.3}$$

Here the sequence of the words in the candidate collocation is irrelevant. We adopted the probability using Bayes' theorem by averaging the probability of collocates w_1 given w_2 and probability of w_2 given w_1.

Phi-Coefficient: In statistics, the Phi coefficient (Φ) is a measure of association for two binary variables. Φ is also related to the chi-square statistic as in Equation 4.4:

$$\Phi = \sqrt[2]{\frac{\chi^2}{n}} \tag{4.4}$$

where n is the total number of observations and χ^2 is the chi-square distribution. Two binary variables are considered positively associated if most of the data falls along the diagonal cells. Here, the binary distinction denotes the positional information of the words. If we have a 2×2 table for two random variables x and y which denotes the presence of words w_1 and w_2 respectively, we have the following matrix:

	$y = 1$	$y = 0$	Total
$x = 1$	n_{11}	n_{10}	n_{x_1}
$x = 0$	n_{01}	n_{00}	n_{x_0}
Total	n_{y_1}	n_{y_0}	N

TABLE 4.1: Phi-matrix

where, n_{11} is the actual bigram frequency of (w_1, w_2), n_{10} is the frequency of bigrams containing w_1 but not w_2, n_{01} is the frequency of bigrams containing w_2 but not w_1, n_{00} is the frequency of bigrams neither containing w_1 nor w_2. Note that n_{x_1} and n_{x_0} are the

summation of their respective rows and n_{y_1} and n_{y_0} are the summation of their respective columns. Alternative words are in place of absent w_1 or w_2. The Phi coefficient describes the association of x and y and is shown in Equation 4.5

$$\Phi = \frac{n_{00}n_{11} - n_{01}n_{10}}{\sqrt[2]{n_{x_0}n_{x_1}n_{y_0}n_{y_1}}} \tag{4.5}$$

Finally, a MWE system combination model was developed which gives a weighted combination score (Chakraborty et al., 2011) to each of the associations. Weights are set based on the training and tuning of both the English and Bengali side separately. We train the English side using an open source MWE corpus[5] containing both the training and development set, and similarly the Bengali side was trained using MWE resources described in (Chakraborty et al., 2014). All statistical measures are considered in the weighted scheme to assign weights to the candidate phrases. After weight tuning on the development data, the optimal weights are assigned with each of the individual scores. The individual score of each measure is normalized before assigning weights, therefore, all of them fall in the range of 0 to 1. For each measurement, the scores are sorted in descending order; the intuition is that the higher the value of the statistical measure for a candidate phrase, the more it behaves like an MWE. A predefined cut-off score (70% i.e., > 0.7) was considered and the candidates having scores above the threshold value are treated as MWEs.

We extracted MWEs separately from the source and target sentences and prepared another MWE-aligned pseudo-parallel corpus containing only MWEs (like the preparation of pseudo-parallel corpus in Section 4.3.1). This MWE-aligned corpus is later used as additional parallel training data for our hybrid system.

4.3.3 Preordering

To address the problem of word reordering in SMT, language models play a crucial role in positioning the target words in an acceptable order. But language models also have their limitations; to keep the model size within acceptable limits, a language model typically considers up to 5-grams, which can not capture long distance dependencies and hence is not sufficient to make decisions about good translations. If we increase the value of n

[5] `https://www.ukp.tu-darmstadt.de/data/lexical-resources/wikimwe/`

then the reordering cost involved becomes much higher in terms of computational effort and requirements; besides longer n-grams in language models suffer from data sparsity.

In the PB-SMT framework, reordering is typically handled by two models: a distortion model and a lexicalized reordering model (Koehn et al., 2005; Galley and Manning, 2008). The distortion model was proposed by the IBM Models (Brown et al., 1993). IBM models 1 and 2 define the distortion parameters in terms of the word positions in the sentence pair instead of the actual words at those positions. The distortion probability also depends on the source and target sentence lengths. Models 4 and 5 limit this by replacing absolute word positions with relative word positions. However, all these models are limited to only word movements; they do not consider phrasal movement. Koehn et al. (2003) proposed a relative distortion model in PB-SMT. The model works by considering the difference between the current phrase position and the previous phrase position in the source sentence. The basic PB-SMT model considers word movements up to 6 tokens which could be increased to consider long distance reordering; however, higher distortion limits often result in degraded performance (Koehn et al., 2007).

The lexicalized reordering model conditions reordering on the PB-SMT phrases. It consists of three types of reordering – monotone (M), swap (S), and discontinuous (D) – by considering the orientation of the previous and the next phrases. The orientation is called monotone if the previous source phrase is aligned with the previous target phrase. The swap orientation occurs when the next source phrase is aligned with the previous phrase in the target; and the orientation is termed as discontinuous if neither of the two above mentioned cases are true. The reordering model is built by calculating the probabilities of the phrase pairs being associated with the given orientation. Notwithstanding the reordering models used in the state-of-the-art PB-SMT, the differences in word ordering between distant languages often result in poor translation quality.

In this section we discuss (inter alia) a simple yet effective language-independent approach to pre-reordering based on word alignment which follows Holmqvist et al. (2012). This method has the advantage that it does not require any reordering rules, neither hand-crafted nor automatically acquired. It also avoids any language-dependent preprocessing of the target language; it only requires chunking of the source language. There are two important distinctions between the work presented here and in Holmqvist et al. (2012). Firstly, the main objective of Holmqvist et al. (2012) was to improve word alignment, not

reordering. They do not use the reordered training set to train the final system. Secondly, Holmqvist et al. (2012) reordered source words based on word alignment, whereas we suggest reordering source chunks. We also showed that chunk-level reordering is much more effective than word level reordering.

4.3.3.1 Tree Based Reordering

To compare the effectiveness of our word alignment based reordering approach, we build a linguistically motivated syntactic reordering approach which follows target language ordering rules. For tree-based reordering (Xia and McCord, 2004; Collins et al., 2005) we only consider repositioning the verbs at the end of the sentence or clause. We categorize each source sentence into three basic types[6]: simple, complex and compound, and reposition the verbs accordingly. For identifying the basic sentence type we first parse the source sentences. The parse trees are categorized into the above mentioned three types by analyzing the structure of the tree and presence of keywords such as 'that', 'which', and 'who' as well as by looking at the presence of tags like 'CC', 'WHNP', 'SBAR', and 'S'.

4.3.3.2 Word Alignment-based Reordering

In this reordering approach we first run the GIZA++ word alignment tool on the original parallel corpus bidirectionally which produces $1 - to - n$ alignments for both directions. Then a symmetrization matrix is built on these two unidirectional word alignments and the 'grow-diagonal-final-and' (GDFA) heuristic is applied which produces many-to-many alignments. The GDFA heuristic is often believed to be the most favourable word alignment heuristic for PB-SMT, and is used in the Moses vanilla settings. This word alignment serves as the basis for our source reordering approach.

Once the word alignment has been obtained, chunks are identified in the source (i.e., English) side of the training set which are then reordered following the word alignment. For chunk identification in English sentences, we used the CRF Chunker[7] (Sha and Pereira, 2003).

[6] We consider English as source language for this experiment.

[7] `http://crfchunker.sourceforge.net/`

For reordering a source sentence, the algorithm starts with the chunked source sentence and the word alignment for that sentence pair. Let us consider the following chunked source sentence, the target sentence and the word alignment:

$$S = (s_1, s_2, s_3, \ldots, s_p) = (C_1, C_2, C_3, \ldots, C_m)$$

where

$$C_i = (s_j, \ldots, s_{j+n})$$

and

$$T = (t_1, t_2, t_3, \ldots, t_q)$$

where S is a source sentence, T is the corresponding target sentence, and s, t and C represent source words, target words and source chunks, respectively. The alignment between words in S and T is given by:

$$A = \{a_1, a_2, \ldots, a_r\}, \text{ where } a_k = [s_j, t_l]$$

For the sake of simplicity the algorithm assumes 1-based indexing while 0-based indexing is used for the actual alignments.

The algorithm uses a list of indices, $list_{pos}$, for each source chunk. $list^i_{pos}$ stores indices of target words which are linked to the component words of the i^{th} source chunk (C_i) via word alignment, i.e.,

$$list^i_{pos} = \{j : t_j \in T \wedge \exists k : s_k \in C_i \wedge [s_k - t_j] \in A\}$$

In an ideal scenario, all tokens in a source sentence, or at least some tokens in every source chunk should be aligned to some tokens in the corresponding target sentence; but that is not always the case. If no correspondence can be found with the target via word alignment for any of the tokens belonging to C_i, the source chunk position of C_i is simply added to $list^i_{pos}$. Finally, the entries in each $list_{pos}$ are sorted in ascending order, and the chunks are arranged according to the first entry in the corresponding $list_{pos}$.

The pseudo-code of the algorithm used to reorder source chunks according to word alignment information is given in Algorithm 2.

Algorithm 2: Word alignment-based source chunk reordering

```
for i = 1 to m chunks in the source do
    new_pos_i = NULL;
    for j = 1 to n source words in chunk_i do
        for k = 1 to r alignments in A do
            if a_k =[s_j, t_l] then
                add l to list new_pos_i;
            end
        end
    end
    if new_pos_i is NULL then
        add i to new_pos_i;
    end
end
for i = 1 to m new_pos lists do
    Sort the items in new_pos_i in ascending order;
end
reordered_sen = NULL;
while not(all new_pos lists are empty) do
    i = index of the first new_pos list containing the smallest first entry;
    j = first entry in new_pos_i;
    Append chunk_j to reordered_sen;
    new_pos_i = NULL;
end
```

4.3.4 Example Based Phrase Alignment

Example based phrase pairs are extracted based on the work described in Chapter 3 and in (Cicekli and Güvenir, 2001), who proposed a compiled approach of EBMT that automatically extracts translation templates from sentence-aligned bilingual text by observing the similarities and differences between two example pairs.

Since this particular approach has a cubic run-time complexity, it takes a significant amount of time to extract phrase pairs even from a small corpus. Therefore we used

heuristics to reduce the time complexity. We divided the entire corpus into n clusters based on sentence length such that similar length sentences belong to the same cluster. We extract *atomic* example-based translations from each of these clusters.

4.4 Hybrid Word Alignment

Our hybrid word alignment model is trained on the parallel bilingual training corpus with aligned MWEs, NEs and EBMT phrases as additional training materials. The hybrid word alignment model is a combination of three statistical word alignment models as described below.

4.4.1 Word Alignment Using GIZA++

GIZA++ (Och and Ney, 2003a) is a statistical word alignment tool which implements maximum likelihood estimators for IBM models 1-6 and an HMM alignment model. The model parameters of GIZA++ are generally estimated from large amounts of parallel data. Symmetrization methods are able to provide some improvements in MT where the parallel corpora are trained bidirectionally to establish the word alignment. The two alignment tables are reconciled using different heuristics, e.g., union, intersection, grow-diagonal-final and grow-diagonal-final-and heuristics (Koehn, 2010). In spite of these heuristics, the word alignment quality provided by GIZA++ often remains low and calls for further improvement.

4.4.2 Berkeley Aligner

Like GIZA++, the Berkeley Aligner (Liang et al., 2006) is also a statistical aligner which is used to align words in a bilingual parallel corpus. The Berkeley Aligner allows the use of both unsupervised and supervised approaches to align words from parallel corpora. We initially train on the parallel corpus using the fully unsupervised method of producing Berkeley word alignments. The Berkeley aligner is an extension of the Cross Expectation Maximization word aligner. The aligner uses agreement between two simple sequence-based models during training and facilitates substantial error reductions over standard

models. Moreover, it is jointly trained with HMMs, and as a result the alignment error rate (Vilar et al., 2006) is substantially reduced.

4.4.3 SymGiza++

SymGiza++ (Junczys-Dowmunt and Szał, 2012) modified the counting phase of each model of Giza++ to allow updating of the symmetrisised models between the chosen iterations of the original training algorithms. It computes symmetric word alignment models with the capability of taking advantage of multi-processor systems. Experimental results show that the alignment quality improves by more than 17% compared to Giza++.

4.4.4 Hybridization

Our hybrid word alignment method combines three different statistical word alignments – Giza++ word alignment with grow-diag-final-and (GDFA) heuristic (Koehn, 2010), Berkeley word alignment and SymGiza++ word alignment and for each of these we use our bilingual training data together with the translation pairs extracted using the pre-alignment methods (described in Section 4.3) for NEs, MWEs and example-based phrases. We prepared an one-to-one pre-alignment set containing only one-to-one alignment pairs of NEs, MWEs and atomic translation pairs of EBMT phrases. We have followed the strategies to combine all word alignment tables as described in (Pal et al., 2013a).

The hybridization method uses the following heuristic. We consider either of the alignments generated by GIZA++ GDFA (a_1), Berkeley aligner (a_2), or SymGiza++ (a_3) as the standard alignment. One-to-one pre-alignment set (a_4) generated by the methods described in Section 4.3.1 and 4.3.2 to produce additional alignment points (contains only one-to-one alignment pairs). We combine the four alignments a_1–a_4 following the method described in Algorithm 3. Although alignment pairs from the pre-alignment methods are contained within the word alignment training data (for a_1, a_1, and a_3), the pre-alignments (a_4) are specifically used (cf. Algorithm 3) to remove some noisy alignment points (involving the source NE, MWE tokens and atomic EBMT phrases) produced by the statistical aligners.

Algorithm 3: Producing alignment combination

- **Step 1:** Choose a standard alignment (S_a) from a_1, a_2 or a_3. ▷ The empirically best preforming aligner among the individual aligners (a_1, a_2 or a_3) is considered as S_a.
- **Step 2:** Produce a combined alignment $S_c = S_a \cup (a_2 \cap a_3)$, if a_1 is considered as S_a.
- **Step 3:** Delete all the alignment points $a_{ij} \in S_c$ such that $\exists a_{ik} \in a_4$ where $j \neq k$.
- **Step 4:** Update S_c as $S_c = S_c \cup a_4$.

4.5 Forest-to-String Based SMT

Forest-to-String Based SMT (FSBSMT) (Galley et al., 2004; Mi et al., 2008; Wu et al., 2011; Neubig, 2013) is an extension of tree-based SMT. Current tree-based systems suffer from a major drawback: during translation they only use the 1-best parse tree, which might result in incorrect translation due to parsing errors. In forest-based systems, the decoder produces translations of a packed forest of exponentially many k-best parses. A *forest* is a compact representation of all the parse trees for a given input sentence under a context-free grammar.

There are two separate steps preformed by existing standard tree-based systems (Yamada and Knight, 2001): (i) parsing of the source input sentence into a 1-best tree τ and (ii) decoding, where the decoder searches for the best derivation δ^* that translates source tree τ into a target-language string among all possible derivations D:

$$\delta^* = argmax_{\delta \in D} P(\delta|\tau) \tag{4.6}$$

Equation 4.6 can be unpacked as:

$$\begin{aligned} \delta^* = argmax_{\delta \in D} P(\delta|\tau)^{\lambda_0} \times e^{\lambda_1|\delta|} \\ \times P_{lm}(s)^{\lambda_2} \times e^{\lambda_3|s|} \end{aligned} \tag{4.7}$$

Equation 4.7 can be represented as a log-linear model:

$$\delta^* = argmax_{\delta \in D} \lambda_0 \log P(\delta|\tau) + \lambda_1|\delta| + \lambda_2 \log P_{lm}(s) + \lambda_3|s| \tag{4.8}$$

where, $e^{\lambda_1|\delta|}$ is the penalty term on the number of rules in a derivation, $P_{lm}(s)$ is the language model score and $e^{\lambda_3|s|}$ is the length penalty term on the target translation s.

The decoding step of FSBSMT translates the parse forest using the set of translation rules. A technique of pattern-matching from tree-based decoding is applied to convert a parse forest into a translation forest. The decoder chooses the best derivation from the translation forest and finally produces the translation output in the form of a target string. Therefore, in FSBSMT the derivation probability $P(\delta|\tau)$ is now replaced by $P(\delta|\hbar)$ where $\hbar$ is the parse forest, and this is the product of probabilities of translation rules $r \in \delta$.

$$P(\delta|\hbar) = \prod_{r \in \delta} P(r) \tag{4.9}$$

Each $P(r)$ is defined as the product of five different probabilities as in Equation 4.10 . Let t and s be the source-side tree and target-side string of rule r, respectively, $P(t|s)$ and $P(s|t)$ are the two translation probabilities, and $P_{lex}(t|s)$ and $P_{lex}(s|t)$ are the two lexical probabilities. $P(t|\hbar)$ denotes the source side parsing probability of the current translation rule r in the parse forest.

$$P(r) = P(t|s)^{\lambda_4} \times P(s|t)^{\lambda_5} \times P_{lex}(t|s)^{\lambda_6} \times P_{lex}(s|t)^{\lambda_7} \times P(t|\hbar)^{\lambda_8} \tag{4.10}$$

We incorporated FSBSMT as an alternative MT engine for the multi-engine framework (Pal et al., 2016a) described below.

4.6 Multi-Engine Hybrid System

The MT system combination framework implies selecting the best hypothesis translation from multiple hypotheses produced by different systems. In order to apply this framework

to the translations produced by our systems we implemented the Minimum Bayes Risk (MBR) coupled with the Confusion Network (MBRCN) framework as described in (Du et al., 2009). The MBR decoder (Kumar and Byrne, 2004) selects for each sentence the best system output from the n outputs by minimizing the BLEU (Papineni et al., 2002) loss. This output is known as the backbone. A confusion network (Matusov et al., 2006) is built from the backbone while the remaining hypotheses are aligned against the backbone using an edit-distance based alignment method (TER alignment). The features used to score each arc in the confusion network (CN) are word posterior probability, target language model (3-gram, 4-gram), and length penalties. Minimum Error Rate Training (MERT) (Och, 2003) is applied to tune the CN weights (Pal et al., 2014c).

4.7 Experiments with English–Bengali Data

In this section we describe the system performance on a low-resource language pair - English–Bengali. We experimented with various experimental settings which are detailed in the following subsections.

4.7.1 Data

We used an English–Bengali parallel corpus[8] containing 25,000 sentences from the travel and tourism domain (Pal et al. (2010) used the same data for their experiment). Corpus cleaning was carried out first by calculating the global mean ratio of the number of characters in a source sentence to that in the corresponding target sentence and then filtering out sentence pairs that exceed or fall below 20% of the global ratio (Tan and Pal, 2014). Tokenization and punctuation normalization were performed using Moses scripts. Finally, we filtered the parallel training data using a maximum allowable sentence length of 100 tokens and sentence length ratio of 1:2 (either direction).

After cleaning, the English–Bengali parallel corpus contained 23,492 parallel sentences consisting of 569,600 source tokens and 489,609 target tokens. We randomly selected 500

[8]This corpus is produced in the *EILMT* project funded by the Department of Electronics and Information Technology (DEITY), Ministry of Communications and Information Technology (MCIT), Government of India.

sentences each for the development set and the test set from this filtered parallel corpus and treated the rest as the training corpus.

4.7.2 Experiment with Forest-to-String Based SMT

The effectiveness of the FSBSMT approach is demonstrated by comparing it against the Hierarchical phrase-based SMT (HPBSMT) (Chiang, 2005) model which serves as our baseline. For building the baseline HPBSMT system, we use the maximum phrase length of 7 and a 5-gram language model. For performing word alignment for the baseline systems, we used the Berkeley Aligner (BA) as BA generally provides better alignment than the GIZA++ implementation of IBM word alignment model 4 with grow-diagonal-final-and heuristics. The phrase extraction process was carried out using the hierarchical model (Chiang, 2005). For FSBSMT (Neubig, 2013) (described in Section 4.5), rule extraction from forests is performed using the method described in (Mi and Huang, 2008). To build our FSBSMT systems, we used Egret[9] to parse the English sentences as it provides high accuracy parsing as well as the output of k-best parses in the packed forest format.

The 5-gram target language model was trained using KENLM (Heafield, 2011) on the target-side of the training data. Parameter tuning for both HPBSMT and FSBSMT was carried out using both k-best MIRA (Cherry and Foster, 2012) and Minimum Error Rate Training (MERT) (Och, 2003) on the held-out development set. After the parameters were tuned, decoding was carried out on the held out test set. In the set of experiments presented here, we first integrated the hybrid word alignment model (c.f., Section 4.4) within both the hierarchical phrase-extraction (Chiang, 2005) as well as the state-of-the-art forest to string based phrase extraction model.

4.7.2.1 Results

To test the effect of the hybrid word alignment model on the forest based system, we compared the systems with various experimental settings. We evaluated the systems

[9] `http://code.google.com/p/egret-parser/`

SYSTEM		Experiment	BLEU	NIST	TER	METEOR
HPB	BA	1	12.53	4.34	72.93	40.97
	SYM	2	11.20	4.35	71.10	39.67
	GIZA	3	11.62	4.25	73.90	40.45
	BA_FB	4	12.96	4.38	72.41	41.27
FB	GIZA	5	17.79	4.62	66.78	41.61
	GIZA_NEA	6	18.30	4.70	**66.27**	42.03
	BA	7	21.28	4.77	69.37	41.88
	BA_NEA	8	21.40	4.81	69.20	42.05
	HWA_NEA	9	22.03	4.91	67.67	**42.94**
	HWA_NEA_EBMT	10	**22.37**	**4.92**	67.53	42.52

TABLE 4.2: Systematic evaluation results for English–Bengali. HPB:=HPBSMT, FB:=FSBSMT; All FB outputs provide statistically significant improvements over HPB.

using three well known automatic MT evaluation metrics: BLEU (Papineni et al., 2002), NIST[10], METEOR (Lavie and Agarwal, 2007) and TER (Snover et al., 2006a).

The evaluation results are reported in Table 4.2. We used HPBSMT (Experiment 1) implemented with BA as our baseline model. As reported in Table 4.2, BA performed better than the other statistical word aligners such as GIZA++ (Experiment 3) and Sym-Giza++ (Experiment 2). The BLEU score of the baseline is 12.53. Experiment 5 is a simple GIZA++ implementation of FSBSMT while Experiment 6 is an extension of Experiment 5 where we make use of the NE aligned parallel data as additional parallel training examples. Similarly, the Experiment 7 system is a BA implementation of FSBSMT and Experiment 8 additionally uses NE alignments as extra training material.

The use of a combination of multiple alignments, i.e., hybrid word alignment (HWA) implemented in the FSBSMT system, improves the BLEU score further. The HWA combined with NEA and prior high frequency EBMT phrases (Experiment 10 in Table 4.2) provided the overall best performance in terms of both BLEU (22.37) and NIST (4.92), while HWA with NEA produced the overall best METEOR score (42.94) and the Experiment 6 system resulted in the best TER (66.27) score. The proposed FSBSMT system

[10]http://www.itl.nist.gov/iad/mig/tests/mt/2008/doc/ngram-study.pdf

provides 78.5% relative (9.84 absolute) BLEU points improvement over the baseline HPB-SMT. The relative improvement in terms of BLEU compared to the vanilla settings of HPBSMT is 92.5% (relative).

4.7.3 Experiment with Word Alignment based Pre-reordering

The parallel dataset (English–Bengali) used for the experiments is described in Section 4.7.1. For identification of chunks, the English training set and testset sentences are first POS-tagged using the Stanford POS tagger[11]. Chunks are identified from the POS-tagged sentences using a CRF chunker[12](Sha and Pereira, 2003). The source side of the datasets were parsed using the Stanford Parser[13] (de Marneffe and Manning, 2008) for tree based reordering.

The MT experiments were carried out using the standard log-linear phrase-based SMT toolkit MOSES (Koehn et al., 2007), GIZA++ (Och and Ney, 2003a) implementation of IBM word alignment model 4 with the 'grow-diagonal-final-and' heuristic for performing word alignment. Phrase extraction was performed following Koehn et al. (2003). The feature weights were tuned using MERT (Och, 2003) on a held-out development set in terms of BLEU. For language modelling purpose we used the SRILM toolkit (Stolcke, 2002) with Kneser-Ney smoothing (Kneser and Ney, 1995) on the target side of the bilingual training data.

We carried out the experiments with a 4-gram language model and maximum phrase length of 7 as this produced the best results for the baseline PB-SMT system. Table 4.3 presents the experimental results. We carried out experiments on tree-based and word alignment-based source reordering[14]. To compare the effect of word alignment-based reordering at chunk- and word-level, we carried out experiments on both. For the sake of completeness we also carried out experiments on word-based SMT (setting the phrase length to 1) to see whether chunk-level reordering could bring any improvement over baseline word-based SMT. We also replicated the experiment of (Holmqvist et al., 2012) on this dataset. Holmqvist et al. (2012) reported a 1-pass reordering experiment, while

[11] `http://nlp.stanford.edu/software/tagger.shtml`
[12] `http://crfchunker.sourceforge.net/`
[13] `http://nlp.stanford.edu/software/lex-parser.shtml`
[14] We frequently switch the terms 'reordering' and 'pre-reordering'. In this work, we pre-reordered the source-side of the parallel corpus.

we carry out both 1-pass and 2-pass experiments. In the 2-pass experiment, the process of reordering the source side is simply carried out twice, i.e., the reordered source side is subjected to reordering once again. We also carried out a chunk-reordering experiment in PB-SMT where the chunks are reordered based on the final alignments obtained by the 1-pass experiment of (Holmqvist et al., 2012).

It is to be noted that for applying any pre-reordering technique, the test set (and in case of tuning, the development set) needs to be reordered as well using the same technique that was applied to the training data. For the tree-based reordering approach we reordered the test set and the development set using the same set of rules. For the word alignment based reordering experiments, the test set is reordered using monolingual PB-SMT systems built on the original source training data and the corresponding reordered source training data. For the monolingual PB-SMT systems, we do not perform automatic word alignment since the word alignments between the source training set and the reordered training set are already known. We create two lexical translation tables where each source word has only one translation option, i.e., the same word itself in the target, with a translation probability of 1.0. It is to be noted that both these lexical translation tables are exactly the same. The phrase table and the reordering table are built on these alignments using Moses. Since the purpose of this monolingual PB-SMT system is to reorder the source sentences, we do not use a language model for this monolingual PB-SMT model. A monolingual PB-SMT system built thus essentially just reorders the source sentences. The 'TR' column in Table 4.3 indicates whether or not the test set is reordered (using monolingual MT) in the corresponding experiment.

Experiments	**Prior reordering**	**Level**	**TR**	**Exp**	**BLEU**	**NIST**	**METEOR**	**TER**
Word-based SMT	none (baseline)		no	1	8.87	3.61	0.3028	86.95
	alignment-based	word	yes	2	8.97	3.54	0.2985	88.86
		chunk	yes	3	9.94*	3.71	0.3107	86.64
Phrase-based SMT	none (baseline)		no	4	10.68	4.13	0.3035	73.37
	tree-based		yes	5	11.53*	4.22	0.3126	72.75
	alignment-based	word	yes	6	11.11	4.08	0.3073	75.34
		chunk	yes	7	**12.65***	4.29	0.3144	73.00
	alignment-based word reordering, 1-pass		no	8	11.25	4.09	0.3129	75.25
	alignment-based word reordering, 2-pass		no	9	11.47	4.12	0.3141	75.14
	alignment-based chunk reordering, 2-pass		yes	10	**13.17***	4.28	**0.3161**	**72.66**

Table 4.3: Evaluation results obtained on the reordering experiments.

We carried out evaluation of the MT quality using four automatic MT evaluation metrics: BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), NIST (Doddington, 2002) and TER (Snover et al., 2006a). For the PB-SMT experiments, tree-based reordering brings some improvements over the PB-SMT baseline. Word alignment-based reordering at word-level also provides some improvements over the PB-SMT baseline; however the improvements are smaller than those obtained in tree-based reordering. Word alignment-based reordering at chunk-level improves over both and provides the overall best BLEU score among all 1-pass PB-SMT experiments (Experiment 4–8). A similar trend is observed for the word-based SMT experiments for which both word- and chunk-level reordering prove to be beneficial over the baseline while chunk-level reordering appears to be more effective than word-level reordering.

Our approach to alignment-based chunk-reordering (Experiment 7) outperforms alignment-based word-reordering (Experiment 8) described in (Holmqvist et al., 2012). However, tree-based reordering produced the best scores as per TER among all 1-pass PB-SMT experiments.

The 2-pass approach to alignment-based word-reordering (Experiment 9) also improves over the 1-pass approach (Experiment 8) across all metrics; however the improvements are small. Our final experiment (Experiment 10) with chunk reordering based on the final alignments obtained by Experiment 8 produces the overall best scores in BLEU and TER. Statistical significance tests were carried out using the bootstrap resampling method (Koehn, 2004) and the '*' marked scores represent statistically significant improvements on BLEU over the respective baseline systems.

Figure 4.2 shows the effect of prior reordering of the source on word alignment. Figure 1.a shows the initial word alignment extracted by the baseline system for a sentence pair. Figure 1.b presents the correct (i.e., manual) alignment and Figure 1.c shows the final word alignment obtained by chunk-reordered (CR, Experiment 7) and word-reordered (WR, Experiment 6) PB-SMT systems for the sentence pair. Figure 1.b in addition shows whether the source chunks could be ordered properly (which is indeed the case here) and how they could minimize the number of cross links (28 down to 2 here). The correct alignments are shown as solid lines and the incorrect ones as dotted lines in Figure 1.a and Figure 1.c. English chunks are shown in brackets and Bengali chunks are shown as underlined. It is to be noted that chunking in the target side is not required in alignment-based

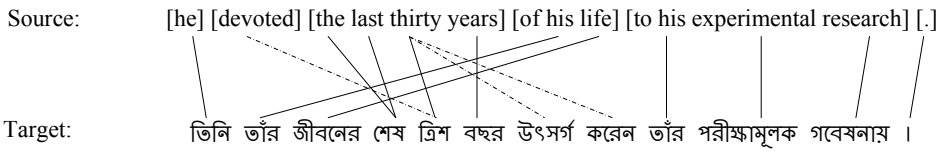


FIGURE 1.a – Initial word alignment

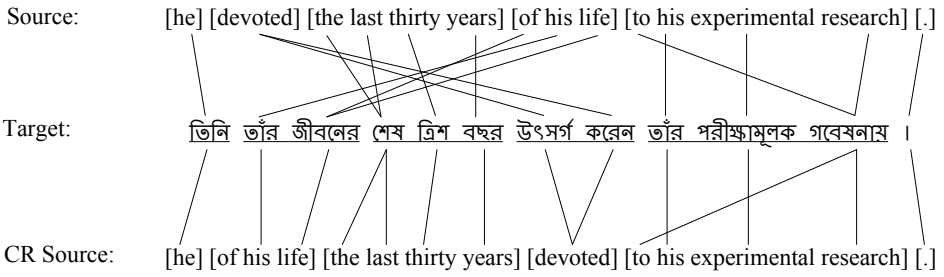


FIGURE 1.b – Correct word alignment

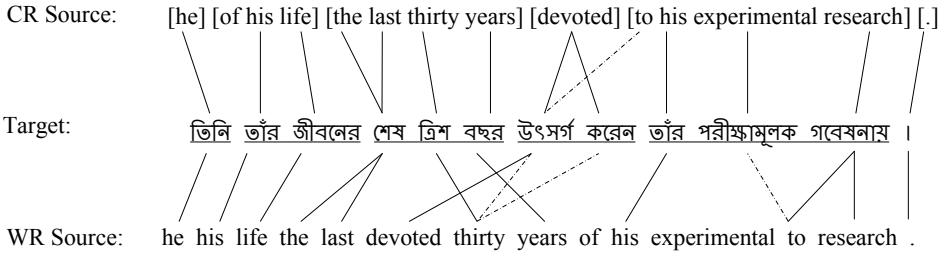


FIGURE 1.c – Final word alignment

FIGURE 4.2: Word alignments with unordered and reordered source.

reordering; the target side has been chunk-marked (i.e. underlined) in Figures 1.b and 1.c just for visualization of source-target chunk associations. In the initial word alignment (cf. Figure 1.a) 11 out of 14 word associations are correct (precision=0.79, recall=0.73). However, when the source sentence is (chunk-) reordered based on this initial word alignment, the association between the source and target chunks becomes monotone (cf. Figure 1.c). In the final word alignment between the word-reordered source and the target sentence, 12 out of 15 alignments are correct (precision=0.8, recall=0.8), an improvement over the baseline, while 13 out of 14 alignments are correct (precision=0.93, recall=0.87) between the chunk-reordered source and the target sentence. Thus, word-alignment based source reordering improves both precision and recall for word alignment. This example illustrates two important improvements: firstly, word-alignment based chunk reordering of the source results in fewer cross-chunk alignments, in this case zero (cf. Figure 1.c), and secondly and more importantly, it improves the accuracy of the word alignment. From this example it is also evident that word-alignment based chunk reordering is more

effective than word-alignment based word reordering in PB-SMT. This approach to reordering can be considered as a bootstrapping approach to word alignment since it is based on word alignment and the purpose of it is to improve the word alignment quality. Word alignments produced by statistical word aligners are never perfect even for sizable amount of data; if they were perfect it would have defeated the purpose of reordering. In this real-world scenario it makes more sense to reorder at chunk level than at word level since both rely on imperfect word alignments while chunk-level reordering preserves some source language syntax and is less affected by noisy word alignments.

Due to the unavailability of the gold-standard word alignment, improvement in terms of word alignment quality could not be measured empirically; however the example presented in Figure 4.2 clearly demonstrates the usefulness of word-alignment based source chunk reordering in improving word alignment quality. Although this approach calls for the test set to be reordered (as opposed to (Holmqvist et al., 2012)) and is sensitive to errors in chunking, it was still able to produce significant improvements over the baseline systems. We inspected the lexfile[15] and phrase table sizes for the PB-SMT experiments and found that lexfile and phrase table sizes were inversely proportional to the BLEU scores obtained from them, which suggests that prior reordering also reduces the data sparsity problem.

4.8 Experiment with English/Indian Language (IL)– Hindi Data

We tested our hybrid system (described in Section 4.8.2) on the English/Indian Language (IL)– Hindi translation shared task organized by ICON 2014[16]

We conducted an analysis of the training data to filter noisy sentences, and append extracted NEs to the sentence pairs as additional training data. We were provided with 24,000 sentence pairs for the training set, 500 sentence pairs for the development set and 500 sentence pairs for the test set in each of the five language pairs (i.e., English–Hindi, Bengali–Hindi, Marathi–Hindi, Tamil–Hindi, and Telugu–Hindi) for each of the two domains: health and tourism. The general domain was obtained by combining the health

[15]The lexfile (i.e., lexical translation table) is prepared from the statistical word alignment during the training phrase of PB-SMT. The lexfile acts as a probabilistic bilingual lexicon.

[16]`http://ltrc.iiit.ac.in/icon/2014/contests.php`

Criteria	TOURISM	HEALTH
Initial Total	24000	24000
After Filtering	23207	23515
NE alignments added	24741	25148

TABLE 4.4: Summary of pre-processing of training data (number of sentences): Bengali→Hindi

and tourism data. Table 4.4 shows the number of sentences after filtering out (second row) and the number after adding NEs (third row) to the baseline training data (first row) for BN–HI. Similar numbers were obtained across all language pairs.

4.8.1 TM Implementation

Translation Memories (TM) are an important part of CAT tools. Since many translations are highly repetitive, it is useful to find existing translations for the entire source input sentence or part of the source input sentence and to reuse them. TMs reduce the workload of translators. Below we explore a way of integrating a TM in an MT focused translation workflow. Our TM stores existing translations that are collected from the training data. Our TM also contains the EBMT phrases and parallel NEs extracted from the training data. The basic functions of the TM are:

- **Case I:** If the source sentence is found in the TM, it will immediately return the target output sentence.
- **Case II:** If a sequence of words in the input sentence is found in the TM, the source sequence is also replaced with the corresponding target word sequence in the input sentence. The corresponding target sequences are marked as an arbitrary XML tag (e.g., <zone translation="eine englische Übersetzung">an English translation</zone>.) in the input sentence. The input sentences containing this XML are presented to the decoder. Any phrases (source word sequences) from the phrase table that overlap with that corresponding target sequence spans in the XML mark-up are completely ignored by the Moses decoder[17] (Koehn et al., 2007). E.g., let us

[17]In this case, during decoding process, the -xml-input flag with *exclusive* is used for our hybrid translation (cf. `http://www.statmt.org/moses/?n=Advanced.Hybrid`).

consider an input English sentence “This is <zone translation=“eine englische Übersetzung”>an English translation</zone>” is presented to the Moses decoder with ‘-xml-input exclusive’ option, the decoder retains the “eine englische Übersetzung” phrase in the generated German translation. Therefore, the complete translation will be “Das ist eine englische Übersetzung” .

4.8.2 Hybrid System

Our Hybrid approach combining TM, EBMT, and SMT was investigated with multi-way translation such as NE substitution, EBMT phrase substitution (cf. Case II in Section 4.8.1), followed by the SMT decoder. As mentioned earlier, we implemented four different systems, namely Baseline SMT, Baseline SMT with NE alignment (NEA), NEA with EBMT phrase alignment (NEA-EBMT) and a TM-EBMT-SMT hybrid system. In order to achieve optimal performance from the component modules, we finally generated a multi-engine translation output using a confusion network-based system combination (described in Section 4.6).

NEA System: For the NEA system, we appended the extracted parallel NE list described in Section 4.3.1 to the training data.

NEA-EBMT System: In order to build the training corpus for this model, we appended the extracted parallel NE list and also the EBMT parallel phrases (cf. Section 4.3.4) to the training sentence pairs.

TM-EBMT-SMT hybrid system: The TM repository consists of our parallel training data, a parallel NE list (cf. Section 4.3.1) and EBMT parallel phrases (cf. Section 4.3.4). When a new sentence is input to the system for translation, the hybrid MT system first checks whether the translation is already present in the stored TM. If the input sentence is found in TM then the output is immediately returned. If there are no exact matches in the TM, then the system looks for word sequence matches in the TM repository. If a sequence of source tokens in the input sentence is found in the TM repository, then the source sequence is immediately replaced with the target sequence from TM using XML mark-up. The generated sequence serves as an input to the SMT system. The SMT system then produces the final translation output.

System Combination: In our experiments, four MT hypotheses (Baseline, NEA, NEA-EBMT, TM-EBMT-SMT) are fed to the System Combination framework (cf. Section 4.6) from which one is selected as the backbone. The features used to score each arc in the confusion network are word posterior probability, target language model (3-gram, 4-gram), and length penalties. Minimum Error Rate Training (MERT) (Och, 2003) is applied to tune the CN weights.

4.8.3 Baseline Settings

The standard log-linear PB-SMT model serves as our baseline system. For building the baseline system, we experimented with various maximum phrase lengths for the translation model and n–gram settings for the language model. We found that using a maximum phrase length of 7 and a 5-gram language model produced the best results in terms of BLEU scores for our baseline model. We use target (Hindi) side of the training data to build 5-gram language model.

The other experimental settings were: GIZA++ implementation of IBM word alignment model 4 with grow-diagonal-final-and heuristics for performing word alignment and phrase-extraction (Koehn et al., 2003). The reordering model was trained on msd-bidirectional (i.e. using both forward and backward models) and conditioned on both source and target language. The reordering model was built by calculating the probabilities of the phrase pairs being associated with the given orientation such as monotone, swap and discontinuous. The 5-gram target language model with Kneser-Ney smoothing (Kneser and Ney, 1995) was trained using SRILM (Stolcke, 2002). Minimum Error Rate Training (MERT) (Och, 2003) was carried out on a held-out development set (devset). After the parameters were tuned, decoding was carried out on the held out test set.

Note that all the systems described in Section 4.8.2 employ the same PB-SMT settings (apart from the feature weights which are obtained via MERT) as the Baseline system.

4.8.4 Result and Analysis

The system outputs are evaluated with respect to BLEU score (Papineni et al., 2002). For each of the three domains (health, tourism, and general), each of the five source

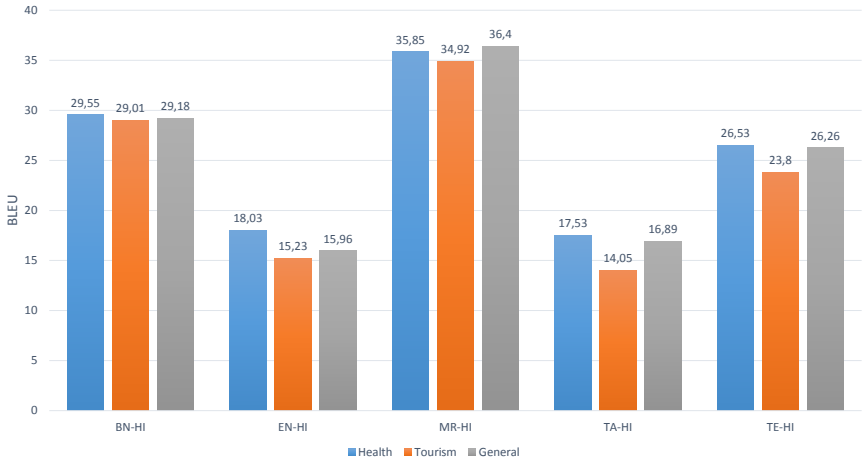


FIGURE 4.3: BLEU scores for all 5 language pairs on all three domains: Health, Tourism, and General

.

SYSTEM	HEA	TOUR	GEN
Baseline	29.29	28.63	29.00
NEA	28.81	28.67	28.80
NEA-EBMT	27.83	27.51	27.73
TM-EBMT-SMT	29.18	28.94	29.14
Combination	**29.55**	**29.01**	**29.18**

TABLE 4.5: Summary of Results on testset data (BLEU score): Bengali→Hindi

languages (Bengali (BN), English (EN), Marathi (MR), Tamil (TA), and Telugu (TE)) was translated into Hindi (HI) using four separate MT systems (Baseline [SMT], NEA, NEA-EBMT, and TM-EBMT-SMT). The four MT outputs are then fed into the System Combination framework to select the final MT output. Thus we ran a total of 60 SMT (four MT systems for five language pairs in three domains) systems (in addition to three System Combination runs per language) for this task.

The optimal system combination hybrid system obtained the overall average BLEU (Papineni et al., 2002) score of 24.61 and average TER (Snover et al., 2006a) score of 57.86.

Figure 4.3 is a graphical representation of the BLEU scores (y-axis) for all five language pairs (x-axis) across all three domains (three shades of bars). Table 4.6 shows evaluation

results for each language pair in three domains. Table 4.7 shows systematic comparison between the baseline and our hybrid system for all the language pairs in three domains. In each language, it was observed that the MT systems perform best on the Health domain. One reason for this could be that the vocabulary size in the health domain was nearly 1.75 times smaller than in the tourism domain, implying less data sparsity and noise.

When comparing across languages (trained on approximately the same size and type of data), Marathi → Hindi was observed to be the best performing system. This was as expected since out of the five languages under study, Marathi is most similar to the target language (Hindi), followed by Bengali. Both Tamil and Telugu belong to the Dravidian family of languages which are significantly different from Hindi and therefore the scores are lower. English, unlike the other languages, has an entirely different grammar (SVO versus SOV in Indian languages). Note that a strong sense of linguistic purism is found in Tamil (Ramaswamy, 1993) which opposes the use of foreign loanwords. Also, the meta-linguistic base of Tamil is Old Tamil unlike most other Indian languages for which it is Sanskrit. These factors are most probably the cause for a lower MT performance on the Tamil–Hindi language pair.

In order to compare and contrast individual performances of each of our MT engines, Table 4.5 shows the BLEU scores for the Bengali → Hindi language pair in all three domains. As observed above, the combination output is the best performing system and outperforms an individual component output by as high as 0.26 BLEU points. One reason for the NEA-EBMT system under-performing is that we did not extract all possible example-based phrase pairs due to time complexity (cf. Chapter 3, Section 3.6.1). The hybrid system displays definite gains over others. Similar relative performance was observed in the other languages.

	BN-HI		**EN-HI**		**MR-HI**		**TA-HI**		**TE-HI**		**Avg. Score per Domain**	
	BLEU	**TER**	**BLEU**	**TER**	**BLEU**	**TER**	**BLEU**	**TER**	**BLEU**	**TER**	**BLEU**	**TER**
Health	29.55	50.01	18.03	68.14	35.85	46.74	17.53	72.1	26.53	53.21	25.498	58.04
Tourism	29.01	49.77	15.23	71.72	34.92	44.93	14.05	69.67	23.8	55.91	23.402	58.4
General	29.18	49.67	15.96	67.86	36.4	45.01	16.89	69.57	26.26	53.53	24.938	57.128
Score per Language Pair	29.247	49.817	16.407	69.24	35.723	45.56	16.157	70.447	25.53	54.217		

TABLE 4.6: Evaluation scores of our system-combination submission in ICON-2014 on 5 language pairs in three domains; Overall average BLEU Score : 24.613 TER: 57.856

		Health		**Tourism**		**General**	
		Hybrid	**Baseline**	**Hybrid**	**Baseline**	**Hybrid**	**Baseline**
BN-HI	**BLEU**	29.55	29.29	29.01	28.71	29.18	29.04
	TER	50.01	49.94	49.77	50.00	49.67	49.43
EN-HI	**BLEU**	18.03	17.85	15.23	15.00	15.96	13.57
	TER	68.14	68.54	71.72	71.32	67.86	75.13
MR-HI	**BLEU**	35.85	34.83	34.92	34.85	36.40	35.56
	TER	46.74	47.82	44.93	45.17	45.01	44.87
TA-HI	**BLEU**	17.53	17.57	14.05	14.02	16.89	16.36
	TER	72.10	70.60	69.67	69.93	69.57	69.47
TE-HI	**BLEU**	26.53	25.54	23.8	23.09	16.26	25.15
	TER	53.21	53.79	55.91	56.14	53.53	54.37

TABLE 4.7: Systematic comparison between system-combination and Baseline system

4.9 Experiments with English–German Data

In the section above we reported findings on our hybrid system in low-resource data settings. To investigate a rich resource data setting, this section examines the performance of our hybrid system (detailed in Section 4.9.2) on the WMT 2015 English → German translation task. The hybrid system is trained on English–German parallel data (cf. Section 4.9.1). The test set for the language pair was drawn from user-generated comments on the news articles[18].

4.9.1 Data

We utilized all the parallel training data provided by the WMT 2015 shared task organizers for English–German translation[19]. The training data include Europarl, News Commentary and Common Crawl. The provided data is noisy and contains some non-German as well as non-English words and sentences. Therefore, we applied a Language Identifier (Shuyo, 2010) on both bilingual English–German parallel data and the monolingual German corpora used for language model training. We discarded those parallel sentences from the bilingual training data which were detected as belonging to some different language by the language identifier. The same method was also applied to the monolingual data.

Further corpus cleaning was carried out first by calculating the global mean ratio of the number of characters in a source sentence to that in the corresponding target sentence and then filtering out sentence pairs that exceed or fall below 20% of the global ratio (Tan and Pal, 2014). We sorted the entire parallel training corpus based on sentence length. Tokenization and punctuation normalization were performed using Moses scripts. In the final step of cleaning, we filtered the parallel training data on maximum allowable sentence length of 100 word tokens and sentence length ratio of 1:2 (either direction). Approximately 36% of the sentences were removed from the total training data during the cleaning process. Table 4.8 shows the parallel data statistics after cleaning and filtering. After filtering, our monolingual data contained approximately 26M sentences.

[18]The test data was collected from WMT 2015 translation task.
[19]`www.statmt.org/wmt15/translation-task.html`

Data	Sentences	Tokens	
		EN	DE
Europarl and news	1,623,546	36,050,888	34,564,547
Common crawl	1,811,826	37,456,978	35,172,840
Total	3,435,372	73,507,866	69,737,387

TABLE 4.8: Parallel training data statistics after cleaning

4.9.2 Hybrid System

A hybrid approach was investigated by combining multiple knowledge sources such as NEA, EBMT Phrases and MWEs and following different strategies. We implemented several systems, namely:

(1) Baseline PB-SMT (cf. Section 4.9.4),

(2) Baseline PB-SMT with NE alignment (NEA) (cf. Section 4.3.1, Method 2),

(3) NEA with EBMT phrase extraction (NEA–EBMT) (cf. Section 4.3.1, 4.3.4),

(4) NEA with EBMT phrase extraction and single-tokenized MWE[20] (NEA–EBMT–MWE) and

(5) LM–NEA–EBMT–MWE system-combination (see Section 4.9.2.1).

The baseline SMT system is trained on the cleaned English-German parallel corpus. The NEA system makes use of NE aligned parallel data as additional parallel examples. Similarly, EBMT phrase pairs as well as NE aligned data are also used as additional training examples in the NEA–EBMT system. The NEA–EBMT–MWE system is very similar to the above mentioned NEA–EBMT system, the only difference being that the identified source side English MWEs are converted into single tokens for NEA–EBMT–MWE. In order to achieve optimal performance from the component modules, we finally generated a system-combination translation output using confusion network-based system combination (cf. Section 4.9.3.1).

[20]We extracted MWE on the English side and performed single-tokenization. The extraction method is described in Section 4.3.2

4.9.2.1 LM-NEA-EBMT-SMT hybrid system

In this system, we experiment with the above-described models with varying sizes of monolingual data. We experimented with four folds of monolingual data to train the Language Models (LMs):

- LM_1: Only using the target (i.e. German) side of the parallel training data (where the target data size is L in terms of number of sentences) for language modeling
- LM_2: L + double size of L, collected from the cleaned monolingual corpus
- LM_3: L + triple size of L from the cleaned monolingual corpus
- LM_4: L + all the cleaned monolingual data

Therefore, there were 16 different systems (four systems, i.e., Baseline, NEA, NEA–EBMT and NEA–EBMT–MWE, each with four LM settings) output available for system combination.

4.9.3 MIRA-MERT coupled tuning

The Minimum Error Rate Training (MERT) (Och, 2003) method has been the most popular method used for parameter tuning in SMT; it has some nice properties such as simplicity, effectiveness and speed. However, it does not scale well for systems with large numbers of features. The Margin Infused Relaxed Algorithm (MIRA) (Cherry and Foster, 2012), an alternative tuning method, works well with a large number of features, although, the optimization problem in MIRA is much more complicated than MERT. We linearly interpolate the weights learned from these two optimization methods w_{mira} and w_{mert} as in Equation 4.11.

$$w_t = \lambda w_{mira} + (1 - \lambda) w_{mert} \tag{4.11}$$

We calculate the λ parameter ($0 < \lambda < 1.0$) based on the iteration in which the BLEU score on the development set is highest. We apply the λ value for reranking the hypothesis based on weights of w_{mira} and w_{mert} on the n-best hypothesis generated from the test set. The results of our experiments are presented in Table 4.7.2.1 (cf. System 3).

4.9.3.1 System Combination

System Combination is a technique which combines translation hypotheses (outputs) produced by multiple MT systems. We applied a system combination method (cf. Section 4.6) on the outputs of the different MT systems described earlier.

4.9.4 Baseline Settings

We used the standard log-linear PB-SMT model as our baseline. For building the baseline system, we used a maximum phrase length of 7 and a 5-gram language model. The other experimental settings include word alignment using SymGIZA++ aligner (Junczys-Dowmunt and Szał, 2012)[21] and the phrase-extraction following Koehn et al. (2003). The reordering model was trained on hier-mslr-bidirectional (i.e. using both forward and backward models) and conditioned on both source and target language. The reordering model was built by calculating the probabilities of the phrase pairs being associated with the given orientation such as monotone, swap and discontinuous. The 5-gram target language model was trained using KENLM (Heafield, 2011). Parameter tuning was carried out using both k-best MIRA and MERT on the held-out development set. After the parameters were tuned, decoding was carried out on the held out test set.

Systems	**BLEU**	**BLEU(Cased)**	**TER**
Baseline	16.7	16.2	89.6
System 1	18.1	17.5	88.2
System 2	18.1	17.6	87.8
System 3	19.0	18.4	85.3
System 4	20.0	19.5	84.1
System 5	20.3	19.7	83.8
System 6	20.7	20.2	83.5
System 7	**22.6**	**22.1**	**82.3**

TABLE 4.9: Systematic comparison between system-combination (System 7), six best performing individual systems and Baseline system

[21]SymGIZA++ is a modified version of GIZA++ word alignment models by updating the symmetrisizing models between chosen iterations of the original word alignment training algorithms.

4.9.5 Results and Analysis

As described in Section 4.9.2.1, we developed sixteen different systems. Instead of using all these sixteen different systems, we used only the six best performing systems for system combination. Performance is measured on the development set. Table 4.9 reports the final evaluation results obtained on the test dataset. The six best individual systems are as follows:

- System 1: NEA–EBMT (selective high frequency EBMT phrases) with baseline PB-SMT settings and LM_1.
- System 2: System 1 experimental settings + single tokenized source MWEs (i.e. NEA–EBMT–MWE).
- System 3: System 2 with MIRA-MERT coupled tuning (cf. Section 4.9.3).
- System 4: System 3 with LM_2.
- System 5: System 3 with LM_3.
- System 6: System 3 with LM_4.

System 6 provides the individual best performing system. System combination (System-7 in Table 4.9) of the six best performing individual systems brings considerable improvements over each of the individual systems. A hybrid system (System 6) with NE alignment, EBMT phrases, single-tokenized source MWEs, and MIRA-MERT coupled tuning (cf. Section 4.9.3) results in the best performing system. However, confusion network-based system combination outperforms all the individual MT systems. The fact that the systems were tuned with BLEU scores may be one of the reasons behind the poor TER scores produced by the systems. This work was submitted to WMT 2015 translation task. Our hybrid system ranked 10^{th} among 24 submissions.

4.10 Conclusions and Future Work

The chapter presented how effective pre-processing of NEs and MWEs in the parallel corpus and direct or indirect incorporation of their alignments in the word alignment model

can improve SMT system performance. In data driven approaches to MT, specifically for scarce resource language pairs, this approach can help to improve state-of-art MT quality as well as the word alignment quality.

The Indian languages/English to Hindi (cf. Section 4.8) and English to German (cf. Section 4.9) hybrid systems with NE alignment, EBMT phrases, single-tokenized source MWEs, and MIRA-MERT coupled tuning resulted in the best performance. However, confusion network-based system combination outperforms all the individual MT systems. The fact that the systems were tuned with BLEU scores may be one of the reasons behind the poor TER scores produced by the systems.

We also presented a method of source chunk pre-ordering based on word alignment. Source chunks are reordered based on their associations with the target words and the target word order. The testset is reordered using monolingual PB-SMT built on the original source training data and the reordered source training data. Our experiments showed that word alignment based source chunk pre-ordering is more effective than word alignment based source word pre-ordering and tree-based reordering; and it produced statistically significant improvements on both. On manual inspection we found significant improvements in terms of word alignments. This method also reduces the data sparsity problem. The method presented in the paper has the advantage that it does not require any language specific tools like parsers except a chunker for the source language.

This chapter also reported research on integrating hybrid word alignment in FSBSMT. Experimental results on an English–Bengali dataset show that FSBSMT with Berkeley alignment results in a large improvement (69.83% relative, 8.75 absolute BLEU points) over the state-of-the-art HPBSMT baseline. Systems like HPBSMT which work only with 1-best parse tree may suffer from parsing errors. FSBSMT alleviates this problem by considering a packed forest of k-best parses.

Additional integration of prior aligned named entities and high frequency EBMT phrases into the proposed system also brings about further improvements. The enhanced system provides 78.5% relative (9.84 absolute BLEU points) improvement over the baseline HPBSMT system and 5.12% relative improvement (1.09 absolute BLEU points) over an FSBSMT baseline system with Berkeley alignment.

We introduced two research questions (i.e., RQ2 and RQ3) at the beginning of this chapter. The use of parallel/pseudo-parallel NEs, MWEs and parallel EBMT phrases as additional training examples successfully improved the MT performance for Indian languages to Hindi, English–Hindi and English–German language pairs. This addresses RQ2, i.e., "how can SMT better profit from the existing training data?". Prior reordering also provides some benefits in terms of both word alignment quality and MT performance; this also covers RQ2 to some extent. RQ3, i.e., "What could improved hybrid implementations of MT be like?" is an open research question. We provided a proposal in terms of a two level of hybridization architecture as follows:

- Hybrid word alignment: Alignment combination of multiple word alignments provided by different statistical aligners and a rule based aligner.
- System combination: Combination of different MT engines developed using different MT methodologies each of which operates below the hybrid word alignment architecture.

We successfully showed that the resulting hybrid MT system outperforms all the individual component systems (Pal et al., 2014c,a, 2015a, 2016a).

In future, we would like to apply a similar methodology with other alignment combination methods and compare between them. We will also focus on improving our hybrid word alignment model by considering the strength of alignment points given by the various word alignment models. For multi-engine system combination, we would like to incorporate neural machine translation as an another component engine into our hybrid framework. Finally, tight coupling of SMT and NMT, by taking advantage of the translations of longer phrases in PB-SMT and better context dependent translations in NMT, is a very difficult proposition which we would like to explore.

Chapter 5

Automatic Post Editing

For many applications the performance of state-of-the-art MT systems is useful but far from perfect. MT technologies have gained wide acceptance in the localization industry. Computer aided translation (CAT) followed by post-editing has become the de-facto standard in large parts of the translation industry which has resulted in a surge of demand for professional post-editors. This, in turn, has resulted in substantial quantities of PE data which can be used to develop automatic post-editing (APE) systems. This chapter is focused on addressing two research questions: **RQ4:** *How can we build an effective automatic post-editing system which can improve the translation quality of the first-stage MT system?* and **RQ5:** *To what extent is an APE system able to reduce final post-editing effort in terms of increasing productivity?*

APE systems assume the availability of source language input text (SL_{ip}), target language MT output (TL_{mt}) and target language PE data (TL_{pe}). An APE system can be modelled as an MT system between $SL_{ip}\_TL_{mt}$ and TL_{pe}. However, if we do not have access to SL_{ip}, but have sufficiently large amounts of parallel TL_{mt}—TL_{pe} data, we can still build an APE model between TL_{mt} and TL_{pe}.

Translations provided by state-of-the-art MT systems suffer from different types of errors including incorrect lexical choice, word ordering, word insertion, word deletion, etc. The APE work presented in this chapter is an effort to improve the MT output by rectifying some of these errors. For this purpose we adopt various strategies, including:

1. A hybrid word alignment model integrated within two different statistical machine translation (SMT) frameworks: Phrase-Based Statistical APE (PB-SAPE) and Hierarchical PB-SAPE (HPB-SAPE). Both are trained on TL_{mt} and TL_{pe}.

2. An operation sequence Model (OSM). We adopt the OSM model for MT (Durrani et al., 2011, 2015) to monolingual APE.

3. A deep neural network (DNN) based approach: our neural network model of APE (NNAPE) is based on a bidirectional recurrent neural network (RNN) model and consists of an encoder that encodes an MT output into a fixed-length vector from which a decoder provides a post-edited (PE) translation.

4. A system combination framework. The focus of this study is twofold – to study how existing word alignment techniques and system combination frameworks can be intelligently used to improve monolingual APE, and whether the improvements in APE measured in terms of automatic evaluation metrics translate to measurable productivity gains in human post-editing in commercial translation workflows.

   Core parts of the research presented in this chapter have been published in (Pal et al., 2015c, 2016b,c,f)

Figure 5.1 schematically shows the research presented and the research questions addressed in this Chapter.

5.1 Introduction

In the context of MT, "post-editing" (PE) is defined as the corrections performed by humans on the translations produced by an MT system (Veale and Way, 1997), often with minimal amount of manual effort (TAUS Report, 2010) and as a process of modification rather than revision (Loffler-Laurian, 1985).

The quality of translations produced by MT systems has improved substantially over the past few decades. This is particularly noticeable for some language pairs (e.g., English–Italian) and for some specific domains (e.g., technical documentation). However, some language pairs, e.g., English–German, have proved to be difficult for MT. Texts produced by MT systems are now widely used in the translation and localization industry. MT

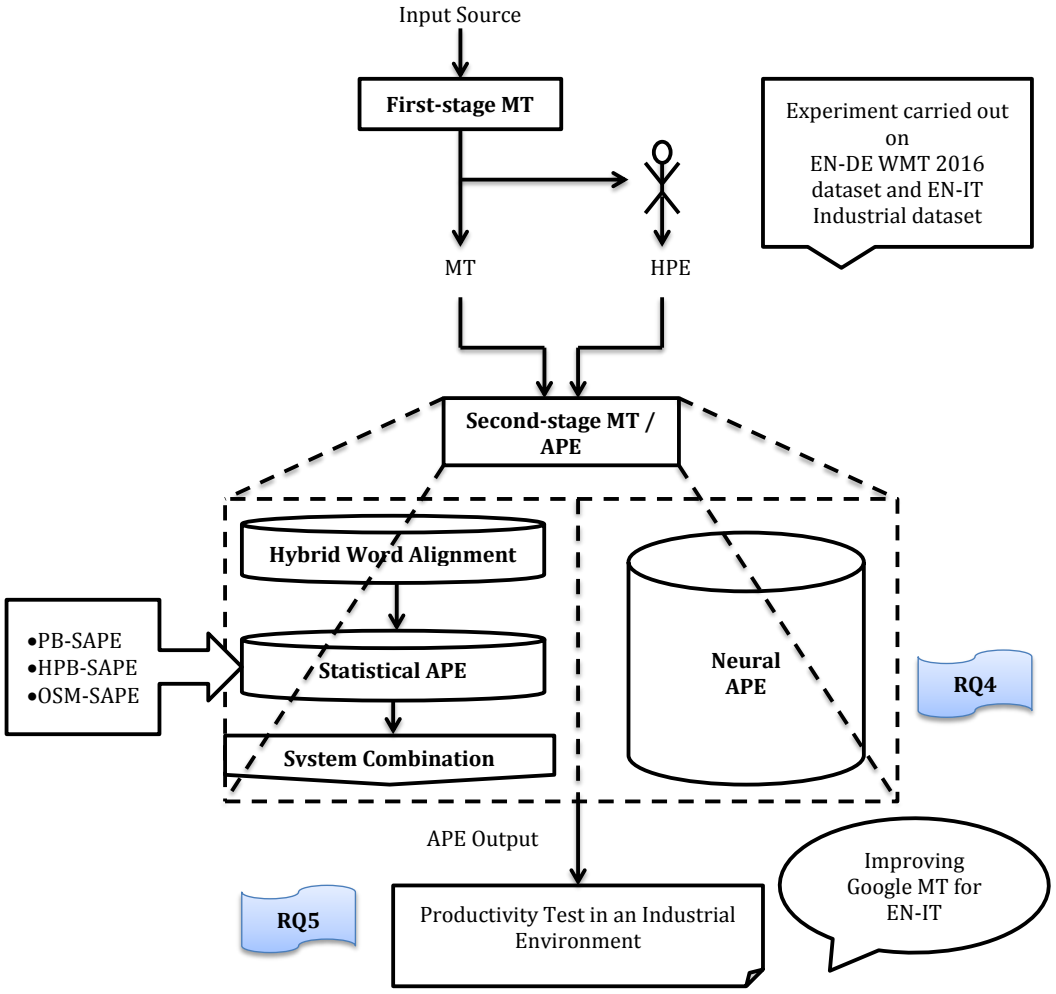


FIGURE 5.1: Schematic design of the research and the research questions presented in this Chapter.

output is post-edited by professional translators and MT has become an important part of the translation workflow. A number of studies confirm that post-editing MT output can improve translators performance in terms of productivity and it may also impact translation quality and consistency (Guerberof, 2009; Plitt and Masselot, 2010; Zampieri and Vela, 2014).

The ultimate goal of MT systems is to provide translations that can be post-edited with the least amount of effort by human translators. One of the strategies to improve MT output quality is to apply APE methods (Knight and Chander, 1994; Simard et al., 2007a,b). APE methods work under the assumption that some errors in MT systems (e.g., incorrect lexical choice, wrong word orderings, erroneous word insertion, deletion) are recurrent and can be corrected automatically during a post-processing stage thus providing better output to be post-edited by human experts. APE methods are applied before human post-editing takes place and, if effective, these methods can increase translators' productivity.

MT systems primarily make two types of errors – lexical and reordering errors. However, due to the statistical and probabilistic nature of modelling in SMT, one of the currently dominant MT paradigms, it is non-trivial to rectify these errors in the SMT models themselves. Human post-edited data are often used in incremental MT frameworks as additional training material. However, often this does not fully exploit the potential of these rich PE data: e.g., PE data may just be drowned out by a large SMT model. An APE system trained on human post-edited data can serve as an MT post-processing module which can improve overall performance. An APE system can be considered as an MT system, translating predictable error patterns in MT output to their corresponding corrections. In order to automatically post-edit, in part of our research we adopt Phrase-Based (PB-SMT), Hierarchical Phrase-Based (HPB-SMT) and operation sequence Model (OSM) for SMT to build our Statistical APE (SAPE) system. Because in the OSM model the translation and reordering operations are coupled in a single generative story, the reordering decisions may depend on preceding translation decisions and translation decisions may depend on preceding reordering decisions. Our OSM-based SAPE model provides a natural reordering mechanism and deals with both local and long-distance reorderings consistently. Furthermore, we also develop a DNN-based monolingual neural

MT (NMT) APE system as well as a system combination based approach to test the potential of an APE model in commercial environments.

NMT (Kalchbrenner and Blunsom, 2013; Cho et al., 2014b) is a newly emerging approach to MT. The motivation behind the use of a DNN based approach in our APE task is that on the one hand DNNs represent language in a continuous vector space which eases the modelling of semantic (rather than surface) similarities (or distance) between phrases or sentences, and on the other hand DNNs can also consider contextual information, e.g., utilizing all available history information in deciding the next target word, which is not an easy task to model with standard SMT-based APE systems (Simard et al., 2007b; Pal, 2015).

Unlike the SAPE systems found in the literature (Simard et al., 2007a,b; Pal, 2015; Pal et al., 2015c), where each individual component (e.g., word alignments, phrase alignments, language models) is estimated separately, our NNAPE system builds and trains a single large neural network that accepts a 'draft' translation (TL_{mt}) and outputs an improved translation (TL_{pe}).

Another direction of APE research presented in this chapter explores the use of system combination in APE. System combination in MT has been studied extensively (Matusov et al., 2006; Du et al., 2009; Pal et al., 2014c), except in the context of APE. Here we use system combination architectures at three different levels: (i) sequential combination between the first-stage system and APE, (ii) combination of multiple alignments at the level of APE and (iii) parallel combination of APE systems (including the first-stage MT system). More precisely, our approach makes use of a hybrid implementation of multiple alignments combined with Phrase-Based SAPE (PB-SAPE) and hierarchical PB-SAPE (HPB-SAPE) and a system combination framework (a multi-engine pipeline) that combines the best translations from the enhanced PB-SAPE, HPB-SAPE and the raw MT output. System combination and hybrid word alignment strategies are commonly used in MT. However, to the best of our knowledge, the work presented in this chapter is the first approach to APE that uses system combination and hybrid word alignment methods in APE.

5.2 Related Work

APE approaches cover a wide methodological range. Simard et al. (2007a,b) applied SMT for post-editing that handles the repetitive nature of errors typically made by rule-based MT (RBMT) systems. The SMT APE system was trained on the output of a rule-based MT system as the source language and reference human translations as the target language. This APE system based on PB-SMT, was able to correct systematic errors produced by the RBMT system and reduce post-editing effort. The approach achieved large improvements in performance not only over the baseline rule-based system but also over a similar PB-SMT used in a standalone mode. Denkowski (2015) proposed a method for real time integration of post-edited MT output into the translation model. He extracted a grammar for each input sentence and applied it to the model. Rosa et al. (2012) and Mareček et al. (2011) applied a rule-based approach to APE for English–Czech MT outputs on the morphological level. They used 20 hand-written rules based on the most frequent errors encountered in translation. The method efficiently corrects morpho-syntactic categories of a word such as number, case, gender, and person as well as dependency labels. Intuitively, integration of source-language information in APE is useful to improve the APE performance. Béchara et al. (2011) proposed "source-context aware" APE. The source side of the parallel training data for APE is modified with the automatically created word alignments between source and MT. This technique results in a new source language consisting of source–MT joint token pairs. Chatterjee et al. (2015b) examined the potentiality of two different statistical APE methods: (Simard et al., 2007b) and (Béchara et al., 2011). They systematically tested these two different APE approaches in controlled conditions over several language pairs and analyzed them. They found that inclusion of source language information into statistical APE results in consistent improvements in all language pairs. To overcome data sparsity issues, Chatterjee et al. (2015a) proposed a pipeline where the best language model and pruned phrase table are selected through task-specific dense features.

While various automatic or semi-automatic post-processing techniques to implement corrections of repetitive errors have been developed, the overall resulting MT output usually still needs to be post-edited by humans in order to produce publishable quality translations (Roturier, 2009; TAUS/CNGL Report, 2010). Even though MT output needs human

PE, it is often faster and cheaper to post-edit MT output than to perform human translation from scratch. In some cases, recent studies have even shown that the quality of MT plus PE can exceed the quality of human translations (Fiederer and OBrien, 2009; Koehn, 2009; De Palma and Kelly, 2009) as well as increase productivity. Aimed at cost-effective and time saving use of MT, the PE process needs to be further optimized (TAUS/CNGL Report, 2010).

System combination is a technology where multiple translation outputs from potentially very different MT systems are combined using e.g., confusion networks (Matusov et al., 2006). The confusion networks are built using backbone selection using either multiple hypotheses as backbones (Leusch and Ney, 2010) or a single backbone (Rosti et al., 2007b; Du et al., 2009) using TER (Snover et al., 2006a) or BLEU (Papineni et al., 2002). These alignment metrics select the hypothesis that agrees most with the other hypotheses on average. System combination can improve translation quality significantly which motivated us to apply the system combination strategy for the APE task.

Parra Escartín and Arcedillo (2015a,b) studied the impact of various factors and methods in APE on productivity gains. However, those studies were not conducted to observe PE effort in commercial environments.

Recently, a number of papers have presented the application of neural networks in MT (Kalchbrenner and Blunsom, 2013; Cho et al., 2014a,b; Bahdanau et al., 2015). These approaches typically consist of two components: an **encoder** encodes a source sentence and a **decoder** decodes into a target sentence.

In this chapter we present different approaches to APE – PB-SAPE, HPB-SAPE, OSM based SAPE and neural network based APE (NNAPE). Our NNAPE model is inspired by the MT work of Bahdanau et al. (2015) which is based on bidirectional recurrent neural networks (RNN). Unlike Bahdanau et al. (2015), we use long-short term memory (LSTM) (Hochreiter and Schmidhuber, 1997) rather than gated recurrent units (GRU) (Cho et al., 2014b) as hidden units. RNNs allow processing of arbitrary length sequences. However, they are susceptible to the problem of vanishing and exploding gradients (Bengio et al., 1994). To tackle vanishing gradients in RNNs, two architectures are generally used: GRU and LSTM. According to empirical studies (Chung et al., 2014; Józefowicz et al., 2015) the two architectures yield comparable performance. GRUs tend to train faster than LSTMs. On the other hand, given sufficient amounts of training

data, LSTMs may lead to slightly better results. Since our task is monolingual and we have more than 200K sentence pairs (English–Italian Google translate output and the corresponding PE translation) for training, we use a full LSTM (as the hidden units) to model our NNAPE system. To the best of our knowledge the NNAPE work presented in this chapter is the first approach that uses neural networks for APE.

5.3 Hybrid Word Alignment

Previous research in MT demonstrated that a combination of information coming from multiple word alignment models can improve translation quality. This can be achieved in different ways, e.g., by combining two bidirectional alignments (Och, 2003; Koehn et al., 2003; DeNero and Macherey, 2011), combining an arbitrary number of alignments (Tu et al., 2012; Pal et al., 2013a), or by constructing weighted alignment matrices over 1-best alignments from multiple alignments generated by different models (Liu et al., 2009; Tu et al., 2011). This motivated us to explore the alignment combination model for APE.

Our hybrid word alignment method combines word alignments produced by three different statistical word alignment methods: (i) GIZA++ (Och and Ney, 2003a) word alignment with the grow-diag-final-and (GDFA) heuristic (Koehn, 2010), (ii) Berkeley word alignment (Liang et al., 2006), and (iii) SymGiza++ (Junczys-Dowmunt and Szał, 2012) word alignment, as well as two different edit distance based word aligners based on TER (Translation Edit Rate) (Snover et al., 2006a) and METEOR (Lavie and Agarwal, 2007). We follow (Pal et al., 2013a) in combining word alignment tables. However, we additionally used 3-word (i.e., trigram) consistent phrases to generate more alignment links (cf. Section 5.3.3). We integrate the word alignments obtained with this hybrid model into PB-SAPE (Pal et al., 2015c) and HPB-SAPE (Pal, 2015).

5.3.1 Statistical Word Alignment

GIZA++ is a statistical word alignment tool which implements IBM models 1–5, an HMM alignment model, as well as the IBM-6 model for covering many-to-many alignments. The Berkeley word aligner uses an extension of Cross Expectation Maximization and is jointly trained with HMM models. SymGiza++ is a modification of GIZA++. It

modifies the counting phase of each model of GIZA++ in order to allow for updates of the symmetrisized models between the iterations of the original training algorithm. SymGiza++ computes symmetric word alignment models with the capability of taking advantage of multi-processor systems.

5.3.2 Edit Distance-Based Word Alignment

We use two different edit distance style word aligners where alignments are based on edit distance style MT evaluation metrics – TER and METEOR.

5.3.2.1 TER Alignment:

TER is an edit distance based automatic MT evaluation metric that measures the ratio between the number of edit operations that are required to turn a translation hypothesis H (i.e., the MT output) into a reference translation R (in this case the PE translation) to the total number of words in R. The allowable edit operations include insertion (I), substitution (S), deletion (D) and phrase shifts (Sh). As a by-product of finding the minimum edit distance, TER also produces an alignment between the hypothesis and the reference. TER is computed as in equation 5.1.

$$TER(H,R) = \frac{(I + D + S + Sh) * 100\%}{total\ number\ of\ words\ in\ R} \tag{5.1}$$

For the monolingual SAPE task, we make use of TER alignment as a potential alignment between TL_{mt} and TL_{pe}. The TER alignment between a TL_{pe} and TL_{mt} is illustrated in the example given below. The vertical bar '|' represents a match and I, D and S represent three post-editing operations – insertion, deletion and substitution, respectively.

TL_{mt}:	w_1	w_2	w_3	ϵ	w_4	w_5	w_6	w_7	w_8	w_9
	|	D	S	I	|	|	S	|	|	|
TL_{pe}:	$\bar{w_1}$	ϵ	$\bar{w_2}$	$\bar{w_3}$	$\bar{w_8}$	$\bar{w_4}$	$\bar{w_5}$	$\bar{w_6}$	$\bar{w_7}$	$\bar{w_9}$

5.3.2.2 METEOR Alignment

METEOR (Lavie and Agarwal, 2007) is an automatic MT evaluation metric which provides an alignment between a H and R. Given a pair of strings H and R to be compared, the alignment is a mapping between the words in H and R, which is built incrementally by three sequences of word-mapping modules:

(i) **E**xact: maps if the words are exactly the same.

(ii) **P**orter stem: maps if the words are the same after stemming.

(iii) **W**N synonymy: maps if the words are synonyms in WordNet. (Miller, 1995).

If multiple alignments exist, METEOR selects the alignment for which the word order in the two strings is most similar (i.e. the alignment which has the fewest number of crossing alignment links). The final alignment is produced as the union of alignments from the three stages (i.e., Exact, Porter stem and WN synonymy).

5.3.3 Producing Additional Alignments

To generate additional alignment points between parallel sentence pairs, we perform phrase extraction Koehn et al. (2003)[1] between T_{mt} and T_{pe}. We extract all phrase pairs, T_{mt} phrase (e) and T_{pe} phrase ($\bar{e}$), that are continuous and consistent with the edit distance based monolingual alignments. This phrase extraction process is performed individually for both TER and METEOR based alignments. A phrase pair $(e, \bar{e})$ is consistent with alignment a if Equation 5.2 is satisfied.

$$(\forall w_i \in e : (w_i, x) \in a \wedge x \in \bar{e}) \wedge \ (\forall \bar{w}_i \in \bar{e} : (y, \bar{w}_i) \in a \wedge y \in e) \tag{5.2}$$

Unaligned words in a phrase pair are aligned to all the phrase internal words in the other language. Figure 5.2 depicts the process of generating additional alignments where the solid links represent edit distance based alignments and the dotted links represent the newly established alignments. The newly established alignment points are added to the corresponding (i.e., TER or METEOR) alignment matrix.

[1]For this task, we use 3-words phrases.

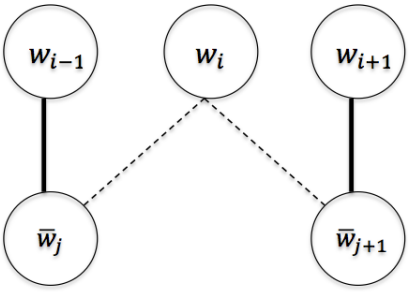


FIGURE 5.2: Producing additional alignments (w_i–$\bar{w}_j$, w_i–$\bar{w}_{j+1}$)

5.3.4 Alignment Hybridization

The alignment hybridization method follows the following heuristic. We consider either of the alignments generated by GIZA++ with the grow-diag-final-and heuristic (Koehn, 2010) (a_1), Berkeley aligner (a_2), or SymGiza++ (a_3) as the standard alignment since edit distance based alignments, TER (a_4) and METEOR, fail to align many words in the monolingual MT–PE parallel sentences. From the five alignments a_1–a_5, we compute the alignment combination as follows.

Algorithm 4: Producing alignment combination

- **Step 1:** Choose a standard alignment (S_a) from a_1, a_2 or a_3 ($S_a \leftarrow$ Empirically best preforming aligner among the individual aligners (a_1, a_2 or a_3)).
- **Step 2:** Produce a combined alignment $S_c = S_a \cup (a_2 \cap a_3)$, if a_1 is considered as S_a.
- **Step 3:** Delete all the alignment points $a_{ij} \in S_c$ such that $\exists a_{ik} \in a_4 \cup a_5$ where $j \neq k$.
- **Step 4:** Update S_c as $S_c = S_c \cup a_4 \cup a_5$.

5.4 Phrase-Based SAPE

Our PB-SAPE system is modelled similar to the PB-SMT, in which the post-edited translation (TL_{pe}), $e_1^L = e_1...e_i...e_L$ for a given MT translation (TL_{pe}), $f_1^J = f_1...f_j...f_J$ is

chosen to maximize Equation (5.3):

$$argmax_{L,e_1^L} P(e_1^L|f_1^J) = argmax_{L,e_1^L} P(f_1^J|e_1^L) * P(e_1^L) \tag{5.3}$$

where $P(f_1^J|e_1^L)$ is the SAPE translation model and $P(e_1^L)$ the target language model. In log-linear phrase-based SAPE, the posterior probability is directly modeled as a log-linear combination of features (Och and Ney, 2003a), involving M translation features, and the language model, as in Equation (5.4):

$$logP(e_1^L|f_1^J) = \sum_{m=1}^{M} \lambda_m h_m(f_1^J, e_1^L, s_1^k) + \lambda_{LM} logP(e_1^L) \tag{5.4}$$

where $s_1^k = s_1 \ . \ . \ . \ s_k$ denotes a segmentation of the (TL_{mt}) and (TL_{pe}) sentences respectively into the sequences of phrases $(\hat{e}_1^k = \hat{e}_1 \ . \ . \ . \ \hat{e}_k \ )$ and $(\hat{f}_1^k = \hat{f}_1 \ . \ . \ . \ \hat{f}_k \ )$ such that (we set $i_0 = 0$) in Equation (5.5):

$$\begin{aligned} &\forall 1 \leq k \leq K, \\ &s_k = (i_k, b_k, j_k), \\ &\hat{e}_k = e_{i_{k-1}+1} ... e_{i_k} \\ &\hat{f}_k = f_{b_k} ... f_{j_k} \end{aligned} \tag{5.5}$$

and each feature $\hat{h}_m$ in Equation (5.4) can be rewritten as in Equation (5.6):

$$h_m(f_1^J, e_1^L, s_1^k) = \sum_{k=1}^{K} \hat{h}_m(\hat{f}_k, \hat{e}_k, s_k) \tag{5.6}$$

where $\hat{h}_m$ is a feature that applies to a single phrase-pair. It thus follows in Equation (5.7):

$$\sum_{m=1}^{M} \lambda_m \sum_{k=1}^{K} \hat{h}_m(\hat{f}_k, \hat{e}_k, s_k) = \sum_{k=1}^{K} \hat{h}(\hat{f}_k, \hat{e}_k, s_k) \tag{5.7}$$

where $\hat{h} = \sum_{k=1}^{K} \lambda_m \ \hat{h}_m$.

5.5 Hierarchical Phrase-based SAPE

Hierarchical PB-SMT is based on Synchronous Context Free Grammar (SCFG) (Aho and Ullman, 1969). SCFG rewrites rules on the right-hand side with aligned pairs (Chiang, 2007).

$$X \rightarrow < \gamma, \alpha, \sim > \tag{5.8}$$

where X represents a non-terminal, γ, α represent sequences of both terminal and non-terminal strings and $\sim$ represents an one-to-one correspondence between occurrences of non-terminals appearing in γ and α.

The weight of each rule is defined as:

$$w(X \rightarrow < \gamma, \alpha, \sim >) = \prod_i \phi_i(X \rightarrow < \gamma, \alpha, \sim >)^{\lambda_i} \tag{5.9}$$

where ϕ_i is a feature defined for each rule and λ_i is the weight of ϕ_i. The features are associated with four probabilities: phrase probabilities $P(\gamma|\alpha)$, $P(\alpha|\gamma)$, lexical weights $P_w(\gamma|\alpha)$, $P_w(\alpha|\gamma)$ (estimate how well the words in α translate the words in γ) and a phrase penalty $exp(-1)$.

There exist two additional rules called “glue rules” or “glue grammar”:

$$S \rightarrow < SX, SX > \tag{5.10}$$

$$S \rightarrow < X, X > \tag{5.11}$$

These rules are used when no rules match or the span exceeds a certain length. These rules simply monotonically connect translations of two adjacent blocks together.

The weight of the above rules is defined as:

$$w(S \rightarrow < SX, SX >) = exp(-\lambda_g) \tag{5.12}$$

where λ_g controls the model's preference for hierarchical phrases over a serial combination of phrases.

The weight ($w(d_g)$) of the derivation grammar (d_g) for generated source (f_d) and target (e_d) strings, is the product of the weights of the rules used in translation $w(r)$, language model probability P_{lm} and the word penalty $exp(-\lambda_{wp}|e|)$ with some control over the length of the target output (e). The representation of d_g can be defined as a triplet $< r, i, j >$, where, r is the grammar rule to rewrite a non-terminal that spans $f_{d_i}^{j}$ on the source side.

$$w(d_g) = \prod_{<r,i,j> \in d_g} w(r) \times P_{lm}^{\lambda_{lm}} \times exp(-\lambda_{wp}|e|) \tag{5.13}$$

5.6 OSM based APE

Our OSM based SAPE system is based on an n-gram operation sequence model (Durrani et al., 2015) which integrates translation and reordering operations into the phrase-based APE system. Traditional PB-SMT (Koehn et al., 2003) provides a powerful translation mechanism which can be directly used to model a PB-SAPE system (Simard et al., 2007a,b; Pal et al., 2015c) using TL_{mt}–TL_{pe} as the parallel training corpus. Like PB-SMT, PB-SAPE is subject to drawbacks such as ignoring some dependencies among neighboring phrases and a limited capability of handling discontinuous phrases. Our OSM-APE system is based on the phrase-based n-gram APE model. However, the reordering approach is essentially different: it considers all possible orderings of phrases instead of pre-calculated orientations. The model represents the automatic post-editing (monolingual translation) process as a linear sequence of operations such as the lexical generation of post-edited translation and their orderings. The translation and reordering decisions are conditioned on n previous translation and reordering decisions. The model is able to consistently model both local and long-range reorderings. Traditional OSM based MT models use a sequence of three operations:

- Generation of a sequence of source and/or target words.
- For reordering operations, insertion of gaps at explicit target positions.

- Forward and backward jump operations

The sequence operation is based on n-gram models. The probability of the n^{th} operation depends on the $n-1$ preceding operations. For generating TL_{pe} from a given TL_{mt}, the decoder provides a sequence of hypotheses H: h_1,...,h_n and the APE model estimates the probability $p(TL_{mt}, TL_{pe})$ (cf. Equation 5.14), using a sequence of I operations $o_1, ...o_I$ given m words[2] of context.

$$p(TL_{mt}, TL_{pe}) \approx \prod_{i=1}^{I} p(o_i | o_{i-m+1} ... o_{i-1}) \tag{5.14}$$

The decoder searches for the best translation (pe^*) as in Equation 5.15 using the language model $p_{lm}(TL_{pe})$,

$$pe^* = argmax_{\{TL_{pe}\}} \frac{p(TL_{mt}, TL_{pe})}{p_{pr}(TL_{pe})} \times p_{lm}(TL_{pe}) \tag{5.15}$$

where $p_{pr}(pe) \approx \prod_{i=1}^{I} p(w_i | w_{i-m+1} ... w_{i-1})$, is the prior probability that marginalizes the joint probability $p(TL_{mt}, TL_{pe})$. The model is then represented in a log-linear approach (Och and Ney, 2003a) (in Equation 5.16) that makes it useful to incorporate standard features along with several novel features that improve accuracy.

$$pe^* = argmax_{\{TL_{pe}\}} \sum_{i=1}^{I} \lambda_i h_i(TL_{mt}, TL_{pe}) \tag{5.16}$$

λ_i is the weight associated with the feature $h_i(TL_{mt}, TL_{pe})$: $p(TL_{mt}, TL_{pe})$, $p_{pr}(TL_{pe})$ and $p_{lm}(TL_{pe})$. Apart from this, eight additional features were included in the log-linear model:

1. Length penalty: Length penalty based on the length of TL_{pe} in terms of number of words.

2. Deletion penalty.

3. Gap bonus: Total number of gaps inserted to produce the TL_{pe} sentence.

4. Open gap penalty: Number of open gaps; this penalty controls how quickly the gap was closed.

[2]We use a 6-gram model trained on SRILM-Toolkit (Stolcke, 2002)

5. Distortion: Distance based reordering that is similar to PB-SMT.

6. Gap distance penalty: The gap between TL_{mt} and TL_{pe} sentences generated during the generation process.

7. Lexical features: TL_{mt}–TL_{pe} and TL_{pe}–TL_{mt} lexical translation probabilities (Koehn et al., 2003).

5.7 System Combination for APE

The system combination framework selects the best translation hypothesis from multiple hypotheses produced by different systems. In order to apply the system combination framework on the translations produced by our SAPE systems and the baseline MT system[3], we implemented Minimum Bayes Risk (MBR) coupled with the Confusion Network (MBRCN) framework as described in Du et al. (2009). The MBR decoder (Kumar and Byrne, 2004) selects for each input sentence the best of the three system outputs by minimizing the BLEU (Papineni et al., 2002) loss. This output is known as the backbone. A confusion network (Matusov et al., 2006) is built from the backbone while the remaining hypotheses are aligned against the backbone using edit-distance based alignment methods (cf. Section 5.3.2). The features used to score each arc in the confusion network (CN) are word posterior probability, target language model and length penalty. Minimum Error Rate Training (MERT) (Och, 2003) is applied to tune the CN weights. In our experiments, both APE hypotheses – PB-SAPE and HPB-SAPE – and the baseline Google Translate (GT) output are passed on to the system combination framework which produces the final system combination output (SC-APE).

5.8 Neural Network based APE

Our NNAPE system is based on a bidirectional (forward-backward) RNN based encoder-decoder[4] (Bahdanau et al., 2015). Our NNAPE model encodes a variable-length TL_{mt}

[3]We used Google Translate as our baseline MT for the experiments reported in this chapter.

[4]We used GroundHog (`https://github.com/lisa-groundhog/GroundHog`) to build our NNAPE system.

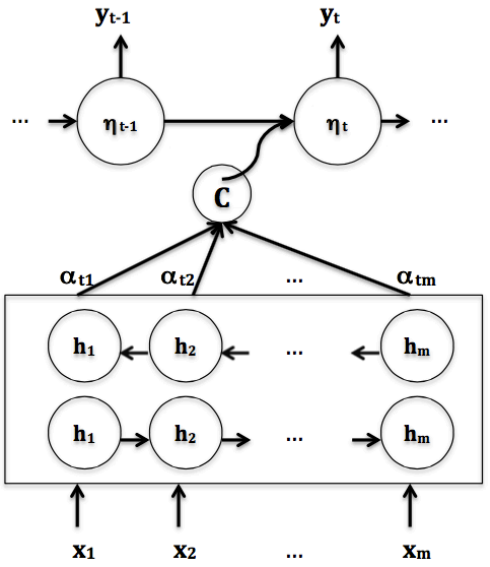


FIGURE 5.3: Generating the t^{th} TL_{pe} word y_t for a given TL_{mt} ($\mathbf{x}$) by our NNAPE System. We followed the same graphical architecture described in Bahdanau et al. (2015).

sequence (e.g. $\mathbf{x} = x_1, x_2, x_3...x_m$) into a fixed-length vector representation and then decodes a given fixed-length vector representation back into a variable-length TL_{pe} sequence (e.g. $\mathbf{y} = y_1, y_2, y_3...y_n$). Input and output sequence lengths, m and n, may differ. In our experiment, using an attention mechanism, we applied variable-length vector representations for the encoder, which is found beneficial as described in (Bahdanau et al., 2015).

A Bidirectional RNN encoder consists of forward and backward RNNs. The forward RNN encoder reads in each $\mathbf{x}$ sequentially from x_1 to x_m and at each time step t. The hidden state h_t of the RNN is updated by using a non-linear activation function f (Equation 5.17), an elementwise logistic sigmoid with an LSTM unit.

$$h_t = f(h_{t-1}, x_t) \tag{5.17}$$

Similarly, the backward RNN encoder reads the input sequence and calculates hidden states in reverse (i.e. x_m to x_1 and h_m to h_1 respectively). After reading the entire input sequence, the hidden state of the RNN is provided a summary c context vector ('**C**' in Figure 5.3) of the whole input sequence.

The decoder is another RNN trained to generate the output sequence by predicting the next word y_t given the hidden state η_t and the context vector c_t (c.f., Figure 5.3). The

hidden state of the decoder at time t is computed as given below.

$$P(y_t|y_1, ...y_{t-1}, \mathbf{x}) = f(\eta_t, y_{t-1}, c_t) \tag{5.18}$$

$$\eta_t = f(\eta_{t-1}, y_{t-1}, c_t) \tag{5.19}$$

The context vector c_t can be computed as

$$c_t = \sum_{i=1}^{m} \alpha_{ti} h_i \tag{5.20}$$

Here, α_{ti}, is the weight of each h_i and can be computed as

$$\alpha_{ti} = \frac{exp(e_{ti})}{\sum_{j=1}^{m} exp(e_{tj})} \tag{5.21}$$

where $e_{ti} = a(\eta_{t-1}, h_i)$ is an alignment model which provides a matching score between the inputs around position i and the output at position t. The alignment score is based on the i^{th} annotation h_i of the input sentence and the RNN hidden state η_{t-1}. The alignment model itself is a feedforward neural network which directly computes a soft alignment that allows the gradient of the cost function to be backpropagated through. The gradient is used to train the alignment model as well as the TL_{mt}–TL_{pe} translation model jointly.

The alignment model is computed $m \times n$ times as follows:

$$a(\eta_{t-1}, h_i) = v_a^T \tanh(W_a \eta_{t-1} + U_a h_i) \tag{5.22}$$

where $W_a \in \mathbf{R}^{n_h \times n_h}$, $U_a \in \mathbf{R}^{n_h \times 2n_h}$ and $v_a \in \mathbf{R}^{n_h}$ are the weight matrices of n_h hidden units.

In equation 5.21, the probability α_{ti} reflects the importance of the annotation h_i with respect to the previous hidden state η_{t-1} in deciding the next state η_t and generating y_t. This informs the decoder in deciding which parts of the source sentence to pay attention to. This implements a mechanism of attention in the decoder and relieves the encoder to encode all information in the source sentence into a fixed length vector by incorporating an attention mechanism in the decoder.

5.9 Experiments with English–Italian Data

We evaluated our APE models on an English–Italian APE task, which is detailed in the following subsections. We evaluated our NNAPE system and System Combination based APE system using two different experimental setups on the same data.

5.9.1 Data

The training data used for the experiments was developed in the MateCat[5] project and consists of 312K TL_{mt}–TL_{pe} parallel sentences. The parallel sentences are English to Italian MT output and their corresponding (human) post-edited Italian translations. Google Translate (GT) is the MT engine which provided the original Italian TL_{mt} output[6]. The data translated by GT and post-edited by human translators includes sentences from the Europarl corpus as well as news commentaries and are mixed with company client data. Since the data contains some non-Italian sentences, we applied automatic language identification (Shuyo, 2010) in order to select only Italian sentences. Automatic cleaning and pre-processing of the data was carried out by sorting the entire parallel training corpus based on sentence length, filtering the parallel data on maximum allowable sentence length of 80 tokens and sentence length ratio of 1:2 (either direction), removing duplicates and applying tokenization and punctuation normalization using Moses (Koehn et al., 2007) scripts. After cleaning the corpus we obtained a sentence-aligned TL_{mt}–TL_{pe} parallel corpus containing 213,795 sentence pairs. We randomly extracted 1,000 sentence pairs each for the development set and test set from the pre-processed parallel corpus and used the remaining 211,795 sentences as the training corpus for the APE engines. The training data features 57,568 and 61,582 unique words in TL_{mt} and TL_{pe}, respectively. We chose the 40,000 most frequent words from both TL_{mt} and TL_{pe} to train our NNAPE model. The remaining words, which are not among the most frequent words, are replaced by a special token ([UNK]). Our NNAPE model was trained for approximately 35 days, which is equivalent to 2,000,000 updates with GPU settings.

[5] `https://www.matecat.com/`

[6] Data for conducting the experiments and manual evaluation were shared by Translated SRL, Rome, Italy. The Italian translation were produced using Google Translate before the year 2013.

5.9.2 Experimental Settings for NNAPE

Our bidirectional RNN Encoder-Decoder contains 1,000 hidden units for the forward backward RNN encoder and 1,000 hidden units for the decoder. The network can be thought of as a multilateral neural network with a single maxout unit as a hidden layer (Goodfellow et al., 2013) to compute the conditional probability of each target word. The word embedding vector dimension is 620 and the size of the maxout hidden layer in the deep output is 500. The number of hidden units in the alignment model is 1,000. The model was trained on a mini-batched stochastic gradient descent (SGD) with 'Adadelta' (Zeiler, 2012). The main reason behind the use of 'Adadelta' was to automatically adapt the learning rate of each parameter ($\epsilon = 10^{-6}$ and $\rho = 0.95$). Each SGD update direction is computed using a mini-batch of 80 sentences.

We compare our NNAPE system with state-of-the-art phrase-based (Simard et al., 2007b) as well as hierarchical phrase-based APE (Pal, 2015) systems. We also compare the output provided by our system against the original Google Translate output. For building our phrase-based and hierarchical phrase-based APE systems, we set maximum phrase length to 7. A 5-gram language model built using KenLM (Heafield, 2011) was used for decoding. We used the Italian human post-edited data (i.e., TL_{pe} used for APE training) to build the 5-gram language model. System tuning was carried out using both k-best MIRA (Cherry and Foster, 2012) and Minimum Error Rate Training (MERT) (Och, 2003) on the held-out development set (devset) (see Section 4.9.3 in Chapter 4). After parameters were tuned, decoding was carried out on the held-out test set.

5.9.3 Evaluation of the NNAPE System

The performance of the NNAPE system was evaluated using both automatic and human evaluation as described below.

5.9.3.1 Automatic Evaluation

The output of the NNAPE system on the 1,000 test set sentences was evaluated using three MT evaluation metrics: BLEU (Papineni et al., 2002), TER (Snover et al., 2006a)

and METEOR (Denkowski and Lavie, 2011). Table 5.1 provides a comparison of the performance of our neural APE model against the baseline phrase-based APE (S_1), baseline hierarchical phrase-based APE (S_2) and the original GT output. We use a, b, c, and d in Table 5.1 for GT, S_1, S_2 and our NNAPE system (NN), respectively, to indicate statistical significance. For example, the S_2 BLEU score $63.87_{a,b}$ in Table 5.1 signifies that the improvements provided by S_2 in BLEU over Google Translate and phrase-based APE are statistically significant. Table 5.1 shows that S_1 provides statistically significant ($0.01 < p < 0.04$) improvements over GT across all metrics. Similarly S_2 yields statistically significant ($p < 0.01$) improvements over both GT and S_1 across all metrics. The NN system performs best and results in statistically significant ($p < 0.01$) improvements over all other systems across all metrics. A systematic trend ($NN > S_2 > S_1 > GT$) can be observed in Table 5.1 and the improvements are consistent across the different metrics. The relative performance gain achieved by NN over GT is the highest in TER.

System	**BLEU**	**TER**	**METEOR**
GT (a)	61.26	30.94	72.73
S_1 (b)	62.54_{a}	29.49_{a}	73.21_{a}
S_2 (c)	$63.87_{a,b}$	$28.67_{a,b}$	$73.63_{a,b}$
NN (d)	$\mathbf{65.22}_{a,b,c}$	$\mathbf{27.56}_{a,b,c}$	$\mathbf{74.59}_{a,b,c}$

TABLE 5.1: Automatic evaluation.

5.9.3.2 Human Evaluation

Human evaluation was carried out by four native speakers of Italian who all had between one and two years of professional translation experience. Since human evaluation is very costly and time consuming, it was carried out on a small portion of the test set consisting of 145 randomly sampled sentences where the NNAPE output differed from the GT output and the NNAPE system was only compared with the original GT output. We used a polling scheme with three different options. Translators were asked to choose the better of the two (GT or NN) outputs. They were also provided the 'uncertain' option in case of a tie. To avoid bias towards any particular system, the order in which the two system outputs were presented was randomized so that the translators did not know which system they were voting for.

We analyzed the outcome of the voting process (four translators each giving 145 votes) and found that the winning NN system received 285 (49.13%) votes compared to 99 (17.07%) votes received by the GT system, while the rest of the votes (196, 33.79%) went to the 'uncertain' option (cf. Figure 5.4). We measured pairwise inter-annotator agreement

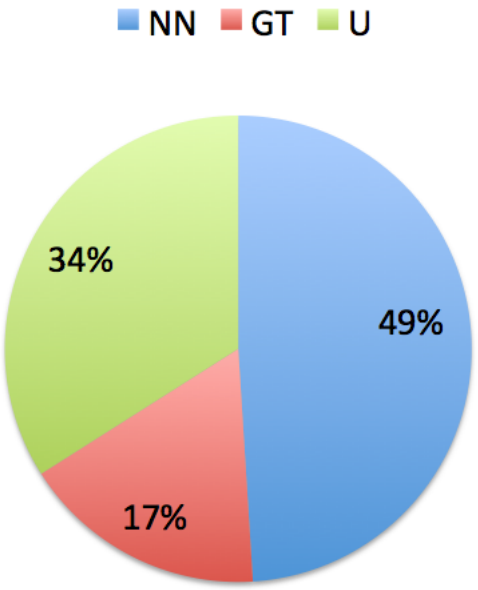


FIGURE 5.4: Polling outcome of NNAPE vs GT

between the translators by computing Cohen's κ coefficient (Cohen, 1960) reported in Table 5.2. The overall κ coefficient is 0.330. According to Landis and Koch (1977) this correlation coefficient can be interpreted as fair.

Cohen's κ	T1	T2	T3	T4
T1	-			
T2	0.141	-		
T3	0.424	0.232	-	
T4	0.398	0.540	0.248	-

TABLE 5.2: Pairwise correlation between translators in the evaluation process.

5.9.3.3 Analysis

The results of both automatic and human evaluation revealed that NNAPE provides additional performance gains over phrase-based and hierarchical SAPE approaches. On manual inspection we found that the NNAPE system drastically reduced the error of

wrong preposition insertion and deletion in Italian GT output and was also able to handle the improper use of prepositions and determiners (e.g. "states" → "dei stati", "the states" → "gli stati"). The use of a bidirectional RNN neural model makes the model sensitive towards contexts. Moreover, NNAPE captures global reordering by capturing contextual features which helps to reduce word ordering errors to some extent.

5.9.4 Experimental Settings for System Combination based APE

In our APE experiments we first integrated the hybrid word alignment model (cf. Section 5.3) into the SAPE engines modelled with PB-SMT (Koehn et al., 2003) and hierarchical PB-SMT (HPB-SMT) (Chiang, 2005). For building our statistical APE system, we used a maximum phrase length of 7 and a 5-gram language model trained using KenLM (Heafield, 2011). We use the Italian human post-edited data to build 5-gram language model. Model parameters were tuned using MERT (Och, 2003) on the held-out development set.

5.9.5 Evaluation for System Combination based APE

During evaluation we take into consideration the output produced by all three APE systems: PB-SAPE with hybrid word alignment, HPB-SAPE with hybrid word alignment and the system combination system (SC-APE) which also includes the output from the first stage system Google MT. We use a PB-SAPE system with GIZA++ alignment as a baseline APE system. The evaluation was carried out in two ways: automatic evaluation and human evaluation of the 1,000 test set sentences automatically post-edited by the SAPE systems. Out of the 1,000 test set sentences, the outputs of the system combination based final post-editing system (SC-APE) were different from the raw Google Translate translation output for 198 sentences, i.e., only 19.8% of the GT translations are post-edited by the SC-APE system, the remaining sentences are not affected by APE. The entire test set was evaluated with automatic evaluation metrics while only the 198 sentences were subjected to human evaluation.

5.9.5.1 Automatic Evaluation

We evaluated the APE systems using three well-known automatic MT evaluation metrics: BLEU, METEOR and TER. We also performed sentence level BLEU evaluation. Table 5.3 provides a comparison in terms of sentence level BLEU evaluation of the individual APE systems. Based on sentence level BLEU scores, the evaluation results presented in Table 5.3 show that 159 out of the 198 translations provided by the SC-APE are of better quality than the GT output. However, for the other 39 translations, the GT output is of better quality than the APE output. This may be partly due to the fact that the human post-edited reference translations are biased towards GT output. However, manual analysis revealed that some of these 39 GT translations are indeed better than the corresponding APE translations. Overall, PB-SAPE, HPB-SAPE and SC-SAPE provide gains over GT ($(APE-GT)/1000$) in terms of translation quality in 0.9%, 3.7% and 12% of the cases, respectively, as measured by S-BLEU.

Systems	**APE**	**GT**	**Tie**	**% Gain**	**% Loss**
PB-SAPE (HWA)	65	56	879	6.5%	5.6%
HPB-SAPE (HWA)	91	54	855	9.1%	5.4%
SC-APE	159	39	802	15.9%	3.9%

TABLE 5.3: Automatic evaluation using Sentence-BLEU over 1,000 test set sentences; $\%\ Gain = \frac{APE}{1000}$ & $\%\ Loss = \frac{GT}{1000}$

Table 5.4 provides a comparison between the baseline PB-SAPE based on GIZA++ word alignment, PB-SAPE and HPB-SAPE based on hybrid word alignment (HWA), SC-APE and GT. The comparison is carried out in terms of BLEU, METEOR and TER scores. A general trend can be observed across all metrics. The baseline PB-SAPE system fails to improve over GT, while HWA based PB-SAPE, HPB-SAPE and SC-APE improve the translation quality over GT according to all metrics. Among the three HWA based APE systems, SC-APE performs best followed by HPB-SAPE and PB-SAPE in all metrics. The SC-APE system provides 5.9%, 11% and 2.4% relative improvements over GT in BLEU, TER and METEOR, respectively, and all these improvements are statistically significant ($p < 0.01$) over all. The HPB-SAPE system also provides promising improvements (4.2%, 7.3% and 1.2% in BLEU, TER and METEOR, respectively) over GT while PB-SAPE system yields modest improvements.

Metric	PB-SAPE (Baseline)	PB-SAPE (HWA)	HPB-SAPE (HWA)	SC-APE	GT (First-Stage MT)
BLEU	59.90	62.70	63.87	**64.90**	61.26
TER	33.52	29.92	28.67	**27.52**	30.94
METEOR	69.54	73.31	73.63	**74.54**	72.73

TABLE 5.4: Automatic evaluation of the systems over 1,000 test set sentences.

5.9.5.2 Human Evaluation

The human evaluation process was carried out with four professional translators by introducing a polling system. The evaluation was carried out with the same poling system and same translators as described in Section 5.9.3.2. The polling system offered each voter three choices for every source English segment. Two of these options correspond to two different translation options. Translators act as voters and make a choice between the SC-APE output and the GT first-stage translation, based on whichever translation option they find better and more suitable for post-editing. Translators were also provided with a third option called *uncertain* (U), applicable whenever they are uncertain about which translation is better, i.e. when they deem both the GT and APE translations to be of equal quality (including equally unusable).

Table 5.5 shows the results of the polling scheme (human evaluation) of the raw GT output compared to the final automatic post-editing (SC-APE) output. The values in the table represent how many translations were chosen by each translator for individual systems. The polling based evaluation was carried out with 145 (of the 198) sentences. We discarded sentences containing fewer than six words (either in their source or their translation). Table 5.5 shows that translators preferred APE output over the raw MT output. Translators did not have any knowledge about which translations were from which system as the two translation options were presented to them in a random order. The winning APE system received on an average 49.3% votes compared to 17% votes received by the GT system, while 33.7% votes were neutral as the translators were undecided for those sentences.

The SC-APE system received a total of 280 votes and it received votes from at least one translator for 105 unique segments, while GT received 112 total votes for 61 unique segments and 188 votes were received for 94 unique segments for the *uncertain* category.

After detailed analysis we found that all four translators agreed on 27 APE translations, 6 GT translation and 9 neutral cases among the 145 sentences.

Translator	**Degree**	**Expertise**	**Experience in years**	**APE**	**GT**	**U**
T1	Translation	EN,FR → IT	1	91	22	32
T2	Linguistic and Cultural Studies	EN,FR, ES, CA → IT	2	57	17	71
T3	European Languages and Cultures	EN, FR, ES, DE → IT	1	72	37	36
T4	Business & Administration	EN → IT	1	65	23	58

TABLE 5.5: Outcome of polling with four expert translators for 145 sentences. (EN:English, DE:German, FR:French, ES:Spanish, CA:Catalan, IT:Italian).

For the 145 sentences, we measured pairwise inter-annotator agreements between the translators by computing Cohen's κ coefficient Cohen (1960). The overall κ coefficient was 0.331. According to Landis and Koch (1977) this correlation coefficient can be interpreted as fair.

5.9.5.3 Time and Productivity Gain Analysis

In order to investigate the effectiveness of the APE system in terms of time and productivity gains, a completely new test set was distributed among the four translators. The new test data consisted of real-life client segments consisting of 119 sentences with total 3,120 words in the same domain. SC-APE and GT translations were presented separately to the translators within their daily usage interface (MateCat).

Table 5.6 shows the statistics of how much time on average each individual translator took for the post-editing task. Table 5.6 also shows the average number of words (per minute, day) post-edited by each translator. We calculated productivity gain by comparing column 2 (SC-APE) with column 3 (GT) in Table 5.6. Table 5.6 shows that SC-APE improved the productivity of the translators in general. Among the four translators, SC-APE resulted in improved productivity for three translators (T1, T2 and T3), while for one translator

(T4) it seems to have resulted in a productivity loss. If we look at the seconds/word, words/minute, and words/hour measures on the GT data for the four translators, it is easily noticeable that T1 is the most efficient post-editor among them, followed by T2, T4 and T3. However, when the translators worked on the SAPE output, T2 was found to be the most productive while T4 was found to be the least productive. The productivity changes varied from 46.6% to -40%, which indicates that the utility of SAPE also varies from person to person. However, even taking into account the negative productivity of T4, average productivity increased 12.96% with SC-APE. One thing to be noted here is that the productivity loss of T4 should possibly not be considered for evaluation. We spoke to T4 after the evaluation and found that the translator was not solely concentrating on the post-editing job, instead switching amongst different jobs.

	SC-APE			**GT**				
	secs /word	**words /min**	**words /hour**	**secs /word**	**words /min**	**words /hour**	**Gain /hour**	**% Gain**
T1	2.81	21	1260	2.92	20	1200	60	5.0
T2	2.7	22	1320	3.88	15	900	420	46.6
T3	4.82	12	720	6.75	9	540	180	33.3
T4	9.80	6	360	5.84	10	600	-240	-40.0

TABLE 5.6: Post editing statistics over GT and SC-APE.

We also conducted a detailed evaluation of the post-editing carried out by the four translators. The results are reported in Table 5.7. Column 2 (fine grained evaluation score) in Table 5.7 shows the average of scores assigned to each translator by MateCat based on 5 criteria: tag issues (mismatches, white spaces), translation errors (mistranslation, additions/omissions), terminology and translation consistency, language quality (grammar, punctuation, spelling) and style (readability, consistent style and tone). MateCat also classifies each translator to one of the 4 performance levels[7] – excellent (3), acceptable (2), poor (1) and fail (0), for each of the above mentioned 5 criteria. Column 3 (weight based on quality) shows the sum of the scores indicating performance levels for the 5 criteria. By multiplying the values in column 2 and column 3, we arrive at the final assessment score assigned to each translator.

[7] `http://www.matecat.com/support/revising-projects/revising-translation-jobs/`

	Fine grained Evaluation score (s_f)	**Weight based on quality (w_q)**	**Final Assessment $fa = s_f \times w_q$**
T1	4.46	7	31.22
T2	4.44	6	26.64
T3	1.33	2	2.66
T4	2.74	1	2.74

TABLE 5.7: Assessment of the post-editors based on their performance and quality.

By weighting the percentage gain (cf. last column in Table 5.6) with the final assessment scores (cf. last column in Table 5.7), as in Equation 5.23, we obtain an average productivity increase of 21.76%. Even considering the negative productivity of T4, this overall productivity gain is significant.

$$average\ productivity\ gain = \frac{\sum_{i=1}^{4} gain_i \times f_{ai}}{\sum_{i=1}^{4} f_{ai}} \tag{5.23}$$

5.10 Experiment with English–German Data

The MT outputs provided by the WMT-2016 APE task (Turchi et al., 2016; Bojar et al., 2016) (c.f. Table 5.8) are considered as the first-stage MT system translation. The training data consist of 12K triplets of source (SL), MT output (TL_{mt}) and human post-edits (TL_{pe}). The description of the released data are detailed in Bojar et al. (2016). For building our SAPE system, we experimented with various maximum phrase lengths for the translation model and n-gram settings for the language model. We found that using a maximum phrase length of 10 for the translation model and a 6-gram language model produced the best results in terms of BLEU (Papineni et al., 2002) scores.

The experimental settings for building our APE system include word alignment between TL_{mt} and TL_{pe} trained on three different aligners: Berkeley Aligner (Liang et al., 2006), METEOR aligner (Lavie and Agarwal, 2007) and TER (Snover et al., 2006a). We used phrase-extraction (Koehn et al., 2003) and hierarchical phrase-extraction (Chiang, 2005) to build our PB-SAPE and hierarchical phrase-based statistical (HPB-SAPE) system respectively. The reordering model for PB-SAPE was trained with the hierarchical, monotone, swap, left-to-right bidirectional (hier-mslr-bidirectional) method (Galley and

Manning, 2008) and conditioned on both source and target language. The 5-gram target language model was trained using KenLM (Heafield, 2011) on TL_{pe} training data. We performed smoothing of the phrase table using the Good-Turing smoothing technique (Foster et al., 2006). System tuning was carried out using Minimum Error Rate Training (MERT) (Och, 2003) optimized with k-best MIRA (Cherry and Foster, 2012) on a held-out development set of 500 sentences randomly extracted from the training data. Therefore, all models were built on 11,500 parallel TL_{mt}–TL_{pe} sentences. After the parameters were tuned, decoding was carried out on the held-out development set ('Dev' in Table 5.8) as well as on the test set.

Table 5.8 presents the statistics of the training, development and test sets released for the English–German APE Task organized in WMT-2016. These data sets did not require any preprocessing in terms of encoding or alignment.

	SEN	Tokens		
		EN	DE-MT	DE-PE
Train	12,000	201,505	210,573	214,720
Dev	1,000	17,827	19,355	19,763
Test	2,000	31,477	34,332	35,276

Table 5.8: Statistics of the WMT-2016 APE Shared Task Data Set. SEN: Sentences, EN: English, DE: German.

We carried out various experiments with different settings using this dataset and the results obtained on the development set are reported in Table 5.9. In the set of experiments reported in Table 5.9, three word alignment models – one statistical based aligner i.e., the Berkeley aligner (Liang et al., 2006) and two edit distance based aligners i.e., the METEOR aligner (Lavie and Agarwal, 2007) and the TER aligner (Snover et al., 2006a), are integrated separately within both PB-SAPE and the HPB-SAPE systems which resulted in three different PB-SAPE (Experiment 2, 3 and 4 in Table 5.9) and HPB-SAPE (Experiment 5, 6 and 7 in Table 5.9) systems. These systems are compared against the raw MT output (*Baseline* in Table 5.9).

It is evident from Table 5.9 that in this task the METEOR aligner performs better than the other two aligners. Therefore, we used METEOR based alignment for our OSM based PB-SAPE model ('OSM' in Table 5.9). The experiment results show that compared to other systems in Table 5.9 the OSM based model performs better in terms of BLEU (Papineni

et al., 2002), TER and METEOR. Evaluation results also reveal that both PB-SAPE and HPB-SAPE system perform better than the baseline system on the development set data. The OSM system achieves 3.06% relative (1.99 absolute BLEU points) improvement over the baseline.

System		**Exp.**	**BLEU**	**MET**	**TER**
Baseline	WMT MT-PE	1	65.02	47.79	24.42
PB-SAPE	Berkeley Aligner	2	65.89	48.23	24.51
	METEOR Aligner	3	65.97	48.34	24.36
	TER Aligner	4	65.14	47.85	24.96
HPB-SAPE	Berkeley Aligner	5	66.09	48.31	24.56
	METEOR Aligner	6	66.55	48.58	24.51
	TER Aligner	7	65.19	47.91	24.97
OSM	METEOR Aligner	8	67.01	48.80	24.04

TABLE 5.9: Systematic Evaluation on the WMT-2016 APE Shared Task Development Set

Table 5.10 presents the evaluation results obtained on the test set. According to the test set evaluation, our system achieves similar improvements to those seen when using the development set data. Two baseline systems are reported in Table 5.10; *Baseline1* represents the raw MT output and *Baseline2* is based on Statistical APE (Simard et al., 2007b) (a phrase-based system (Koehn et al., 2007) built using Moses[8] with default settings). Two different systems – *OSM_Primary* and *OSM_Constrastive* were submitted to the WMT-2016 APE shared task. The difference between the two submissions is that the *OSM_Primary* system was tuned with all phrase-based setting parameters including OSM parameters while *OSM_Constrastive* was also tuned with similar parameters but excluding OSM parameters. The tuning process of the OSM parameters is conducted with MERT and optimized with MIRA. Our primary submission obtained a BLEU score of 64.10 (1.99 absolute points and 3.2% relative improvement in BLEU) and a TER score of 24.14 (0.66 absolute points and 0.25% relative improvement in TER) over *Baseline1*. Compared to the *Baseline2* system, our primary submission achieved 0.63 absolute points and 0.99% relative improvement in BLEU and 0.50 absolute points and 0.20% relative improvement in TER.

[8]http://www.statmt.org/moses/

System	BLEU	TER
Baseline1	62.11	24.76
Baseline2	63.47	24.64
OSM_Primary	**64.10**	**24.14**
OSM_Constrastive	64.00	**24.14**

TABLE 5.10: Evaluation on the WMT-2016 APE Shared Task Test Set

5.11 Conclusions and Future Work

We applied different approaches to APE on two different datasets (English–Italian and English–German). We tested the NNAPE and SC-APE systems on the English–Italian dataset. The OSM based APE system was evaluated on the English–German WMT 2016 APE dataset.

The NNAPE system provides statistically significant improvements over existing state-of-the-art APE models and produces significantly better translations than GT which is a very strong first-stage MT system and very difficult to beat. This enhancement in translation quality through APE should reduce human PE effort. Human evaluation revealed that the NNAPE generated PE translations contain fewer lexical errors and more importantly NNAPE rectifies erroneous word insertions and deletions, and improves word ordering. We evaluated our system in a real-life setting in a commercial environment to analyze time and productivity gain provided by the proposed automatic post-editing. We found an average productivity gain of 21.76% for English–Italian APE (Pal et al., 2016b). This addresses RQ5 ("To what extent is an APE system able to reduce final post-editing effort in terms of increasing productivity?") raised at the beginning of this chapter.

The use of a single statistical aligner in our PB-SMT based baseline APE fails to improve over raw Italian Google MT output; instead it degrades the performance, as was also reported by Béchara et al. (2011). This motivated us to use alignment combination models including both statistical and edit-distance based methods in our hybrid word alignment model for APE. By improving word alignment, the APE system automatically acquires better lexical associations and the "hybrid" PB-SAPE system shows improvements over the Google MT baseline. The reason for using a hierarchical phrase extraction model for APE is that it makes the model more sensitive to syntactic structures. HPB-SAPE

captures global reordering by SCFG, helping to correct word order errors to some extent. Integration of our hybrid word alignment into the APE model resulted in both PB-SAPE (S_1) and HPB-SAPE (S_2) producing better translations than GT. SC-APE of S_1, S_2 and GT provided further improvements over raw MT output. We performed a statistical significance test on GT, S_1, S_2 and SC-APE, which showed that S_1 provides statistically significant ($0.01 < p < 0.04$) improvements over GT across all metrics. Similarly S_2 yields statistically significant ($p < 0.01$) improvements over both GT and S_1 across all metrics. Our SC-APE system performs best and results in statistically significant ($p < 0.01$) improvements over all other systems across all metrics.

This chapter also presented the OSM based APE system submitted in the English–German APE task at WMT-2016. The system demonstrates the crucial role METEOR-based alignment and OSM based SAPE can play in SAPE tasks. The use of statistical aligners in the PB-SMT/HPB-SMT pipeline improve the APE system. However, performances with respect to the translations provided by the baseline are not promising. This is the reason behind using edit distance-based word alignment into the pipeline. The reason for using the OSM model is that the model tightly couples translation and reordering. Apart from that, the OSM model also considers all possible re-orderings instead of performing search only on a limited number of pre-calculated orderings. The proposed system, an OSM-based SAPE approach, was successful in improving over the PB-SAPE as well as HPB-SAPE performance.

We successfully showed that both our APE experiments (English–Italian and English–German) outperform the first-stage MT systems. Human evaluation also revealed that the translation quality of the APE system is much better than the first-stage MT system. However, some of translations produced by the first-stage MT system are still better than APE, however, they are much less in number. Therefore, in terms of overall translation performance, APE is an effective solution. Furthermore, APE acts as a 2^{nd} stage MT system, therefore, it does not implicate reconfiguration or modification of the first-stage MT system (Pal et al., 2015c, 2016b,c,f). This addresses RQ4, i.e., "How can we build an effective automatic post-editing system which can improve the translation quality of the first-stage MT system?".

The WMT-2016 APE shared task was a great opportunity to test APE methods that can later be applied in real-word post-editing and CAT tools. We are currently working

on implementing the APE methods described in this chapter in CATaLog, a recently-developed CAT tool that provides translators with suggestions originating from MT, translation memory (TM) and APE (Nayek et al., 2015; Pal et al., 2016e,d). In so doing, we aim to provide better suggestions for post-editing and we would like to investigate how this impacts human post-editing performance by carrying out user studies.

In future, we want to extend the APE system by incorporating source language knowledge into the network and to compare LSTM with GRU hidden units.

Chapter 6

Interactive Translation Workflow

This chapter presents *CATaLog Online*, a new web-based computer-aided translation (CAT) tool developed towards improving existing CAT workflows. *CATaLog Online* is freeware software that can be used through a web browser and requires only a simple registration. The tool features a number of editing and logging functions enhanced with several novel features. The tool has been developed with the goal of improving post-editing productivity and experience. *CATaLog Online* employs a novel color coding scheme that highlights matching and non-matching fragments in each suggested translation memory (TM) segment and indicates which parts of the TM segments provide more reliable translations. Instead of the traditional fuzzy matching used in TM-based CAT tools, *CATaLog Online* uses an edit distance style similarity metric and Lucene retrieval scores to identify and re-rank the relevant TM suggestions. *CATaLog Online* provides a post-editing environment with simple yet helpful project management functions. It provides translation suggestions from TM, MT and automatic post-editing (APE), and furthermore it records detailed logs of post-editing activities that are not available in most commercial CAT tools. To test the new approaches presented in this chapter, we carried out a user study on an English–German translation task using the tool. User feedback revealed that the users preferred using *CATaLog Online* over existing CAT tools in some respects, especially by selecting the output of the APE system and taking advantage of the color coding scheme for TM suggestions. In this chapter, we also introduce another important function of TM: proposing a new translation suggestion from the top TM suggestions in both the desktop version (*CATaLog*) and web version of *CATaLog Online*. Traditionally,

TMs do not generate any new translations; therefore, this feature presents a step beyond traditional TMs. Besides, this improves human-computer interaction (HCI) issues with TMs since this new functionality generates a new translation based on the translation template chosen by the user. This chapter addresses **RQ6:** *How can human interaction with CAT tools be optimized in existing MT workflows?*

Core parts of the research presented in this chapter have been published in (Nayek et al., 2015; Nayak et al., 2016; Pal et al., 2016e,d).

Figure 6.1 schematically shows the research presented and the research questions addressed in this Chapter.

6.1 Introduction

The use of computer software is an important part of modern translation workflows (Lagoudaki, 2008; Zaretskaya et al., 2015c). A number of tools are widely used by professional translators, most notably CAT tools and terminology management software. These tools increase translators' productivity, improve consistency in translation and, in turn, reduce cost of translation.

The most important component in state-of-the-art CAT tools is the TM. TMs are databases which store translated segments (such as words, phrases, sentences or even paragraphs) that can be used in future translations. For every new input text, the TM engine checks whether there are segments in the memory which are similar to those of the input text. The TM engine displays the most similar segments together with their translations to the translator. The translators can either accept, reject or modify the suggestions received from the TM engine. As the process is done iteratively, every new translation increases the size and improves the quality of the TM making it more useful for future translations. TMs are particularly useful in domains or text types in which translations tend to be repetitive, most notably in the case of technical or specialized translations.

As discussed in Pal et al. (2016e), the idea behind TMs is relatively simple. However, the process of matching and retrieval of source and target segments is not trivial. Several strategies have been applied to improve TM retrieval engines such as incorporating semantic knowledge and paraphrasing (Utiyama et al., 2011; Gupta and Orăsan, 2014;

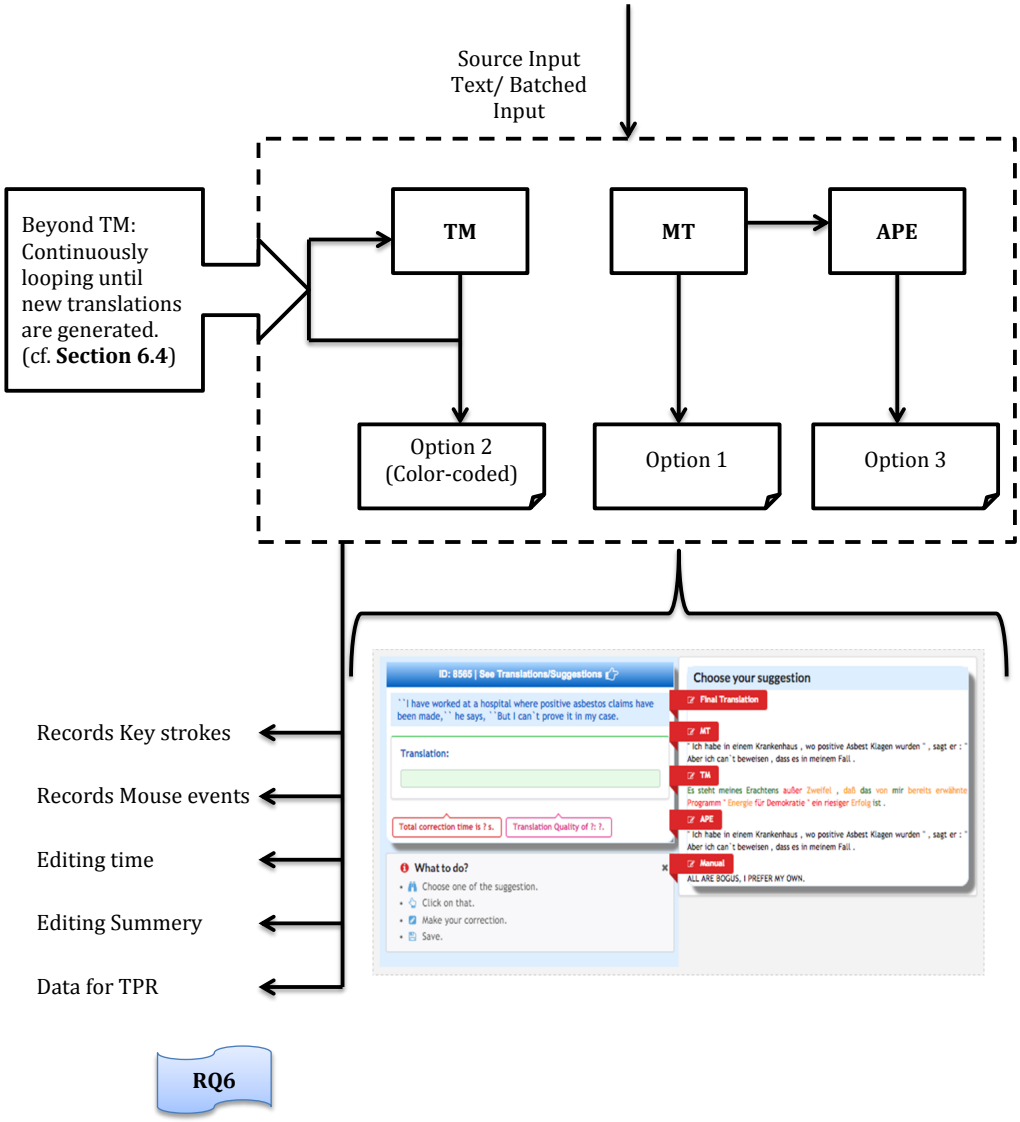


FIGURE 6.1: Schematic design of the research and the research questions presented in this Chapter.

Gupta et al., 2015b) and syntax (Clark, 2002; Gotti et al., 2005; Nayak et al., 2016). A recent trend is the integration of TM and machine translation (MT) output in a single environment (He et al., 2010; Kanavos and Kartsaklis, 2010; Koehn and Senellart, 2010). MT systems have been improving substantially over the past decade, particularly for domain specific translations. Taking advantage of these improvements, a number of CAT tools have been integrating MT outputs along with TM matches. One such tool is MateCat (Cettolo et al., 2013).

In this chapter we discuss new approaches to improve TM performance and CAT tool interfaces. With our contribution we aim to make TM suggestions more useful and accurate as well as to improve CAT tools' usability by providing translators with more intuitive software interfaces. For this purpose, we developed a new CAT tool called *CATaLog* (Nayek et al., 2015) and its web-based counterpart, *CATaLog Online* (Pal et al., 2016e)[1] and carried out user studies to evaluate the impact of the proposed innovations in real-world translation workflows.

Traditionally, TM tools do not generate translations; instead they present the user with matching sentence pairs that are very similar to the sentence being translated. Post-editors, when working with TM tools, seldom find an exact match. Therefore, almost always, the TM suggestions do contain at least a few unmatched fragments of the input sentence. However, it can often be observed that the translations for those unmatched fragments are available in other suggestions or may be in some other sentences in the TM database. Extracting the translations of those unmatched fragments and inserting them into the suggested TM translations can result in a complete translation for a particular input sentence. Although this may lead to loss of fluency in the suggested new translation, it often improves the adequacy of the suggested translation. Thus, it reduces the post-editing effort significantly since the user does not have to translate all of unmatched fragments in the suggested TM translations from scratch. Some of the research presented in this chapter has been published in (Pal et al., 2016e,d; Nayak et al., 2016).

[1] `http://ttg.uni-saarland.de/software/catalog/`

6.2 Related Work

In the translation and localization industries, translators are more frequently acting as post-editors, working with pre-translated texts from TMs or MT output. This has resulted in CAT tools becoming an integral part of a translator's workflow. A number of studies were carried out on the translation process to investigate translators' productivity, cognitive load, effort, time, quality, etc. in CAT environments (O'Brien, 2006; Guerberof, 2009; Plitt and Masselot, 2010; Federico et al., 2012; Guerberof, 2012; Zampieri and Vela, 2014; Vieira, 2014; Koponen, 2016). These studies indicate that the use of TMs and MT output decreases the effort required for translation and improves productivity.

As recent state-of-the-art CAT tools have been integrating TM and MT in a single post-editing environment, research has been carried out to improve translation recommendation systems by predicting whether TM or MT output is more likely to serve as a good translation for a given segment (He et al., 2010; Kanavos and Kartsaklis, 2010). This can be modelled by a binary classifier using text classification algorithms (e.g., Support Vector Machines).

Simard and Isabelle (2009) reported on the integration of phrase-based statistical machine translation (PB-SMT) with TM in a CAT environment. A similar approach was proposed by Zhechev and van Genabith (2010). Koehn and Senellart (2010) proposed an MT–TM integration approach in which TMs are used to retrieve matching segments and an SMT system is used to fill in the gaps by translating parts of the segment which were not retrieved from the TM.

CAT tools make translators' jobs easier. They improve translation performance and significantly increase translation quality through special quality checking tools. There are many CAT tools available in the market from complex desktop solutions, e.g., SDL Trados[2], MemoQ[3], Wordfast[4], to relatively simple but powerful cloud tools, e.g., SmartCAT[5], MateCat[6], Memsource[7], etc. OmegaT[8], an excellent free and open source translation support tool capable of working with translation formats from the leading tools, is a popular

[2]http://www.sdl.com/solution/language/translation-productivity/trados-studio/
[3]https://www.memoq.com/en/
[4]https://www.wordfast.net/
[5]https://www.smartcat.ai/
[6]https://www.matecat.com/
[7]https://www.memsource.com/
[8]http://www.omegat.org/en/omegat.html

alternative to commercial tools. In this chapter, we present our new free web-based CAT tool called *CATaLog Online* which provides a novel and user-friendly online CAT environment for post-editors/translators. Our goal is to support distributed translation, reduce post-editing time and effort, improve the post-editing experience and capture data for incremental MT/APE (automatic post-editing) and translation process research.

6.3 *CATaLog Online*: System Description

This section describes *CATaLog Online*, a novel and user-friendly web-based CAT tool. We discuss its main functionalities and its novel features that distinguish it from other CAT tools.

CATaLog Online offers translations from three engines – TM (Nayek et al., 2015), MT (Pal et al., 2015a) and APE (Pal et al., 2015c), from which users can choose the most suitable translation and, if required, post-edit. Users can upload their own translation memories on the platform or can make use of the *CATaLog Online* back-end translation memory. In addition to using the *CATaLog Online* MT (Pal et al., 2015a) and APE (Pal et al., 2015c) engines, users can also upload translations produced by third-party MT systems.

Upon presenting the tool with a new input (source language) segment[9], the tool retrieves the most similar segments contained in the TM database ranked according to their similarity to the input segment using an edit distance style ranking algorithm presented in Section 6.3.1. An important aspect of computing similarity is finding an alignment between the input and the retrieved segments. *CATaLog Online* aligns the input source language (SL) segment against the SL sentences in the TM database. It also establishes word alignments between TM SL sentences and their corresponding translations. From these two sets of alignments the tool identifies which parts of the TM translation suggestions are relevant with respect to the input sentence and which are not, i.e., which parts of the TM translation suggestions should remain intact after post-editing and which portions need editing. *CATaLog* presents the TM matches using a novel color scheme for better visualization; matched parts are displayed in green and unmatched parts are displayed

[9]'Segment' is a widely accepted term for 'sentence' in the translation industry. In this chapter we also use 'segment' and 'sentence' interchangeably.

in red. The colors help the translators to visualize instantaneously how similar the five suggested segments are to the input segment and how much post-editing effort each TM translation suggestion requires. The color scheme is detailed in Section 6.3.2. Comparing every TM source segments against the input sentence is typically a slow process, particularly if the TM database is very large. To speed up the TM search process, *CATaLog Online* uses the Nutch[10] information retrieval (IR) system which is discussed in Section 6.3.3. In *CATaLog Online*, users can choose between MT output (cf. Section 6.3.4), automatic post-editing (APE) (cf. Section 6.3.5) and TM segments. Instead of using the integrated MT, APE and TM, users can upload translations produced by third-party MT systems and their own private TMs. *CATaLog Online* provides facilities to translate single sentences and can also be operated in batch mode i.e., by uploading a file. Currently our tool offers translation outputs from MT and APE engines for some language pairs. The back-end MT and APE systems are discussed in Sections 6.3.4 and 6.3.5, respectively.

6.3.1 Finding Similar Segments

For finding TM segments that are similar to the input segment, we use alignments provided by Translation Error Rate (TER) (Snover et al., 2009), an automatic MT evaluation metric. The alignments provided by TER also enable us to find similar and dissimilar parts of an input segment and a matching TM source segment. TER is an edit distance style error metric and it provides an edit ratio (often referred to as edit rate or error rate) in terms of how much editing is required to convert a translation hypothesis into a reference translation with respect to the length of the translation hypothesis. TER allows four types of editing operations – *insert*, *delete*, *substitute* and *shift*. *CATaLog Online* establishes the alignment between an input sentence and a matching TM source segment by employing the TER metric (using tercom-7.251[11]). Simard and Fujita (2012) first proposed the use of MT evaluation metrics as similarity functions in TM. They experimented with several MT evaluation metrics, viz. BLEU, NIST, Meteor and TER, and studied their behaviors on TM performance. We use TER as the similarity metric (in an inverse way, since a lower TER value indicates higher similarity) in *CATaLog Online* as it is very fast and lightweight and it directly mimics the human post-editing effort. Among the MT evaluation metrics, TER is known to provide high correlation with human judgements (Snover et al., 2006b).

[10] `http://nutch.apache.org/`
[11] `http://www.cs.umd.edu/~snover/tercom/`

Input: we would like a table by the window .

TM Match: we want to have a table near the window .

TER alignment:

"we","we",C,0

"want","",D,0

"to","would",S,0

"have","like",S,0

"a","a",C,0

"table","table",C,0

"near","by",S,0

"the","the",C,0

"window","window",C,0

".",".",C,0

we want to have a table near the window .
| D S S | | S | | |
we - would like a table by the window .

FIGURE 6.2: TER alignment between input sentence and TM matched segment.

Moreover, the tercom-7.251 package also produces the alignments between a sentence pair from which it is very easy to identify which portions in a TM segment are relevant to the input sentence and which portions need to be worked on.

The TER alignment between an input sentence and a TM segment is illustrated in the example given below (cf. Figure 6.2). Here, *C* represents a match (shown as the vertical bar '|'), and *I*, *D* and *S* represent the three post-editing operations – insertion, deletion and substitution, respectively.

The example given above involves only one deletion and three substitution operations. It does not involve any insertion or shift operations. A shift operation is indicated in the TER alignment by a *C* followed by a non-zero integer (as opposed to "C,0" which represents a match). The non-zero integer value indicates the shifting offset; positive and negative values represent shifting to the right and left, respectively.

Since we want to rank the relevant TM segments based on their similarity to the input sentence, we could directly use the TER score in an inverse way. TER is an error metric and therefore the TER score is proportional to how dissimilar two sentences are; i.e., the lower the TER score, the higher the similarity. However, in *CATaLog Online* we use our own similarity scoring mechanism based on the alignments provided by TER. TER gives equal weight to each edit operation, i.e., deletion, insertion, substitution and shift. However, in actual human post-editing, deletion takes substantially less time and effort than the other editing operations. Different edit costs for different edit operations should yield better results. These edit costs or weights can be adjusted or tuned to obtain better output from the TM. Ideally, these edit costs should be representative of the time and effort required for the corresponding edit operations.

In our system, we assigned a very low weight to the deletion operation and equal weights to the other three edit operations. To illustrate why different editing costs matter, let us consider the example below.

- Input segment: how much does it cost ?
- TM segment 1: how much does it cost to the holiday inn by taxi ?
- TM segment 2: how much ?

If each edit operation were assigned an equal weight, according to the TER score, TM segment 2 would be a better match with respect to the input segment than TM segment 1, as TM segment 2 involves inserting translations for three non-matching words of the input segment ("does it cost"), as opposed to deleting translations of the six non-matching words ("to the holiday inn by taxi") in TM segment 1. However, deleting the translations for the six non-matching words from the translation of TM segment 1, which are already highlighted red (cf. Section 6.3.2) by our tool, takes much less cognitive effort and time than inserting translations for the three non-matching words of the input segment into the translation of TM segment 1. Therefore, TM segment 1 is intuitively a much better choice than TM segment 2 with respect to the above mentioned input segment. This justifies assigning minimal weights to the deletion operation which results in the system giving preference to TM segment 1 over TM segment 2 for this input segment.

The Needleman-Wunsch (Needleman and Wunsch, 1970) algorithm is another edit-distance based algorithm which is widely used in bioinformatics to align or find the similarity between two protein or nucleotide sequences. The outcome of both the TER and Needleman-Wunsch algorithms is an optimal global alignment between two strings. The basic distinction between the TER and Needleman-Wunsch algorithms is that TER (or edit distance in general) is an error metric which tries to minimize the distance (or dissimilarity) between two strings while Needleman-Wunsch is a similarity metric that tries to maximize the similarity between two strings. Both algorithms penalize mismatches; TER assigns a positive *cost* while Needleman-Wunsch algorithm assigns a negative *similarity*. However, unlike TER, the Needleman-Wunsch algorithm rewards matches. The similarity metric that we used for finding similar TM segments is similar to and motivated by the Needleman-Wunsch algorithm. However, TER has the advantage that it also considers shift operations which the Needleman-Wunsch algorithm does not. Shifting is a very meaningful edit operation in the human post-editing scenario. Therefore, we used elements of both the TER and Needleman-Wunsch algorithms to design the similarity metric of *CATaLog Online*. We take the alignment computed by TER but calculate the similarity score using the intuition of the Needleman-Wunsch algorithm by penalizing edit operations and rewarding matches.

The top 100 most relevant TM suggestions returned by the Lucene based search engine, Nutch, (cf. Section 6.3.3) are re-ranked using Equation 6.1, where n_m and s_m refer to the number of matches and match reward scores, respectively; e_i refers to four types of edit operations – *insert*, *delete*, *substitute* and *shift*; n_{e_i} and c_{e_i} refer to number of e_i edit operations required and the corresponding edit cost, respectively. Thus we reward matches and penalize edits to arrive at the final similarity score.

$$S = n_m \times s_m - \sum_{i=1}^{4} n_{e_i} \times c_{e_i} \tag{6.1}$$

By way of example let us consider match reward=0.80, deletion cost=0.10, shift cost=0.20, insertion cost=0.50 and substitution cost=0.70. Let us also consider that the TER alignment between an input segment and a relevant TM source segment is "CCDCCCSCI" where the 3$^{\text{rd}}$ 'C' refers to a shifting operation (say, "C,2"). We rewrite this alignment

as "MMDHMMSMI" where 'M' and 'H' refer to matches and shifts, respectively. Then, according to Equation 6.1 the corresponding TM similarity score is calculated as follows.

$$S = 0.80 \times 5 - 0.10 \times 1 - 0.20 \times 1 - 0.50 \times 1 - 0.70 \times 1 = 2.5$$

The desktop version of the tool, *CATaLog*, lets the user set these match rewards and edit costs. The TM similarity scores (S) are finally normalized (S_n) with respect to the maximum length (in terms of tokens) between input text and the retrieved TM segment. The value of the normalized TM similarity score lies between $-1 \leq S_n \leq 1$.

6.3.2 Color Coding

Like most of the existing TM based CAT tools, *CATaLog*, the back-end TM engine in *CATaLog Online*, presents the user with five most relevant translation suggestions from the TM database, while *CATaLog Online* presents only the top ranking TM suggestion from *CATaLog* along with the translations from the MT and APE engines.

In *CATaLog*, among the top five TM suggestions presented by the tool, the post-editor selects the most suitable TM reference translation to do the post-editing task. To make that decision process easy, *CATaLog* color codes the matched and unmatched parts in both the source and target of the TM suggestions. Green portions indicate that they are matched fragments and red portions indicate mismatches.

Matched and unmatched fragments in the source of the TM suggestions are easily identified through the TER alignments. To identify the corresponding matched and unmatched fragments in the target side of the TM suggestions the tool establishes word alignments between the TM source sentences and their corresponding translations using GIZA++ (Och and Ney, 2003b). However, any other word aligner, e.g., Berkeley Aligner (Liang et al., 2006), could be used to produce this alignment. The TER alignment between the input sentence and the relevant TM source segments, together with the alignment between the source and target of the relevant TM suggestions, are used to generate the color coding of the TM suggestions. The GIZA++ alignment file is directly integrated into the TM tool. The example given below shows an example TM sentence pair along with the corresponding word alignment produced by GIZA++.

- English: we want to have a table near the window .
- Bengali: আমরা জানালার কাছে একটা টেবিল চাই ।
  1 2 3 4 5 6 7
- Alignment: NULL ({}) we ({ 1 }) want ({ 6 }) to ({ }) have ({ }) a ({ 4 }) table ({ 5 }) near ({ 3 }) the ({ }) window ({ 2 }) . ({ 7 })

The word alignment between the TM source sentences and their corresponding translations is computed offline using GIZA++, only once, on the TM database for a specific language pair. TER provides the alignments between an input sentence and the corresponding top five TM source suggestions. Using these two sets of alignments we color the matched fragments of the TM suggestions in green and the unmatched fragments in red. Trados, a popular CAT tool, does not provide color coding at word level. By contrast, *CATaLog* highlights parts of the segment at word level whereas Trados highlights the entire segment according to the match percentage.

Color-coding the TM source segments makes explicit which portions of the matching TM source sentences match with the input sentence and which ones do not. Similarly, color-coding the TM target segments serves two purposes. Firstly, it makes the decision process easier for the translators as to which TM suggestion to choose and work on. Secondly, it guides the translators as to which fragments to post-edit in the chosen TM translation. The reason behind color-coding both the TM source and target segments is that a longer (matched or unmatched) source fragment might correspond to a shorter target fragment, or vice versa, due to language divergence. A reference translation which has more green fragments than red fragments will be a good candidate for post-editing. However, shorter TM translations with high green coverage may not be ideal candidates for post-editing, since post-editors might have to insert translations for many unmatched words in the input sentence.

In this context, it is to be noted that insertion and substitution operations are the most costly operations in post-editing. However, sentences involving insertions and substitutions are not preferred by the TM as it assigns a higher cost for insertion than deletion, and hence sentences involving many insertions are typically not shown as the top candidates by our TM.

The color coding scheme is illustrated with the following example in an English–Bengali translation task. The corresponding TM database consists of English sentences taken from the BTEC[12] (Basic Travel Expression Corpus) corpus and their Bengali translations[13]. For the convenience of non-native speakers, Latin transliteration glosses are provided within parenthesis for the Bengali sentences.

Input: you gave me wrong number .

Source Matches:

1. you gave me the wrong change . i paid eighty dollars .
2. i think you 've got the wrong number .
3. you are wrong .
4. you pay me .
5. you 're overcharging me .

Target Matches:

1. আপনি আমাকে ভুল খুচরো দিয়েছেন . আমি আশি ডলার দিয়েছি . (*Gloss: apni amake vul khuchro diyechen . ami ashi dollar diyechi .*) (*English Gloss: you me wrong change gave . I eighty dollar paid .*)
2. আমার ধারণা আপনি ভুল নম্বরে ফোন করেছেন . (*Gloss: amar dharona apni vul nombore phon korechen .*) (*English Gloss: I think you wrong number 've got .*)
3. আপনি ভুল . (*Gloss: apni vul .*) (*English Gloss: you wrong .*)
4. আপনি আমাকে টাকা দিন . (*Gloss: apni amake taka din .*) (*English Gloss: you me pay .*)
5. আপনি আমার কাছে থেকে বেশি নিচ্ছেন . (*Gloss: apni amar kache theke beshi nichchen .*) (*English Gloss: you me are overcharging .*)

[12] The BTEC corpus contains tourism-related sentences similar to those that are usually found in phrase books for tourists going abroad.

[13] This represents a work in progress.

For the input sentence shown above, the TM system shows the above mentioned color-coded top five TM matches in order of their relevance with respect to the post-editing effort (as deemed by the TM similarity metric) for producing the translation for the input sentence.

It is to be noted that when the post-editor selects a TM segment for post-editing, the input sentence is also color coded accordingly to reflect the corresponding matching and unmatched fragments in the input sentence. This also gives the post-editor an indication of how much post-editing is involved for the chosen TM segment. Red fragments in the input sentence correspond to insertion while red fragments in the TM segments correspond to deletion. Recalling the above example, if the translator chooses the translation of TM segment 1, "you gave me the wrong change . i paid eighty dollars .", the corresponding input source sentence will automatically be color coded as " you gave me wrong number .".

6.3.3 Improving Search Efficiency

Comparing every input sentence against all the TM source segments makes the search process very slow, particularly for large TMs. To improve search efficiency, *CATaLog Online* uses the Nutch[14] information retrieval (IR) system. Nutch follows the standard IR model of Lucene[15] with document parsing, document indexing, TF-IDF (term frequency-inverse document frequency) calculation, query parsing and finally searching/document retrieval and document ranking. In our implementation, each document contains (a) a TM source segment, (b) its corresponding translation and (c) the word alignments.

To generate the search query corresponding to an input segment, all the stop words are removed first from the input segment. After presenting an input segment as a query as a bag-of-words, Nutch retrieves the most relevant documents (i.e., a, b and c) with respect to the query. The set of relevant candidates are ranked by Nutch according to their similarity scores with respect to the query and the retrieved documents are collected and stored in a file. The ranking process also deals with dissimilarity measurement that provides a final fine-grained score to re-rank the retrieved matching segments.

Dissimilarity Measurement:

[14] `http://nutch.apache.org/`
[15] `http://lucene.apache.org/`

Algorithm 5: Dissimilarity-Measure(s_1, s_2)

Data: input s_1 and s_2

Result: Return *score*

begin

$score \leftarrow 0$;

for *all n-grams n contained in* s_1 *or* s_2 **do**

$$f_1 \leftarrow \begin{cases} frequency(n), & \text{if } n \in S_1 \\ 0, & \text{if } n \notin S_1 \end{cases}$$

$$f_2 \leftarrow \begin{cases} frequency(n), & \text{if } n \in S_2 \\ 0, & \text{if } n \notin S_2 \end{cases}$$

$score \leftarrow score + \{\frac{2(f_1 - f_2)}{(f_1 + f_2)}\}^2$

Algorithm 5 is based on (Kešelj et al., 2003) and it provides a measure of dissimilarity between an input segment (s_1) and the corresponding retrieved candidate segments (s_2). For identical segments that have identical n-grams, the dissimilarity score is 0. The algorithm returns a positive dissimilarity score after being presented with s_1 and s_2.

6.3.3.1 Re-ranking

The dissimilarity score returned by Algorithm 5 is subtracted from the similarity score assigned by Nutch and a final score is assigned to every candidate segment for a given input segment. All retrieved candidates are re-ranked with respect to this score. Only the top 100 ranked candidates are retained and passed on to the TM similarity matching procedure (cf. Section 6.3.1) which re-ranks these 100 TM candidates.

6.3.4 Machine Translation

The MT engine behind *CATaLog Online* is a hybrid system (Pal et al., 2015a) trained on additional knowledge such as extracted bilingual named entities and bilingual phrase pairs induced from example-based methods. The final hybrid system is a confusion network based system combination that combines output from multiple systems.

6.3.5 Automatic Post-Editing

The back-end APE system is based on the Operation Sequence Model (OSM) (Durrani et al., 2011, 2015) combined with phrased-based statistical MT (PB-SMT) (Koehn et al., 2003). The system is trained on monolingual data between MT outputs (TL_{mt}) produced by an MT system and their corresponding human post-edited version (TL_{pe}). The system takes as input the output produced by the MT System (described in Section 6.3.4) and provides an automatically post-edited translation. The APE system combines two models: monolingual phrase-based APE (cf. Chapter 5) and OSM with an edit distance style word alignment (METEOR (Lavie and Agarwal, 2007)) between English–German TL_{mt} and TL_{pe} WMT 2016 data (Turchi et al., 2016). Our APE system focuses on systematic errors produced by the first-stage MT system: incorrect lexical choices, incorrect word ordering, and incorrect insertion or deletion of a word. Since, in the OSM model, the translation and reordering operations are coupled into a single generative story, the reordering decisions may depend on the preceding translation decisions which in turn may depend on the preceding reordering decisions. The model provides a natural reordering mechanism and deals with both local and long-distance re-orderings consistently. Additionally, the PB-SMT model successfully reduces lexical errors. The integrated APE system is based on OSM based coupled with phrase based APE described in Pal et al. (2016f).

6.3.6 Translation Process Research with *CATaLog Online*

For a given input segment, the translator chooses the best translation suggestion among the options provided by *CATaLog Online*. The chosen translation suggestion may contain errors like missing words, incorrect word order, wrong lexical choice, presence of irrelevant words, untranslated words, punctuation errors, etc. The system records all the user activities during post-editing such as cursor positions, key strokes, text selection, mouse clicks, etc. *CATaLog Online* provides analytical summaries of post-editing activities on completion of a translation job. It also generates well structured XML formatted logs (cf. Figure 6.3) which is beneficial for translation process research (TPR).

In terms of TPR perspectives, we implemented functions in *CATaLog Online* to record word alignments between source–MT, MT–PE and source–PE. These alignments and post-editing information are beneficial for incremental MT/APE.

<Source>India and Japan prime ministers meet in Tokyo</Source>
<Human_Translation>Treffen des indischen und japanischen Premierministers in Tokio </Human_Translation>
<Engine_Translation>
<MT>Indien und Japan Premierminister in Tokio </MT>
<APE>Indien und Japan Premierminister in Tokio </APE>
<TM>So/G trafen/R vor/G kurzem/R die/G Premierminister/G von/R Indien/G und/G Pakistan/R in/G Pakistan/R zusammen/G ,/G um/R über/R diese/R Frage/R zu/R diskutieren/R ./Y </TM>
</Engine_Translation>
<Log>
<EDIT_Logs>
No. of Corrections="5" | No. of deletion="0" | No. of insertion="2" | No. of substitution="3" | No. of shift="0" | TER="62.5" ||| @Inserted: "treffen", "des"@Substituted: "indien" by "indischen", "japan" by "japanischen", "premierminister" by "premierministers"
</EDIT_Logs>
<EDIT_TIME>2.939</EDIT_TIME>
<Key_Strokes>Indien und Japan Premierminister in Tokio _IS@7@7 ||| Indien und Japan Premierminister in Tokio _IS@6@6 ||| Indien und Japan Premierminister in Tokio _IS@6@6 ||| Indien und Japan Premierminister in Tokio _IS@6@6 ||| Indiens und Japan Premierminister in Tokio S _IS@7@7 ||| Indiens und Japan Premierminister in Tokio _IS@4@7 ||| Indis und Japan Premierminister in Tokio S _IS@5@5 ||| Indisc und Japan Premierminister in Tokio C _IS@6@6 ||| Indisch und Japan Premierminister in Tokio H _IS@7@7 ||| Indischer und Japan Premierminister in Tokio und Japan Premierminister in Tokio R _IS@9@9
.
.
Treffen des indischen und japanischen Premierministers in Tokio S _IS@54@54
</Key_Strokes>
<ST_MT_AL>0-0 1-1 2-2 3-3 4-5 6-4 7-5</ST_MT_AL>
<SEL_ENGINE>PE</SEL_ENGINE>
<MT_APE_AL>0-1 1-1 2-2 3-4 4-4</MT_APE_AL>
<MTM_TRANS_AL>1-0 1-1 0-2 1-3 2-4 3-5 4-6 5-7</MTM_TRANS_AL>
</Log>

Editing summery

Key strokes logs for each editing operations

Source-MT Word alignment

Selected Engine

MT-APE Word alignment

APE-Human Translation Word alignment

FIGURE 6.3: Logs generated by *CATaLog Online*

6.4 *CATaLog_TS*: Beyond Translation Memories

Unmatched parts of the input sentence, which are not present in the TM suggestions, can often be found in some other TM suggestions or in other sentences in the TM database. Therefore, allowing translations of the unmatched parts to be merged into one single sentence in a meaningful way, translation quality of the TM output can be improved and hence post-editing effort can be reduced. Inserting the translations for the unmatched parts into TM suggestions improves translation correctness; however, it may lead to loss of fluency in the generated target text. To avoid this, the TM based translation generation process is guided by the use of parsing, POS tagging and a POS-based back-off n-gram model. The generated translation can be provided as an additional translation suggestion in the TM.

CATaLog (desktop version) generates the top five suggestions based on its own similarity metric (cf. Section 6.3.1). Whenever the post-editor chooses a TM translation suggestion for post-editing, the system tries to fill in the unmatched parts, if any, of that suggestion and presents the user with an additional new translation suggestion. The system components are detailed in the following subsections.

6.4.1 Generating a Dictionary

The main focus of this approach is to fill the unmatched parts of an input sentence by a TM suggestion at the word level. Whenever an unmatched word is found in the input sentence to be translated, the system has to find its translation somewhere in order to propose a complete translation. One way of achieving this is to keep a bilingual dictionary. However, a dictionary is a costly resource for many language pairs. Therefore, to keep it language independent, the system automatically generates a dictionary from the background bilingual corpus available with the translation memory rather than using an external dictionary. For illustration purposes, all the examples presented in this section are in English–Bengali obtained from an English–Bengali parallel corpus of 13,000 sentences. English is considered as the source language and Bengali as the target language. An English–Bengali dictionary is generated from the parallel corpus where English words are stored along with their parts of speech (POS) information and their corresponding

translations in Bengali. In the present work we used the Stanford POS tagger[16] to generate the POS tags for the source side of the parallel corpus. The GIZA++ (Och and Ney, 2003a) implementation of the IBM word alignment model (Brown et al., 1993) is used to produce one to many alignments between source and target language words. From these source–target word alignments the system finds the translation correspondences of the English words available in the parallel corpus. This dictionary is generated offline, only once, and it gets loaded when the TM application is loaded.

The meaning of a polysemous word depends on the context it appears in. In the case of translation, a source word can have completely different translations or may have different suffixes attached to it based on its context. Therefore, to determine which translation is more accurate in a particular context, we look at the neighboring context. Instead of considering the lexical context (i.e., words), we take into consideration the POS context in the present work. In its current version, the system uses a trigram back-off model (cf. Algorithm 6) for determining the contextual translation of a source word. The system generates three contextual dictionaries ($D_{context}$): a ± 2 context based dictionary, a ± 1 context based dictionary and a simple uni-gram dictionary. Here context refers to a POS sequence context. In the ± 2 contextual dictionary, for a particular source word, we store the previous two POS tags, the POS tag of the word under consideration and the next two POS tags. We also store the frequency of a tag sequence (in the training corpus) along with the translation of the word in that context.

Figure 6.4 shows the POS based contextual dictionary entries for the word 'book'. In the second entry in the ± 2 contextual dictionary, the ± 2 context POS tag sequence is MD_PRP_VB_DT_NN; ['2'] represents the zero-based positional index of the POS tag for the word 'book'; 'সংরক্ষণ (*songrakshon*)' is the corresponding translation and '1' appearing at the end represents the frequency of this translation for the word 'book' in this particular POS context. The other two dictionaries also follow the same format.

6.4.2 Finding Translations for Unmatched Parts

In order to find the translation of a non-matching word in the input sentence we perform the operations described in Algorithm 6.

[16] `http://nlp.stanford.edu/software/tagger.shtml`

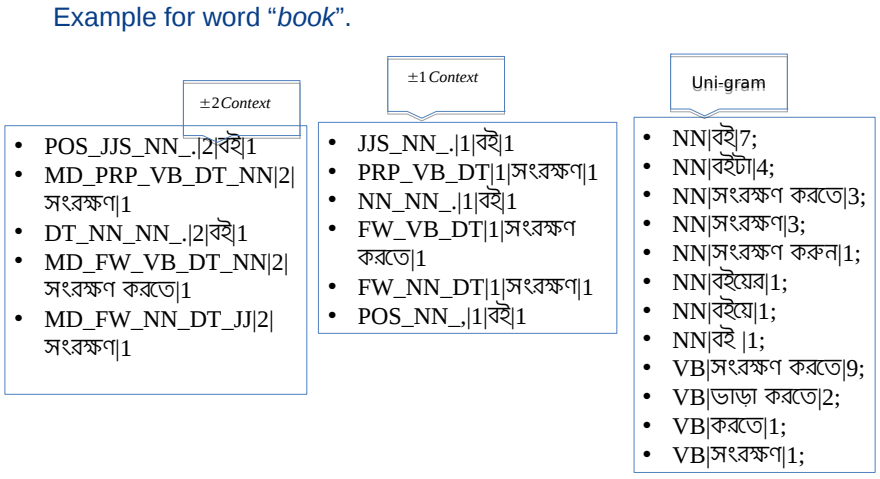


FIGURE 6.4: POS-based context dictionary

In case of multiple matches found in $D_{context}$, the system chooses the most frequent translation from the set of W_t. In case of a frequency tie, which is very unlikely, it chooses any one of the most frequent translations randomly. While doing the POS context matching, we first try to get an exact match. If no exact match is found, the system looks for a basic POS context match. Since *CATaLog* internally uses the TER metric for measuring similarity (cf. Section 6.3.1) between the input sentence and the TM database, as a byproduct, TER also provides the alignment between the input sentence and selected TM source suggestion sentences. From this alignment we can easily find out which words in the input sentence do not match the suggested sentence. The unmatched words (W_u) are searched in $D_{context}$. Considering POS context in the dictionary enables the system to resolve ambiguities for selecting the correct translation for W_u e.g., book: NN|বই; VBD|সংরক্ষণ করা. That is, if the unmatched word in the input sentence is 'book' and it is identified as an NN (noun), the system provides the translation "বই (*boi*)"; similarly, if it is used as VBD (verb), the system picks up the translation "সংরক্ষণ করা (*sangrokshon kora*, English gloss: to reserve)". Thus, POS based dictionary matching reduces the ambiguity to some extent. However, if the word is not present in the dictionary it remains

Algorithm 6: Finding translations of i^{th} unmatched word W_u^i

Data: input W_u^i and $D_{context}$

Result: Return Set of W_t

begin

foreach W_u^i **do**

$POS_{\pm 2} := \{(POS_{i-2}, POS_{i-1}, POS_i, POS_{i+1}, POS_{i+2})\}$; `/* POS context ±2 */`

$POS_{\pm 1} := \{POS_{i-1}, POS_i, POS_{i+1})\}$; `/* POS context ±1 */`

$POS_{uni} := (POS_i)$; `/* POS unigram */`

if W_u^i *and* $POS_{\pm 2}$ *are found in* $D_{context}$ **then**
Return the Set of corresponding W_t

else if W_u^i *and* $POS_{\pm 1}$ *are found in* $D_{context}$ **then**
Return the Set of corresponding W_t

else if W_u^i *and* POS_{uni} *are found in* $D_{context}$ **then**
Return the Set of corresponding W_t

else
Return W_u^i

untranslated. While matching the POS tag we might not find an exact POS tag match. In that case the system tries to find an approximate POS match, i.e., at the basic POS category level (e.g., noun, verb, adjective, etc.).

6.4.3 Finding Positions to Insert Translations

After obtaining the translations (W_t) for all the unmatched words (W_u) from $D_{context}$ (cf. Algorithm 6), we need to find out where to put these W_t in the selected TM translation. Unless the W_t are placed in proper positions, the suggested new translation will become less fluent and unsuitable for post-editing. TM, despite being technologically very simple, has proved itself to be a widely used technology in the localization industry mainly because it presents the user with perfectly fluent translation suggestions for post-editing. Thus, presenting the user with a more accurate but less fluent translation suggestion might not be acceptable. To place W_t in proper positions in the target suggestion translation, the system performs POS tagging[17] and parsing[18] of both the input sentence and the selected source suggestion sentence. Our approach is somewhat similar to Zhechev and van Genabith (2010), however, in case of multiple potential similar structured source subtree

[17]Stanford POS tagger: `http://nlp.stanford.edu/software/tagger.shtml`
[18]Stanford parser: `http://nlp.stanford.edu/software/lex-parser.shtml`

replacements for an unmatched part in the input sentence, they choose a matching source subtree randomly and place the translation in the aligned target subtree position in the target TM suggestion. We used POS tag (cf. Section 6.4.3.1) and parse tree (cf. Section 6.4.3.2) based disambiguation for finding the position(s) of the source token(s) in the TM match for replacement.

6.4.3.1 Finding Position Using POS Tag

First, the system finds a corresponding word (W_c) in the TM source suggestion that does not match with any word in the input sentence. Successively, the system finds the words (W_{ct}) and their positions in the target side of the parallel TM suggestion that W_c corresponds to. The alignment links between W_c and W_{ct} are found from TM source–target word alignment table (cf. Section 6.4.4). Those positions are the potential positions where the W_t can be placed. The W_c can be found by using the POS tag of the W_u, such that the POS tags of W_u and W_c are either the same or from the same POS category. Once a W_c is found and is marked for a W_u, it is not considered for any other W_u. The position of W_c is determined using a POS trigram back off model (cf. Algorithm 7). If multiple candidates are found, the ambiguity is resolved using parse tree information of the input sentence to determine which trigram/bigram sequence is more suitable (cf. Section 6.4.3.2).

6.4.3.2 Finding Position Using Parse Tree

When multiple POS n-gram matches are found, the system resolves this ambiguity using the parse tree of the input sentence. For all the higher order POS n-gram matches, we determine the lowest common ancestor (LCA) node in the parse tree. The n-gram POS sequence choice for which the depth of the common ancestor node is maximized is considered as the most appropriate candidate. If there is a tie, the system chooses one among them randomly. The idea behind choosing the LCA (i.e., maximum depth) is that the lower the common ancestor in the parse tree, the more syntactically coherent the constituent words are. If the LCA is located in an upper level of the tree, the words considered in the n-gram sequence are unrelated and hence the 'corresponding n-gram'

Algorithm 7: Finding W_c for the i^{th} unmatched word W_u^i

Data: input W_u^i and $POS_{context}$

; /* $POS_{context}$ is a set, containing all possible unigram, bigram and trigram contexts of each W_u^i and their corresponding positional information j in TM Source suggestions */

Result: Return {W_c, j}

; /* returns W_c and its position j in the TM Source suggestion */

begin

  foreach W_u^i **do**

    POS_{tri} := {(POS_{i-2}, POS_{i-1}, POS_i), (POS_{i-1}, POS_i, POS_{i+1}), and (POS_i, POS_{i+1}, POS_{i+2})} ; /* Possible trigrams */

    POS_{bi} := {(POS_{i-1}, POS_i), (POS_i, POS_{i+1})} ; /* Possible bigrams */

    POS_{uni}) := (POS_i) ; /* Possible unigram */

    if *any* POS_{tri} *is found in* $POS_{context}$ **then**
      Return {W_c, j}

    else if *any* POS_{bi} *is found in* $POS_{context}$ **then**
      Return {W_c, j}

    else if POS_{uni} *is found in* $POS_{context}$ **then**
      Return {W_c, j}

    else
      Return W_u^i

should be ignored. This motivates the idea behind using the LCA. The process is illustrated using Example 6.1. For the sake of simplicity, we make use of the unigram dictionary to obtain the translation for the unmatched words in the example. However, the system uses a trigram back-off model for this purpose.

Example 6.1.

Input sentence: *i would prefer something in a middle price range .*

TM suggestion: *i would prefer to sit in the back part of the plane .*

TM suggestion translation: আমি বিমানের পিছনের অংশে বসতে পছন্দ করব . *(Gloss: ami bimaner pichoner angshe boste pochondo karbo.)*

Table 6.1 shows the TER alignment between the TM source suggestion and the input sentence along with the edit operations required to turn the TM source suggestion into the input sentence. Table 6.1 shows the word alignment information between the source and target of the TM suggestion.

The W_u in the input sentence in this case are 'something', 'a', 'middle', 'price', 'range'. Unigram dictionary entries for the unmatched words are :

something: NN|একটা কিছু; NN|কিছু; NN|কোন কিছু; NN|কিছু একটা

a: DT|একটা; DT|কোন; DT|এক

middle: JJ|মাঝারি আকারের; JJ|মাঝের

price: NN|দাম; NN|দামটা; NN|মূল্য

range: VBP|দেড়'শ এর মধ্যে বদলাতে থাকে

TM Target Suggestion	TM Source Suggestion	TM Source (POS)	Input Sentence	Input (POS)	Edit Operation
আমি	i	FW	i	FW	M
-	would	MD	would	MD	M
পছন্দ করব	prefer	VB	prefer	VB	M
-	to	TO	-	-	D
বসতে	sit	VB	something	NN	S
-	in	IN	in	IN	M
-	the	DT	-	-	D
পিছনের	back	JJ	-	-	D
অংশে	part	NN	a	DT	S
-	of	IN	middle	JJ	S
-	the	DT	price	NN	S
বিমানের	plane	NN	range	NN	S
.	.	.	.	.	M

TABLE 6.1: TM source–target alignment and TM source–input alignment

We perform the following steps to generate the new TM suggestion for the Example 6.1:

- For every unmatched word (W_u) in the input sentence, the system searches for W_u in TM source suggestions that appear in same or similar contexts as the input sentence. A corresponding word, W_c, found for an unmatched word (W_u) in the input sentence in this way is a potential candidate which could be replaced by W_u.

- Among the unmatched words in the input sentence in Example 6.1, the system first considers the three trigrams: (i) 'would/MD prefer/VB something/NN'; (ii) 'prefer/VB something/NN in/IN'; and (iii) 'something/NN in/IN a/DT' involving the word 'something/NN'.

- Applying Algorithm 7, the third POS trigram matches with the POS trigram 'part/NN of/IN the/DT' in the TM source suggestion. Therefore, 'part' is considered as the corresponding word (i.e. W_c) for the unmatched word (i.e. W_u) 'something/NN' in the input sentence.

- After getting the W_c (i.e., 'part'), the system searches for the position of the corresponding target word W_{ct} in TM target suggestion using the GIZA++ alignment. The GIZA++ alignments between the TM source and TM target suggestion is given below.

  1-1, 3-6, 3-7, 5-5, 8-3, 9-4, 12-2, 13-8

  Here the position index before the hyphen (-) is the word position in the TM source suggestion and the position index after hyphen (-) is the word position in the TM target suggestion. W_c = 'part' is the ninth word in the TM source suggestion and according to the GIZA++ alignment, W_{ct} is the fourth word in the TM target suggestion, i.e., 'অংশে (*angshe*)'. Therefore, W_{ct} = 'অংশে' is replaced by W_t = 'একটা কিছু (*ekta kichu*)', for W_u = 'something' (cf. Algorithm 6). Hence, the TM target suggestion is modified as:

  আমি বিমানের পিছনের একটা কিছু বসতে পছন্দ করব .

- Next, the system tries to find a match for the W_u 'a/DT'. The corresponding POS trigrams are 'something/NN in/IN a/DT', 'in/IN a/DT middle/JJ' and 'a/DT middle/JJ price/NN'. Since the POS sequence 'part/NN of/IN the/DT' starting with 'part/NN' already matched with 'something', this match is not considered again.

However, the system gets a match for the other two trigrams – 'in/IN the/DT back-/JJ' and 'the/DT back/JJ part/NN' where 'the/DT' has not matched with any W_u of the input sentence.

- To resolve the ambiguity we consider the parse tree of the input sentence. The parse tree of the input sentence is shown in Figure 6.5 The numeric values in parentheses in Figure 6.5 represent the depth of the corresponding nodes.

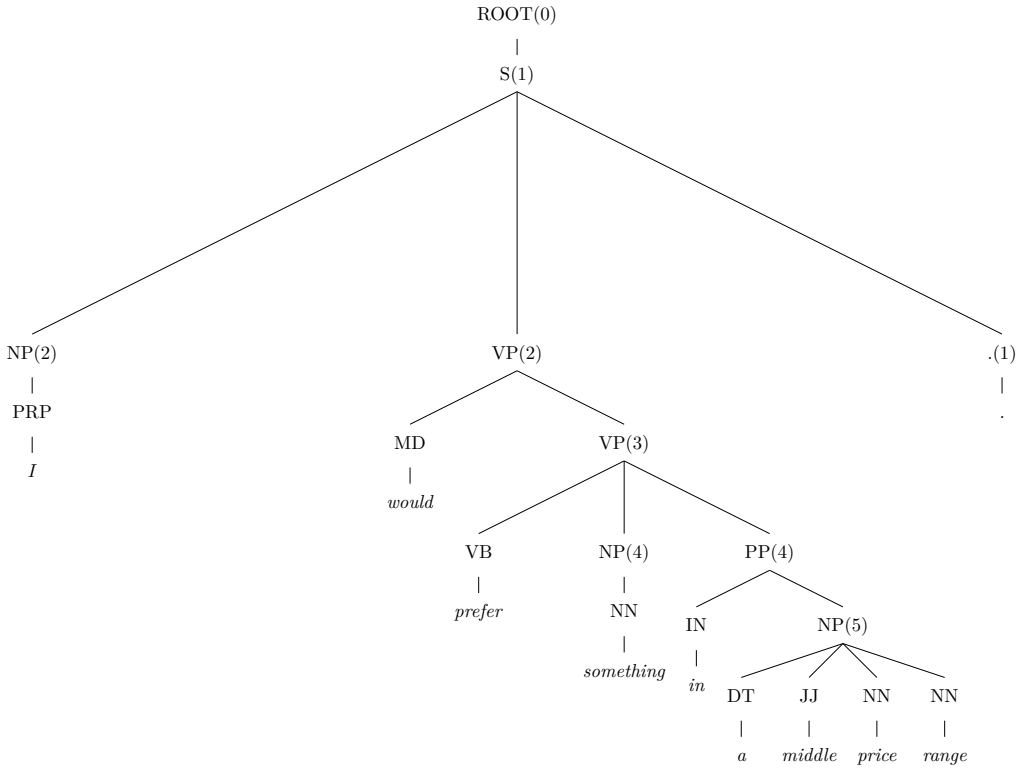


FIGURE 6.5: Parse tree

The trigram 'in/IN a/DT middle/JJ' has the lowest common ancestor at depth 4 whereas 'a/DT middle/JJ price/NN' has the lowest common ancestor at depth 5. We consider the trigram which has the lowest common ancestor at a higher depth (i.e., lower level). Therefore, in this case, the trigram 'a/DT middle/JJ price/NN' is considered and the corresponding matched sequence is 'the/DT back/JJ part/NN' in the TM source suggestion and the word 'the' is the W_c for the W_u 'a'. Subsequently the system looks for the translation W_{ct} for the W_c (seventh word in TM suggestion). However, since there is no alignment corresponding to the seventh source word in the GIZA++ alignment, the translation of 'a' is not placed in the TM suggestion translation.

- Afterwards the system searches for the W_u 'middle/JJ'. The corresponding POS trigrams are (i) 'in/IN a/DT middle/JJ', (ii) 'a/DT middle/JJ price/NN' and (iii) 'middle/JJ price/NN range/NN'. The first two trigrams match with 'in/IN the/DT back/JJ' and 'the/DT back/JJ part/NN' in the TM source suggestion. To resolve this ambiguity the system checks the parse tree again. The POS trigram 'in/IN a/DT middle/JJ' has the LCA at depth 4 while the POS trigram 'a/DT middle/JJ price/NN' has the LCA at depth 5. Therefore, the second trigram is considered and the W_c for 'middle/JJ' is 'back/JJ'. Note that 'back/JJ' is located at position 8 of the TM source suggestion and its translation is 'পিছনের' which is located at position 3 of the TM target suggestion. Therefore the W_t of 'middle/JJ', 'মাঝারি আকারের', is replaced by the third word 'পিছনের' in the TM target suggestion. Thus the modified translation is formed as:

  আমি বিমানের মাঝারি আকারের একটা কিছু বসতে পছন্দ করব .

- The system next searches for 'price/NN' which is translated using 'দাম'. The three POS trigrams to be considered are 'a/DT middle/JJ price/NN', 'middle/JJ price/NN range/NN', and 'price/NN range/NN ./.' . Here the POS sequence 'a/DT middle/JJ price/NN' gets a match with 'the/DT back/JJ part/NN', where 'part/NN' is the corresponding word for 'price/NN'. However, 'part/NN' has already been used earlier; therefore, the system ignores this match. The other two trigrams do not match with any POS trigram in the TM suggestion. Two POS bigrams considered for 'price/NN' are 'middle/JJ price/NN' and 'price/NN range/NN'. Here 'middle/JJ price/NN' matches with 'back/JJ part/NN'; however, it is ignored since the translation position of 'part/NN' has already been replaced. The other bigram does not match either. Therefore the system falls back to the unigram match for 'price/NN'. It matches with 'part/NN' and 'plane/NN'. Since 'part/NN' has already been used, the system considers 'plane/NN' which is at position 12 of the TM suggestion and its translation, 'বিমানের', is at position 2 of the suggestion translation. Therefore, 'বিমানের' is replaced by 'দাম' and the suggested translation is modified as given below.

  আমি দাম মাঝারি আকারের একটা কিছু বসতে পছন্দ করব .

- The system tries to find a match for 'range/NN' later on. However, its trigram, bigram, and unigram POS sequences are either being used already or do not match.

Therefore, its translation is not put in the suggested translation. Finally the word 'বসতে' which is the translation of 'sit' is deleted since 'sit' does not match with any word of the input sentence. Thus, the final translation suggestion is produced as given below.

আমি দাম মাঝারি আকারের একটা কিছু পছন্দ করব .

Since the translations of 'a/DT' and 'range/NN' are not placed in the translation suggestion, their translations 'একটা' and `দেড়শ এর মধ্যে বদলাতে থাকে', respectively, are added to a list and are shown to the post-editor as suggestions. The post-editor can directly use those translations without typing them and can put them in the proper place. In this way the system modifies the TM translation suggestion to generate more appropriate translation candidates. These translation candidates can be post-edited with much less effort.

6.4.4 Placing Translations of Unmatched Words in a TM Suggestion

After finding the positions of the W_c in the selected TM source suggestion, we determine the positions of the corresponding W_{ct} in the TM target suggestion using the source–target alignment[19] for the TM sentence pair. These positions in the TM suggested translation are the potential positions where the translation of the unmatched word could be placed. Since GIZA++ generates one-to-many alignments between source and target, three situations can arise. The length (in terms of number of words) of the translation W_{ct} of W_c could be equal to, shorter, or longer than the length of the translation W_t of W_u.

The potential positions for inserting W_t may also be continuous or discontinuous in the TM suggestion translation. If W_{ct} is continuous, W_{ct} is simply replaced by W_t. If W_{ct} is discontinuous, words in W_{ct} are replaced by words in W_t one by one. If W_{ct} is longer than W_t, the additional words in W_{ct} are simply deleted. If W_t is longer than W_{ct}, then the additional words in W_t are appended with the last word replaced in W_{ct}.

POS tags and parse tree based sentence fusion in TM works well when the input sentence and the suggestion translations are similar in length. If the input sentence and the suggestion sentence differ widely in length, their parse trees may also differ significantly.

[19]TM source target alignment is an offline process and pre-trained using GIZA++

In such cases, the suggested translation may lead to loss of fluency in the target translation. Therefore, we consider only those sentences in the translation suggestion whose lengths are within a predefined limit with respect to the length of the input sentence. TM retrieved source suggestions that are either above or below this predefined limit are discarded.

6.5 *CATaLog_TS_ReRank* – Re-ranking of the TM Suggestion Translations

CATaLog produces five most relevant TM translation suggestions. After producing the newly generated translations corresponding to the TM suggestions (cf. Section 6.4), the first option originally chosen by the TM module might not remain the best translation option. This is also evident from the experimental results obtained (cf. 'first' vs. 'best' in Table 3.4, Section 6.6). This motivated us to perform re-ranking of the produced translation suggestions in order to bring the most suitable translation to the top. Re-ranking deals with various features including:

- Language model probability
- Length of the input sentence
- Length of source side TM suggestions
- Number of unmatched words for which translations are successfully inserted into the corresponding TM translation suggestion
- The original similarity score produced by the *CATaLog* system

CATaLog calculates similarity scores on the basis of TER alignment. The similarity score is computed in Equation 6.2, where n_m and s_m refer to the number of matches and match reward scores, respectively; e_i refers to four types of edit operations – *insert*, *delete*, *substitute* and *shift*; n_{e_i} and c_{e_i} refer to number of e_i edit operations required and the corresponding edit cost, respectively. Thus we reward matches and penalize edits to arrive at the final similarity score.

$$S = n_m \times s_m - \sum_{i=1}^{4} n_{e_i} \times c_{e_i} \tag{6.2}$$

Let us consider, s_m=0.80, c_{e_1}=0.20 (deletion cost), c_{e_2}=0.50 (insertion cost), c_{e_3}=0.70 (substitution cost), and suppose TER alignment between an input segment and TM source segment is "MMDIMISMM". Therefore, the corresponding original TM match score ($OTMS$) is calculated using Equation 6.2 as follows.

$$OTMS = 0.80 \times 5 - 0.20 - 0.50 \times 2 - 0.70 = 2.1$$

Now, let us consider that *CATaLog_TS* has successfully inserted the translation of two words represented as 'I' in the TER alignment. Therefore, two additional match_reward scores are added with $OTMS$ to arrive at the new TM match score ($NTMS$).

$$NTMS = OTMS + 2 \times 0.80 = 3.7$$

We estimate the fluency score of a translation suggestion using a language model and the estimated length of the actual translation of the input sentence. We use a 5-gram language model with back-off smoothing trained on the target side of the TM corpus. We use the SRILM toolkit (Stolcke, 2002) for language modelling. The language model score is normalized by the length of the translation suggestion.

We also use the concept of brevity penalty to penalize a translation if its length is much smaller or longer than the estimated reference translation. Since no reference translation is available for the input sentence, we estimate the length of the translation based on the length of the input sentence. Let the length of the input sentence be SL and the length of translation suggestion be TL. We assume that the reference translation length ($RefLen$) will be in the range between $0.8 \times SL$ and $1.2 \times SL$. If the candidate translation length is out of this range, we assign it a length penalty based on Algorithm 8.

We calculate a fluency score using the language model score (LMS) and length-based penalty (LP) as in Equation 6.3.

$$smoothness\_score = LMS \times LP \tag{6.3}$$

Algorithm 8: Calculate Length Based Penalty

Data: input SL, TL

Result: Return LP ; `/* length penalty */`

Initialization:

$LP \leftarrow 0$;

$minRefLen \leftarrow 0.8 \times SL$; `/* minimum RefLen */`

$maxRefLen \leftarrow 1.2 \times SL$; `/* maximum RefLen */`

$diff \leftarrow 0$;

begin

```
if TL ≥ minRefLen AND TL ≤ maxRefLen then
    LP ← 1.0;
    return LP;
if TL ≤ minRefLen then
    diff ← minRefLen − TL;
if TL ≥ maxRefLen then
    diff ← TL − maxRefLen
LP ← e^(−diff/SL);
return LP;
```

Finally, we re-rank the translation suggestions based on the final score computed as in Equation 6.4.

$$final\_score = smoothness\_score \times NTMS \tag{6.4}$$

6.6 Experiments with the Generated TM Suggestion

In this section we describe automatic evaluation of the new "translation" feature in the *CATaLog* system called *CATaLog_TS* (cf. Section 6.4). *CATaLog_TS* was compared against *CATaLog* (Nayek et al., 2015) and the Moses (Koehn et al., 2007) implementation of the PB-SMT model. We used an English–Bengali parallel corpus which contains 13,000 sentences. This parallel corpus serves as the TM for both *CATaLog* and *CATaLog_TS*. The baseline PB-SMT system is also trained on the same parallel corpus. For building the PB-SMT system, we set the maximum phrase length to 7 and a 5-gram language model was trained using KenLM (Heafield, 2011) on the target side training data. Parameter tuning was carried out using Minimum Error Rate Training (MERT) (Och, 2003)

on a held-out development set containing 500 sentences. Two different test sets were used for evaluation: **testset1** contained 100 sentences and **testset2** contained 500 sentences. We evaluated our system using two well-known automatic MT evaluation metrics: BLEU (Papineni et al., 2002) and TER (Snover et al., 2006c).

CATaLog_TS provides five translation suggestions based on the top five matches retrieved from the TM by *CATaLog*. The term 'First' in Table 6.2 refers to the first (i.e. the top ranked) translation suggestion provided by the *CATaLog* or *CATaLog_TS* system. The term 'Best' refers to the best translation suggestion chosen by sentence level BLEU (S-BLEU) among the five translation suggestions.

Table 6.2 shows that, as far as the 'First' translation suggestion is concerned, *CATaLog_TS* provides 2.13 and 2.03 BLEU points absolute (22.4% and 19.2% relative) improvement over *CATaLog* for testset1 and testset2 respectively. The respective improvements are 8.21 and 9.64 points (12.8% and 14.6% relative) for TER. Similarly, for the 'Best' translation suggestion, the improvements provided by *CATaLog_TS* over *CATaLog* for testset1 and testset2 are 3.59 and 1.91 BLEU points (29.8% and 14.5% relative) and 10.99 and 6.24 TER points (17.1% and 10.3% relative) respectively.

More importantly, for testset1, *CATaLog_TS* 'Best' performs better than the state-of-the-art PB-SMT system in both BLEU and TER. However, in case of testset2, *CATaLog_TS* 'Best' performs better according to TER while Moses fares better according to BLEU. This is probably due to the fact that the Moses system was tuned with the BLEU evaluation metric.

From Table 6.2, we can conclude that *CATaLog_TS* always performs better than *CATaLog*. The TER scores for *CATaLog_TS* are much lower than those for *CATaLog* for both 'First' and 'Best' translation suggestions. BLEU scores also reflect the same trend. Comparison with the Moses system reveals that *CATaLog_TS* provides the lowest TER scores for both the test sets, even if we just consider the 'First' translation suggestion. However, Moses is ahead on testset2 while *CATaLog_TS* fares better on testset1 according to BLEU.

Table 6.2 also shows that after re-ranking the top suggestions, the *CATaLog_TS_ReRank* system provides a much higher BLEU score and lower TER score compared to Moses for testset1. However, in the case of testset2, the BLEU score of the *CATaLog_TS_ReRank*

system is better than the 'Best' option of the *CATaLog_TS* system, but lower than that of Moses. However, for both test sets, the TER score of *CATaLog_TS_ReRank* is considerably better than the other systems. It is to be noted that the *CATaLog_TS* 'Best' system output was decided on the basis of the S-BLEU score, while for the actual evaluation purposes we use BLEU. BLEU is a system level score and does not perform well at sentence-level evaluation; hence the BLEU and TER scores of *CATaLog_TS_ReRank* are better than those of *CATaLog_TS*'s 'Best' system.

Testset	**System**		**Performance**	
			TER	**BLEU**
Set1	*CATaLog*	First	64.10	9.49
		Best	64.41	12.03
	Moses		57.12	14.57
	*CATaLog*_TS	First	55.89	11.62
		Best	53.42	15.62
	*CATaLog*_TS_ReRank		**48.49**	**18.07**
Set2	*CATaLog*	First	65.98	10.58
		Best	60.82	13.15
	Moses		58.44	**18.34**
	*CATaLog*_TS	First	56.34	12.61
		Best	54.58	15.06
	*CATaLog*_TS_ReRank		**53.83**	15.68

TABLE 6.2: Systematic comparison between *CATaLog*, *CATaLog_TS*, *CATaLog_TS_ReRank* and Moses.

6.7 User Studies with *CATaLog Online*

The English–German TM engine developed for *CATaLog Online* is based on the data described in Chapter 4. The same data were also used to build the *CATaLog Online* MT and the internal APE engines. The test data for human evaluation is collected from the WMT-2015 test set data. We randomly choose 400 sentences from the test set for human evaluation.

In order to evaluate *CATaLog Online*, we conducted experiments with three professional translators. All of them are native speakers of German with at least two years of experience in translation. Before the user study was carried out with the translators in a controlled environment, they were provided with the task guidelines and a short introduction about *CATaLog Online*. The translators were asked to perform English to German translation of 200 news sentences with *CATaLog Online* by choosing and editing one of the following three options[20].

(a) the output of *CATaLog Online*'s automatic post-editing system (APE)

(b) the suggestion from *CATaLog Online*'s internal translation memory (TM)

(c) translation from scratch (None)

The selection of options (a) or (b) entails that translators will perform post-editing in most of the cases, while for option (c) they will have to translate from scratch without any help from the TM or the APE.

	200 sentences			**100 sentences**		
	Trans1	Trans2	Trans3	Trans1	Trans2	Trans3
APE	160	169	161	74	85	82
TM	1	16	0	1	7	0
None	39	15	39	25	8	18

TABLE 6.3: Selection of suggestions by translators in *CATaLog Online*.

From the set of 200 sentences each translator received, 100 were repeated (i.e., each translator received 100 common out of 200 sentences), allowing us to measure the agreement between the three translators. Since *CATaLog Online* records an extensive editing log, we collected information concerning the engine used in translation (APE, TM, or translation from scratch), the number of deletions, insertions, substitutions and shifts (both words and characters) as well the editing time (in seconds) for each segment.

The first analysis of the logs is presented in Table 6.3 which gives an overview of the selected suggestions. Table 6.3 shows that all three translators showed a tendency to

[20]In case of tie, i.e., equal quality translations, translators can chose any of the suggestions between APE and TM. However, during experiment, translators did not face this situation.

	Selected suggestions			**E**diting time			**N**umber of edits		
	Trans1	Trans2	Trans3	Trans1	Trans2	Trans3	Trans1	Trans2	Trans3
Trans1	-	0.08	0.20	-	-0.16	-0.06	-	0.49	0.42
Trans2	0.08	-	0.05	-0.16	-	-0.13	0.49	-	0.26
Trans3	0.20	0.05	-	-0.06	-0.13	-	0.42	0.26	-

TABLE 6.4: Cohen's κ measuring agreement for the selected suggestion, editing time and number of edits.

choose the suggestion made by the automatic post-editing system and perform further editing on it. The APE system achieves a selection rate of around 80%. The remaining sentences are either translated from scratch or by using the suggestions provided by the TM. The ratios of the three options chosen by the individual translators are similar for both the entire test set (200 sentences) and the set containing the common sentences (100).

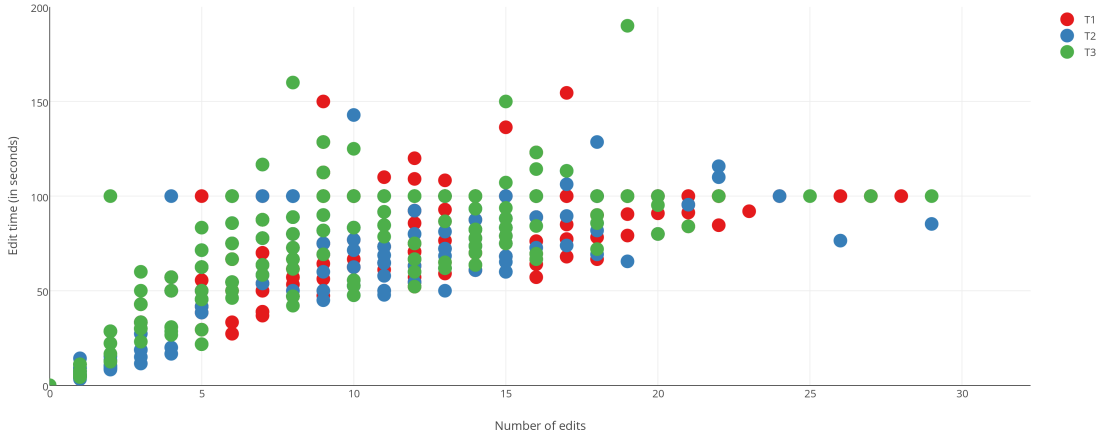


FIGURE 6.6: Correlation between the number of edits and edit time.

For the 100 sentences in common we measured pairwise inter-rater agreement between translators by computing Cohen's κ (Cohen, 1960) for different variables. We concentrated on the suggestions used in the translation process (APE, TM, or translation from scratch), editing time, as well as the overall number of edits. The pairwise inter-rater agreements are presented in Table 6.4.

From Table 6.4 we observe that translators agree only in terms of overall number of edits. Editing time and the selection of a specific suggestion (APE, TM, or translation from scratch) are parameters on which the translators do not agree. We computed Pearson's

correlation coefficient ρ to test whether the total number of edits (with a low κ) influences the post-editing time (with a high κ). We achieved a ρ value of 0.10 indicating a slight correlation between these two parameters. Figure 6.6 depicts a suggestion that requires a higher number of edits also requires more edit time. However, we also noticed cases in which a high number of edits did not require much editing time and vice versa.

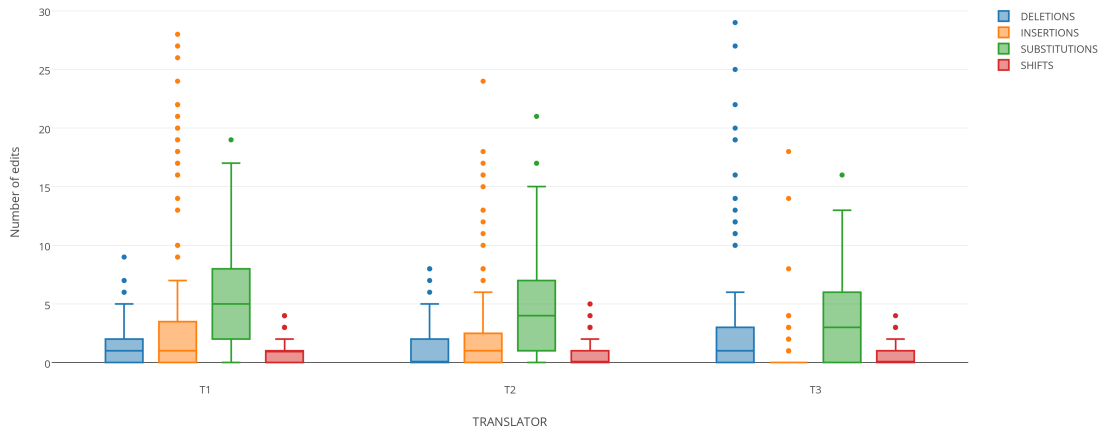


FIGURE 6.7: Box plot distributions of the different types of edits for the three translators (T1,T2, and T3).

Taking a closer look at the type of edits performed, we noticed that the edits with the highest frequency are substitutions, followed by insertions, deletions and shifts. Figure 6.7 depicts the variance of the four edit types. Concluding on the user studies described in this section, we can say that translators have a clear preference for choosing the output of the APE system for performing their translation task, even though they do not make the same choice for the same segments. In terms of editing time, the data show that, in this setting, time is a translator-dependent variable, influencing the low correlation coefficient with respect to the number of edits.

The users also provided informal feedback regarding the tool. The translators rated the tool by comparing it to other CAT tools in terms of usability. The main positive and negative feedback about *CATaLog Online* are summarized below.

Positive Feedback

- The unique coloring system in *CATaLog Online* – offered by none of the existing TM based CAT tools – helped to complete the editing of suggestions from the TM.

- APE suggestions were often really helpful.
- The arrangement of the suggestions in *CATaLog Online* was an advantage.
- Displaying the source sentence without opening a new window, facilitating the comparison between source and suggested target translations, was also rated as positive.

Negative Feedback

- Sometimes verbs were entirely missing or the word order was very unnatural in MT and APE suggestions.
- *CATaLog Online* lacks certain functionalities like a spell-checker and other quality assessment (QA) features.
- It would be useful to have more suggestions from the TM; currently *CATaLog Online* offers just one.
- The unavailability of keyboard shortcuts for navigation between segments, saving a segment, opening a concordance search or copying a term from the glossary is a disadvantage..
- Missing the possibility to display the layout of the source text is a negative.

6.8 Conclusions and Future Work

The chapter presents a new free open-source CAT tool and post-editing interface entitled *CATaLog Online* which offers translation suggestions from TM, MT and APE. The tool is specifically targeted towards improving post-editing productivity and user experience with CAT. A novel feature in the tool is a color coding scheme that highlights matching and irrelevant fragments in suggested TM segments. Color coding the TM suggestion makes the decision process easy for the desktop version user as to which TM suggestion to choose and work on and it also guides the translators as to which fragments to post-edit on the chosen TM translation. The similarity metric employed in the tool makes use of TER, the Needleman–Wunsch algorithm and Lucene retrieval to identify and re-rank relevant TM suggestions. The tool keeps track of all post-editing activities and records

detailed logs in well-structured XML which is beneficial for incremental MT/APE and translation process research.

User evaluation of *CATaLog Online* revealed that translators have a clear preference in choosing the output of the APE system for performing their translation task. They also evaluated positively the color coding scheme for the TM suggestions as well as the arrangement of the suggestions within the translation interface. Results of our experiment conducted with professional translators in an industrial environment (Nayek et al., 2015; Nayak et al., 2016; Pal et al., 2016e,d) show that *CATaLog Online* reflects both user and research perspectives. This addresses our RQ6, i.e., "How can human interaction with CAT tools be optimized in existing MT workflows?". The informal feedback revealed that features such as a spell-checker, QA features and keyboard shortcuts could further improve the tool.

In this chapter, we also reported the introduction of another important function of TM: TM based translation generation. Traditionally, TMs do not generate any new translations; therefore, we present a step beyond traditional TM. Furthermore, this improves HCI issues with TM since this new functionality generates a new translation based on the translation template chosen by the user. Although, this functionality is not currently available in *CATaLog Online*, automatic evaluation of our prototype implementation reveals that this improves TM suggestion quality. This also reduces the human translation effort and answers RQ6 raised at the beginning of this chapter. In future, we will include this functionality in the online version.

CAT functionality can be improved in many different ways. We are exploring contextual, syntactic and semantic features which can be included in similarity score calculation to retrieve more appropriate translations. Another improvement we are currently working on concerns tuning the weights of the different edit operations to optimize system performance.

Chapter 7

Conclusions and Future Work

7.1 Research Contributions and Questions Answered

In this thesis we have posed six research questions (RQ1 through RQ6, listed below) and developed technologies to address each one of them.

RQ1: *How can MT for low resource languages be improved?*

In Chapter 3, we showed that parallel text fragments extracted from a comparable corpus built from Wikipedia articles were able to bring about significant improvements in the performance of an existing machine translation system. For low resource language pairs our approach can help to improve state-of-the-art machine translation quality. Manual inspection on a subset of the output revealed that the additional training material extracted from comparable corpora effectively resulted in better lexical choice and fewer OOV words than in the baseline output. As the parallel text extracted from comparable data does not belong to any particular domain, this work also shows that out of domain data can be useful to enhance the performance of a domain specific MT system for low resource languages. Although we have successfully shown that additional parallel resources extracted from comparable corpora can improve machine translation for low resource language pairs (e.g., English–Bengali), it remains an open research question how our approach would scale to well resourced language pairs with large amounts of parallel data. Substantial further research is required in the area of comparable corpora for MT.

Technological contributions for RQ1:

- Textual entailment and extraction of parallel fragments of texts which is one of the novel contributions in comparable corpus research.

RQ2: *How can SMT better profit from the existing training data?*

Our research reported in Chapter 4 shows how effective pre-processing of NEs and MWEs in the parallel corpus, and their alignment and integration (directly and indirectly) into PB-SMT and forest based SMT can improve system performance. Automatic prior alignment of MWEs, NEs, and example based phrase pairs and their integration into the word alignment model using additional training examples or hybrid alignment techniques improve the system performance significantly. Chapter 4 also presented a method of source *chunk* pre-ordering based on word alignment. Source chunks are reordered based on their associations with the target words and the target word order. The testset is reordered using monolingual PB-SMT built on the original source training data and the reordered source training data. Our experiments showed that word alignment based source *chunk* pre-ordering is more effective than word alignment based source *word* pre-ordering and tree-based reordering and produced statistically significant improvements on both. On manual inspection we found significant improvements in terms of word alignments. This method also reduces the data sparsity problem and reduces the model size. The pre-ordering method presented in Chapter 4 has the advantage that it does not require any language specific tools like parsers except a chunker for the source language.

Technological contributions for RQ2:

- Improved utilization of parallel data using word-alignment based pre-ordering and pre-aligned example based phrase pairs and terminologies including MWEs and NEs.

RQ3: *What could improved hybrid implementations of MT be like?*

Chapter 4 reports a study on integrating hybrid word alignment in forest to string based statistical machine translation (FSBSMT). Experimental results on an English–Bengali dataset show that FSBSMT with Berkeley alignment brings about a huge improvement

(69.83% relative, 8.75 absolute BLEU points) over state-of-the-art hierarchical Phrase based SMT (HPBSMT). Systems like HPBSMT which work only with 1-best parse tree may suffer from parsing errors. FSBSMT alleviates this problem by considering packed forest of k-best parses. Additional integration of prior aligned named entities and EBMT phrases in terms of additional training examples and the inclusion within hybrid word alignment into the proposed system also brings further improvements. The enhanced system provides 78.5% relative (9.84 absolute BLEU points) improvement over the baseline HPBSMT system and and 5.12% relative improvement (1.09 absolute BLEU points) over an FSBSMT system with Berkeley alignment. Chapter 4 also shows that a hybrid system with NE/MWE alignment, EBMT phrases, and single-tokenized source MWEs results in the best performing system. However, a confusion network-based system combination outperforms all the individual MT systems. The fact that the systems were tuned with BLEU scores may be one of the reasons behind the poor TER scores produced by the systems. However, with the emergence of new technologies like neural MT, deep learning, etc., the issue of hybrid MT will continue to be remain an open research question and subject to further experimentation.

Technological contributions for RQ3:

- Parallel combination of word alignments.
- Hybrid implementation of MT in a multi-engine pipeline.

RQ4: *How can we build an effective automatic post-editing system which can improve the translation quality of the first-stage MT system?*

Chapter 5 shows that the use of alignment combination models including both statistical and edit-distance based methods in our hybrid word alignment model for APE improves the translation quality over raw MT text. By improving word alignment, the APE system automatically acquires better lexical associations and the 'hybrid' PB-SAPE system provides improvements over the raw Google MT baseline. The proposed system combination based APE approach (SC-APE) was successful in improving over the baseline APE system (PB-APE basic) performance. Additionally, we showed that a neural network based APE system provides statistically significant improvements over existing state-of-the-art APE models and produces significantly better translations than the Google Translate system

which is a difficult system to beat. This enhancement in translation quality through APE should reduce human PE effort.

Technological contributions for RQ4:

- System combination (parallel combination of alignment systems at the level of the APE and parallel combination of APE and (first stage) MT systems) in the APE stage of a sequential MT-APE combination.
- Introduced neural APE.

RQ5: *To what extent is an APE system able to reduce final post-editing effort in terms of increasing productivity?*

In Chapter 5 we showed that parallel system combination in the APE stage of a sequential MT-APE combination yields substantial translation improvements both measured in terms of automatic evaluation metrics as well as productivity improvements measured in a post-editing experiment. We also showed that system combination on the level of APE alignments yields further improvements. Overall our APE system yields a statistically significant improvement of 5.9% relative BLEU over a strong baseline (English–Italian Google MT) and 21.76% productivity increase in a human post-editing experiment with professional translators.

Technological contributions for RQ5:

- Productivity improvements in real-life scenario.

RQ6: *How can human interaction with CAT tools be optimized in existing MT workflows?*

Chapter 6 presented *CATaLog Online*, a free online CAT tool. We discussed three main components (MT, APE and TM) of the tool and how they can be used in the translation workflow. To the best of our knowledge, *CATaLog Online* provides a wider range of logs than any other commercial CAT tool in the market. This information is very important for

translation process research and translation project management. The tool also supports a polling system developed as a resource for MT and TM evaluation.

We enhanced *CATaLog Online* in terms of both user perspective and translation process. The user perspective includes: the color-coding, analytical summaries of post-editing activities, and well-structured XML formatted logs. The XML formatted logs can be customized according to the user's choice, e.g., the user can download the entire logs or some specific logs for a particular translation job.

In terms of translation process research and development perspectives, we implemented several functions in *CATaLog Online*, e.g., recording word alignments between Source–MT, MT–PE and source–PE, which will be beneficial for incremental MT and incremental APE. Using the post-editing information we would like to build and integrate a further enhanced APE system into *CATaLog Online* which can improve the background MT system output.

In Chapter 6, we furthermore introduced an improved desktop version of *CATaLog*: *CATaLog_TS* and *CATaLog_TS_Rerank*, where we explored how translations of unmatched parts of an input sentence can be discovered and inserted into TM suggestions (generated by the CAT tool) using parse tree and POS tags information to form a new translation which is more suitable for post-editing and can reduce post-editing efforts. In this part of our research, we are beginning to blur the distinction between TM and MT.

Finally, at the end of Chapter 6, we presented a study to quantify the extent to which translators are faster or more productive using *CATaLog Online*.

Technological contributions for RQ6:

- Color coded TM translation suggestions (highlighted TM source and corresponding target fragments are shown in the same interface).
- A wide range of editing logs.
- Alignment between source, TM/MT/APE and the results of human PE.
- Improved TM similarity measure and search technique.
- Additional translation option from APE which learns from human post-edited data.

7.2 Future Work

Based on the work presented in the thesis, several research directions can be explored in future which are listed below.

Comparable Corpora: A future direction would be to propose a scalable and computationally less complex parallel fragment extraction method from comparable corpora. Another important area for further research is on building an MT system entirely from comparable corpora, i.e., without the availability of any seed parallel corpora. In terms of comparable corpora research there are several other options to be explored, such as:

1. To experiment with more advanced and faster methods of collecting comparable corpora from the Web.

2. To evaluate the similarity of documents across languages in a collection of comparable documents by comparing different similarity schemes and propose improved document similarity methods. News corpora represent a relatively untapped source of comparable corpora which do not come with document alignment. In order to make use of news corpora for parallel text extraction, they need to be aligned at the document level first. Paragraph (or sub-document) level alignment is also another important future area of research in this direction.

3. To explore alternate ways for discovering parallel fragments from comparable corpora using computationally less expensive methods.

Concerning the hybrid approaches to MT, APE and the entire translation workflow, there are several directions in which research could be extended.

Neural MT: The translation model in state-of-the-art SMT which represents the translation knowledge derived from the parallel corpus is a by-product of word alignment (e.g., using alignment can be based on other models, like, Berkeley word alignment (Liang et al., 2006), IBM models (Brown et al., 1993)). However, despite being the backbone to the SMT model, the IBM models suffer from erroneous alignment mappings; they can not properly handle *many-to-one*, *one-to-many* and *many-to-many* alignments (Marcu, 2001; Koehn et al., 2003). Moreover, each component of SMT produces errors that propagate through the translation pipeline. While end-to-end NMT approaches solve these issues to

a certain extent and can therefore lead to better translation quality, they are susceptible to other limitations. Below, we list some of the limitations of NMT and mention some future areas of research that could tackle them.

- NMT systems often focus on the most frequent words in the training corpus and ignore the rest, considering them as unseen or unknown. Although this reduces the model complexity, it causes a serious out-of-vocabulary (OOV) problem. Recently proposed subword-based NMT (Sennrich et al., 2016a) helps to reduce the OOV problem to some extent. However, the OOV problem is still considered as a serious issue in NMT.

- The NMT decoder may lack coverage with respect to source sentences and therefore, favor short translations. This, in turn, could lead to inadequate translations.

- NMT does not utilize the benefits from target side monolingual data and target language models which have already been proven to improve translation quality in SMT. In NMT, monolingual target language data also provides substantial improvements. Gülçehre et al. (2015) incorporate a separately trained RNN language model into the NMT model through shallow or deep fusion. Sennrich et al. (2016b) proposed another effective solution to improve NMT performance; they used parallel synthetic data where "source sentences" are obtained from automatically translating the monolingual target sentences through back-translation.

Potential future areas of research to circumvent these limitations could be:

- OOV problems can be solved to some extent using pre-processing and post-processing. Named entities (NEs) often raise OOV issues in MT. During preprocessing, NEs can be identified from the input text and during post processing untranslated NE-OOVs can be just carried over for source–target language pairs having the same script or can be transliterated using dictionary look up (e.g., Wikipedia parallel NE dictionary or parallel terminology bank) or using a dedicated transliteration model in case of different scripts. Replacement of non-NE-OOVs by appropriate in-vocabulary synonyms or word similarities (Singh et al., 2016) during pre-processing can help reduce OOVs in the translation output.

- Alleviate the inadequate translation problem in NMT by using a word rewarding feature within a soft coverage vector to guarantee all source words are translated.
- We will extend the back-translation work proposed by Sennrich et al. (2016b) by using comparable corpora. Instead of using parallel synthetic data, we will explore back-translation to extract parallel segments from comparable corpora. We would also like to apply a neural or statistical language model to re-rank the n-best NMT output to improve translation quality.
- We will work towards factor-based NMT (García-Martínez et al., 2016) as in factored SMT (Koehn and Hoang, 2007) where different linguistic features are considered as different factors.
- To date, only Chen et al. (2016) treats the task of MT as a multi-objective optimization problem. Chen et al. (2016) proposed an effective way for biasing the attention mechanism using a guided alignment training approach to improve translation quality. They expressed a multi-objective optimization task as a single-objective one by means of a linear combination of two loss functions: the original and the new alignment-guided loss. There is a lot of scope for integrating multi-objective optimization frameworks into NMT which we will investigate to handle different linguistic information as different types of objective functions.

Future work in this direction will result in novel technology for training and exploiting NMT engines within the hybrid framework or within *CATaLog Online* for end-to-end high quality translation. Furthermore, future work should also carry out an empirical evaluation of SMT (hybrid) with NMT (hybrid/ensembled (Luong et al., 2015a)) under real-life scenarios.

Neural APE (NNAPE): Future work will also investigate a fully integrated second-stage MT (APE), based on a neural network approach (cf. Figure 5.8, Neural APE (Pal et al., 2016c)) that will improve translation quality and minimize translation effort and cost by exploring a character/sub-word (Sennrich et al., 2016a) (using byte pair encoding) based APE system to rectify morphological errors.

Pal et al. (2017) presented a neural APE model that extends the attention based NMT model to traditional word alignment models and utilizes agreement of bidirectional models for alignment symmetry. The attentions are encouraged to symmetrization in both

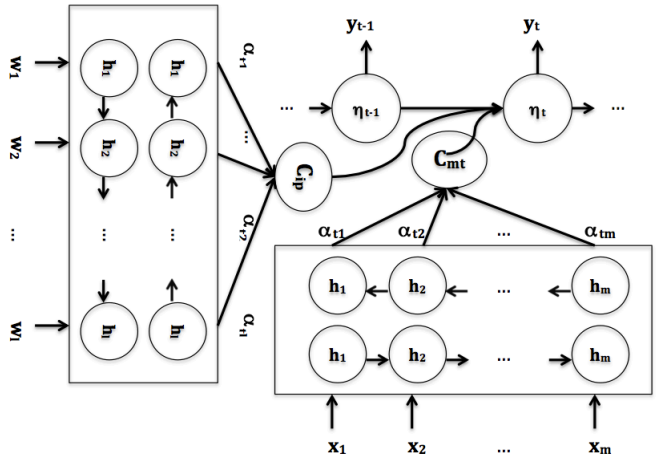


FIGURE 7.1: Generating the t^{th} TL_{pe} word y_t for a given TL_{mt} ($\mathbf{x}$) and SL_{ip} ($\mathbf{w}$).

translation directions. We will extend their approach and also focus on resolving word ordering error by training an alignment model, translation model as well as the reordering model jointly by using three different objective functions within a single optimization framework.

To enhance the NNAPE model described in Chapter 5, we will develop a similar architecture as in Figure 5.8; however, it will have another parallel input layer $\mathbf{w}$, which is another bidirectional RNN encoder for input source sequences, i.e., $P(y_t|y_1, ...y_{t-1}, \mathbf{w}, \mathbf{x})$ (cf. Figure 7.1): by encoding a variable-length sequence of tokens in the source language SL_{ip} (e.g. $\mathbf{w} = w_1, w_2, w_3...w_l$) into a fixed-length vector representation (C_{ip}) as well as TL_{mt} (e.g. $\mathbf{x} = x_1, x_2, x_3...x_m$) into a fixed-length vector representation (C_{mt}) and then decoding a given joint representation of C_{ip} and C_{mt}, back into a variable-length sequence of TL_{pe} (e.g. $\mathbf{y} = y_1, y_2, y_3...y_n$). This model will take SL_{ip} as well as TL_{mt} as input and provide TL_{pe} as output.

Like the NNAPE model in Figure 5.8, this model can also be designed by using an attention based soft alignment model (Bahdanau et al., 2015) which provides a matching score between the inputs around source input position i, MT output position j and the PE output at position t.

Integration of a dynamic incremental APE framework in the MT-APE workflow: One of the main criticisms of the state-of-the-art MT technologies is that translators frequently find the same errors to be corrected in the output of the translation systems (cf. Chapter 5). APE systems described in Chapter 5 can automatically correct systematic

errors to some extent. Future work will be carried out to develop an incremental/online setting where human post-edited/complete MT output that is continuously fed back to the MT or NNAPE system to continuously improve its quality as described in (Chatterjee et al., 2015a) for phrase based APE. The objective is to implement a dynamic, incremental, active learning MT-APE framework, where the MT/APE system will learn from users' feedback or user corrections. The inclusion of online learning techniques into the interactive NMT framework is the next research direction, in order to provide NMT to build more adaptive and productive translation systems. Initial development of such interactive NMT systems has been addressed in (Knowles and Koehn, 2016; Peris et al., 2017).

The future direction towards the improvement of *CATaLog* and *CATaLog Online* would be as follows:

- **Applying the data captured by the polling system and the log information:** We plan to use the polling system and the information obtained in the log functions to investigate translation quality not only at the segment level but also at the document level (Scarton et al., 2015).
- **Rank the translation examples based on their syntactic similarity with the input sentence:** If two TM matches have the same lexical similarity, but one of them is syntactically more similar to the input sentence, then the TM should give priority to the syntactically more similar TM segment. This could be another important area of research in TM match.
- **Rank the TM matches based on their semantic similarity with the input sentence:** If two TM matches have the same lexical similarity, then prefer the TM match for which the differing words in the TM source are semantically closer to the differing words in the input sentence (Gupta et al., 2015a). This could be implemented, e.g., in terms of neural TM matching based on words embeddings (word2vec) for words or segment level similarity.
- **Tuning the edit weights:** Tuning the edit costs corresponding to the different types of editing operations would be an important future work.

Appendix A

CATaLog Online: A Web-based CAT Tool for Distributed Translation with Data Capture for APE and Translation Process Research

We present a free web-based CAT tool called *CATaLog Online* which provides a novel and user-friendly online CAT environment for post-editors/translators. The goal is to support distributed translation where teams of translators work simultaneously on different sections of the same text, reduce post-editing time and effort, improve the post-editing experience and capture data for incremental MT/APE (automatic post-editing) and translation process research. The tool supports individual as well as batch mode file translation and provides translations from three engines – translation memory (TM), MT and APE. TM suggestions are color coded to accelerate the post-editing task. Users can integrate their personal TM/MT outputs. The tool remotely monitors and records post-editing activities generating an extensive range of post-editing logs. Compared with current state-of-the-art CAT tools, *CATaLog Online* provides an enhanced interface, an option to integrate APE and more informative logs to help translation process research.

A.1 Introduction

Machine translation (MT) technology has improved substantially over the past few decades. MT output is no longer used just for gisting but also for post-editing by professional translators as an important part of the translation workflow. Several studies confirm that post-editing MT output increases translators' productivity and improves translation consistency (Guerberof, 2009; Plitt and Masselot, 2010; Zampieri and Vela, 2014). Alongside classical TM matches, computer-aided translation (CAT) tools that integrate MT and TM output are a trend in the translation and localization industries providing translators more useful suggestions. Another important trend is the development of web-based CAT tools which require no local software installation and allow teams of translators to work on the same project simultaneously (e.g., WordFast Anywhere[1], MateCat[2] (Federico et al., 2014), and Wordbee[3], Lilt[4] etc.).

This paper presents *CATaLog Online*, a web-based CAT tool that provides translators MT, TM and APE output and ensures data capture for APE development and translation process research. The MT and APE systems integrated in *CATaLog Online* are based on Pal et al. (2015a) and Pal et al. (2016f), respectively. In this paper, we present the key features implemented in *CATaLog Online* and their importance to translation project managers, translators, and MT and APE developers. Compared to state-of-the-art CAT tools (e.g., MateCat, Lilt) *CATaLog Online* offers the following advantages: (i) color coded TM translation suggestions (highlighted TM source and corresponding target fragments are shown in the same interface), (ii) a wide range of editing logs, (iii) alignment between source, TM/MT/APE and the results of human PE, (iv) improved TM similarity measure and search technique (Pal et al., 2016e), and (v) additional translation option from APE which learns from human post-edited data.

A.2 CATaLog

CATaLog (Nayek et al., 2015) is a TM-based CAT tool which provides core functionalities for *CATaLog Online*. What distinguishes *CATaLog* from existing TM-based CAT tools

[1] https://www.freetm.com/
[2] https://www.matecat.com/
[3] http://www.wordbee.com/
[4] https://lilt.com/

is a set of newly introduced features targeted towards improving post-editing experience in terms of both performance and productivity. These include an improved TM similarity measure, searching and a novel coloring scheme. The color coding introduced into *CATaLog* guides the user during the translation (or post-editing) process. The matching parts in the TM source matches, as well as their translations in the target, are displayed in green, while the non-matching parts in both the TM source and target suggestions are displayed in red. Unaligned words are shown in orange. Similarly, when the user clicks on one of the 5 TM suggestions to start the post-editing task, the corresponding matching and non-matching parts in the input segment are also displayed in green and red, respectively. The color coding scheme not only helps the user to choose the most suitable TM suggestion for post-editing, it also helps the user to identify which parts of a TM match require more post-editing effort and which fragments are reliable translations. Key features of *CATaLog* are presented below.

A.2.1 Similarity Measure

For determining useful TM matches *CATaLog* employs the automatic MT evaluation metric - Translation Edit Rate (TER[5]) (Snover et al., 2006c). TER is intuitively a very useful similarity metric for use in TM as it directly mimics the human post-editing behavior and it shows a very high correlation with human evaluation. Moreover, TER provides the alignments between a segment pair that indicate which parts are common to the pair and which portions differ. TER is essentially an error metric: the lower the TER score, the higher the match. Unlike Simard and Fujita (2012) who first proposed and studied the use of different MT evaluation metrics as measure of similarity in TM, we do not use TER in its original definition: TER weighs each editing operation equally. However, the deletion operation takes much less time and effort compared to the other editing operations in post-editing. Therefore, we assign a lower cost to the delete operation compared to the other three edit operations. However, *CATaLog* allows users to set the editing costs according to their own preference.

The top 100 most relevant TM suggestions returned by the Lucene based search engine (cf. Section A.2.2) are re-ranked using the TER style *CATaLog* similarity score which is computed following Equation A.1, where n_m and s_m refer to the number of matches

[5]http://www.cs.umd.edu/ snover/tercom/

and match reward scores, respectively; e_i refers to four types of edit operations – *insert*, *delete*, *substitute* and *shift*; n_{e_i} and c_{e_i} refer to number of e_i edit operations required and the corresponding edit cost, respectively. Thus we reward matches and penalize edits to arrive at the final similarity score.

$$S = n_m \times s_m - \sum_{i=1}^{4} n_{e_i} \times c_{e_i} \tag{A.1}$$

A.2.2 Searching

To improve search efficiency, *CATaLog online* uses the standard information retrieval (IR) model of Lucene[6] with segment parsing, segment indexing, TF-IDF calculation, query parsing and finally searching/segment retrieval and segment ranking. Here query refers to the segment being translated and each indexed/retrieved segment contains (a) a TM source segment, (b) its corresponding translation and (c) the word alignments. All stop words are removed from the query (i.e. the input segment) before being presented to Lucene. Lucene retrieves the most relevant TM source candidates with respect to the query. The corresponding translations and the word alignments are also fetched. The set of relevant retrieved candidates is re-ranked according to their similarity scores. The ranking process also deals with a dissimilarity measurement (Kešelj et al., 2003) that provides a final score to re-rank the retrieved segments (Pal et al., 2016e).

A.2.3 Color Coding

A new color coding scheme has been introduced into *CATaLog* that guides the user during the translation (or post-editing) process. The matching parts in the TM source matches, as well as their translations in the target, are displayed in green, while the non-matching parts in both the TM source and target suggestions are displayed in red. Unaligned words are shown in orange. Similarly, when the user clicks on one of the 5 TM suggestions to start the post-editing task, the corresponding matching and non-matching parts in the input segment are also displayed in green and red, respectively. The color coding scheme not only helps the user to choose the most suitable TM suggestion for post-editing, it also

[6]http://lucene.apache.org/

helps the user to identify which parts of a TM match require more post-editing effort and which fragments are reliable translations.

A.3 CATaLog Online

CATaLog online provides a novel and user-friendly online CAT environment for post-editors and translators to reduce post-editing time and effort and improve the post-editing experience. The tool remotely monitors and records translator/post-editor activities generating a wide range of post-editing logs (cf. Section A.3.4.1) which are a fundamental source of information for APE and translation process research (cf. Section A.4). *CATaLog online*, on the one hand, produces multiple translation options for an uploaded input text file. On the other hand, it is a language independent tool that enables users to upload their own translation memories.

Figure A.1 shows the main user interface of the tool. On the main user interface[7], users can translate a single segment after choosing the source language and the target language (cf. "Quick Translation" in the main interface). The suggested translations are generated from three different engines: MT, TM and APE. The TM output is color coded. The user has to click the "translation suggestions" link after presenting input source text, choosing the language pair and pressing the "translate it!" button. Unlike other existing CAT tools, *CATaLog online* provides many facilities including file translation, CAT tool environment, user management, project management, translation data capture, etc.

A.3.1 File Translation

CATaLog online provides facilities for batch mode file translation, i.e., a user can input a source file (English in this case) as shown in Figure A.2. The *CATaLog online* file translation option provides a post-editing environment which allows the user to post-edit the selected translation from among the three translation suggestions (MT, TM and APE) in the target language (German in this case). The user has to choose the source–target language pair and upload a text file which contains a set of source segments. The tool first translates this text file at the back end by creating a project and then assigns a unique

[7] `http://santanu.appling.uni-saarland.de/CATaLog/`

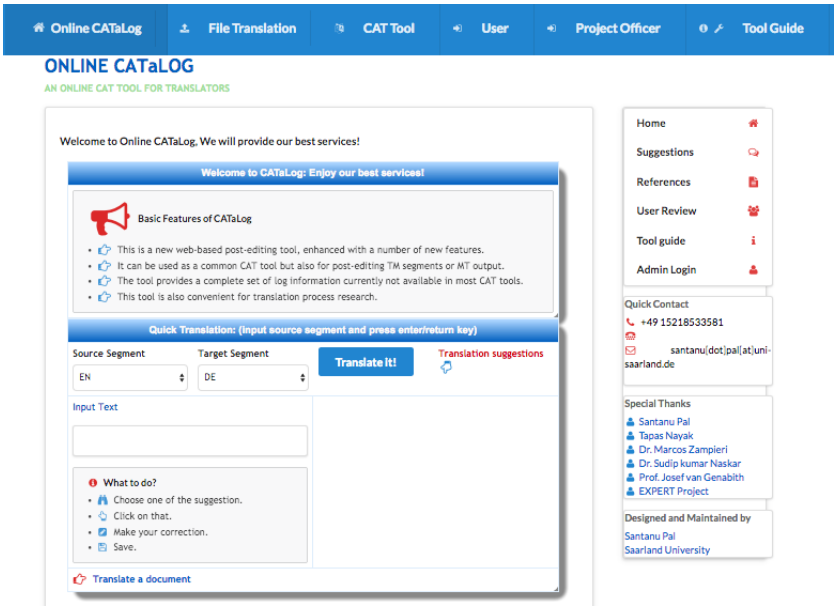


FIGURE A.1: Landing page user interface of *CATaLog online*

job identification number (Job ID) to the user which is displayed on the large red button in the interface (cf. Figure A.6). Each project/job is associated with a unique job URL. The user can either keep this Job ID for future reference or directly go to the job page by clicking on the recent Job ID (i.e., the red button marked with the Job ID). Whenever an user needs to recover his/her project/job, he/she has to simply remember the Job ID and search the project/job using that Job ID (cf. Figure A.6). The File translation interface provides on-the-fly user guidance regarding the "usage" and "tool functionality" in terms of message services.

A.3.2 CAT Tool

The CAT Tool interface (cf. Figure A.3) is very similar to the File Translation interface described in Section A.3.1, however, it differs a little in terms of features and functionalities. A key option facilitates the user to upload their own translation memories in a specific file format. The file format is a tab separated text file as given in Example 1.

Example A.1.

$SourceSegment < TAB > Translation1 < TAB > Translation2$

This option serves as a language independent feature for *CATaLog online*. The CAT Tool interface does not utilize the full functionalities offered by *CATaLog online*, e.g. it

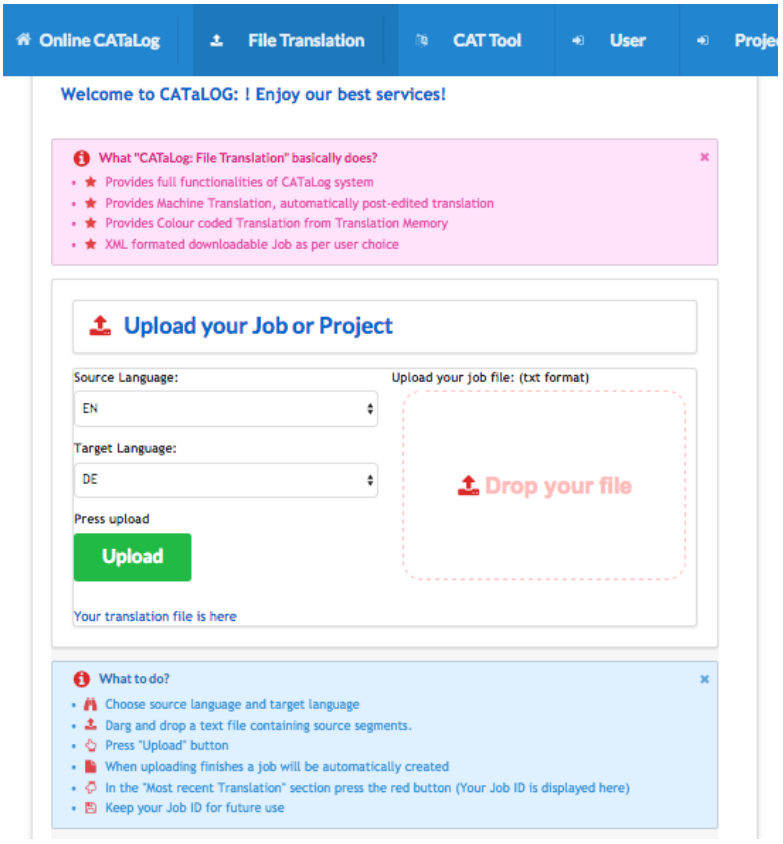


FIGURE A.2: File translation interface

does not use MT or APE translation generated by the tool (cf. Section A.3.1). Users have full freedom to use MT translations generated by their own MT system or third party MT engines (up to two alternatives are supported in the current version for each source segment). Additionally, *CATaLog online* provides color coded translations from the back end TM. When the uploading finishes, the system provides a unique Job ID; the functionality is similar to that described in Section A.3.1.

A.3.3 Project Management

Project managers (PM) initiate a translation project by understanding the project scope, based on input and requests from the client. Translation PMs then review and examine the source files to determine a variety of factors and finally decide how to accomplish the work by distributing the job among translators. The *CATaLog online* project management system is currently in the development stage. The released version supports basic project management activity described below.

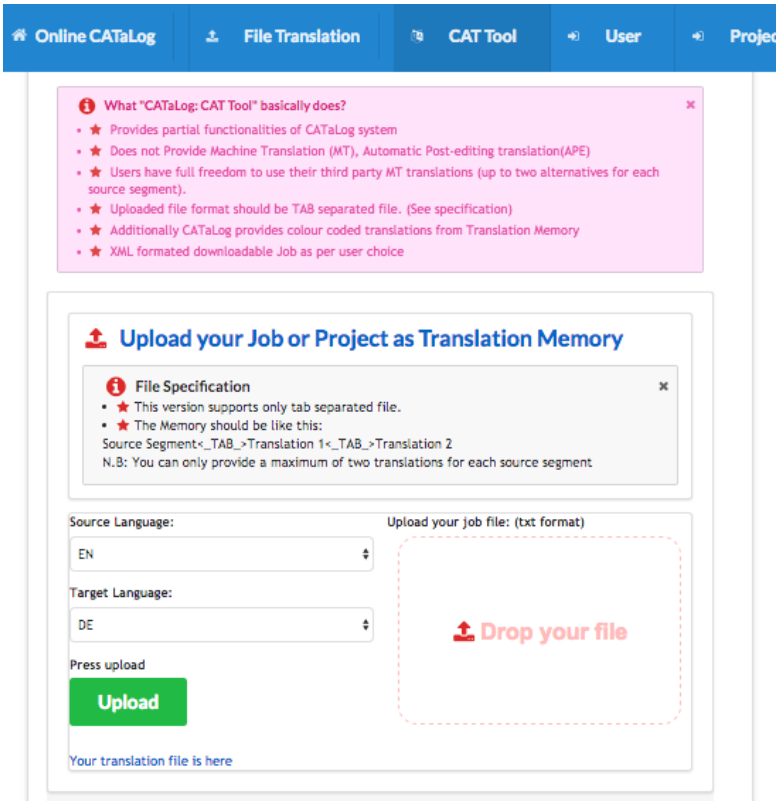


FIGURE A.3: CAT interface

A registered PM creates a translation project for a specific language pair by uploading a source file. Once a project/job has been created, a job Id appears in a row of the job assignment table. Some additional information is also associated with the job Id e.g., issue date, submission date, available translators for that particular language pair, etc. The PM can review the job and assign translation sub-jobs to any of the available translators by setting a submission deadline (cf. Figure A.4).

As soon as the PM assigns a job to a particular registered translator, the translator can see and review that job. The interface provides three options to the translator by which s/he can set the status of her/his activity for that particular job. A translator can either delete the assigned job from her/his profile by setting a "Deny" status or s/he can accept it by setting the "Accept" status (cf. Figure A.5). After finishing a translation task, the translator sets the corresponding job status as "Completed" which is directly updated in the PM's job status where s/he can see the completed and pending jobs. Finally, after reviewing, the PM can download the completed job (cf. Figure A.9) and deliver it to the client.

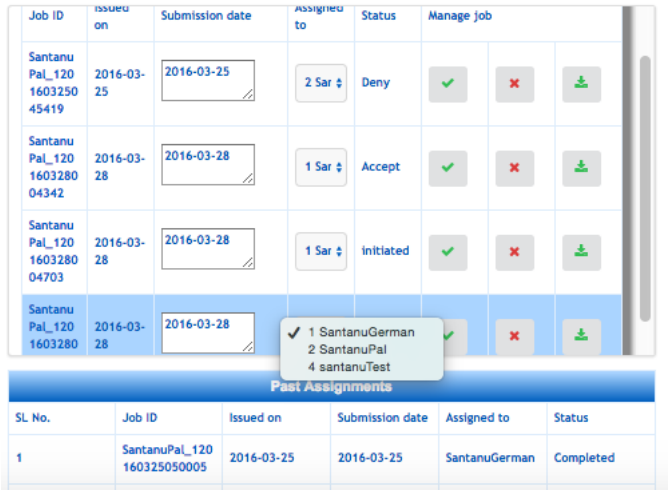


FIGURE A.4: Project Management interface for PMs

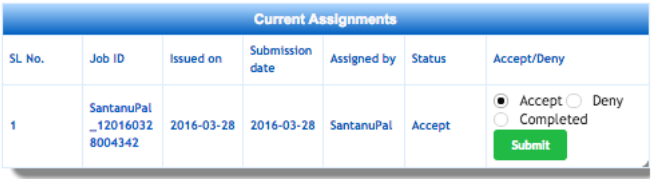


FIGURE A.5: Project Management interface for translators

A.3.4 Job Management

A job is created when the PM or a guest user uploads a source file. The job interface of *CATaLog online* provides three different translation alternatives for each source segment (cf. Figure A.7). One of the alternative translations which is fetched from TM is color coded (cf. Section A.2). The other two outputs are either from MT and APE engines provided by *CATaLog online* (cf. Section A.3.1) or the uploaded third party MT engine outputs (cf. Section A.3.2). As shown in Figure A.7, source segments are listed in the blue panel on the left and the corresponding translation suggestions appear on the right panel upon clicking a link shown above the source segment. The translator has to choose one of these suggestions and post-edit it. Figure A.8 shows the interface when the translator selects the TM suggestion. The final translation appears in the green panel on the left when the translator presses the "Save" button. The editing time (in seconds) is also shown below the final translation panel. After finishing each translation, an editing summary shows the number of editing operations performed by the translator. *CATaLog online* provides an on-the-fly editing guide for each source segment throughout the translation process. In case of re-editing a translation, the previously stored final translation shows

up as the first translation suggestion in the suggestion panel.

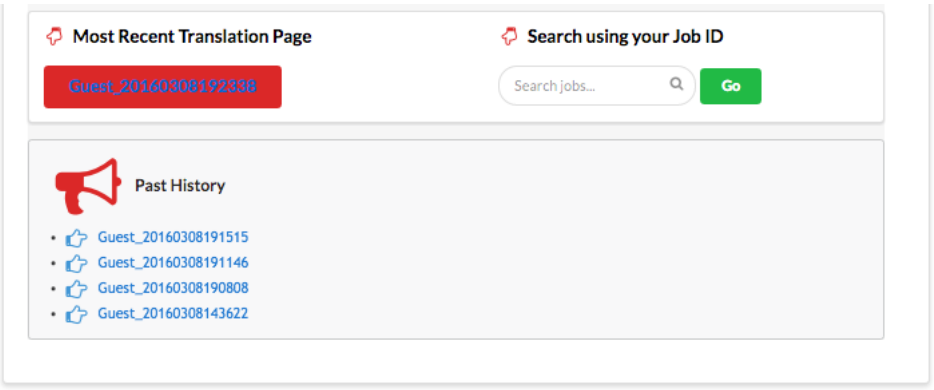


FIGURE A.6: Job search interface of *CATaLog online*

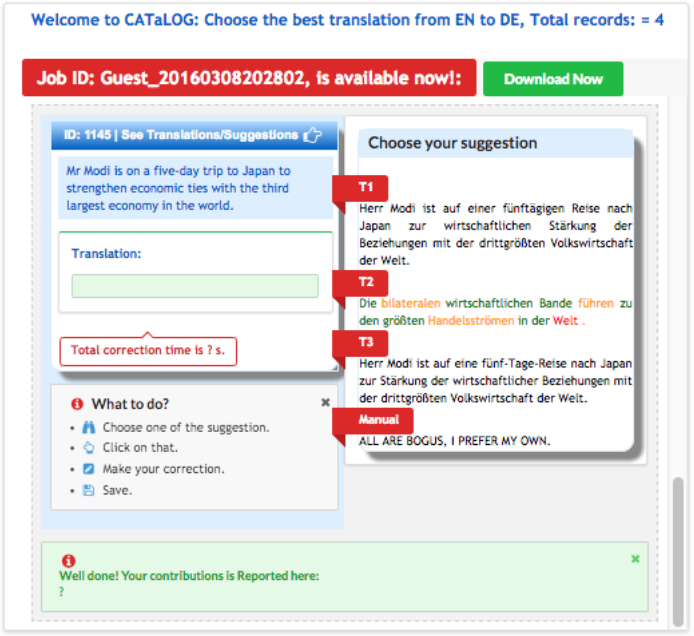


FIGURE A.7: Job interface of *CATaLog online*

A.3.4.1 Editing Log

For a given input segment, the user edits the best translation suggestion which may contain errors such as missing words, incorrect word order, wrong lexical choice, presence of irrelevant words, untranslated words or punctuation errors. The system records the user activities such as key strokes, cursor positions, text selection and mouse clicks. *CATaLog online* provides analytical summaries of post-editing activities during translation and presents well structured XML formatted logs. The XML formatted logs can be customized

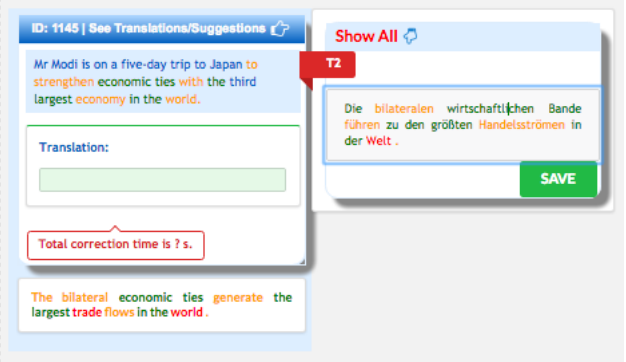


FIGURE A.8: Job interface of TM selection

according to the user's choice, e.g., the user can download entire logs or some specific logs for a particular translation job (cf. Figure A.9).

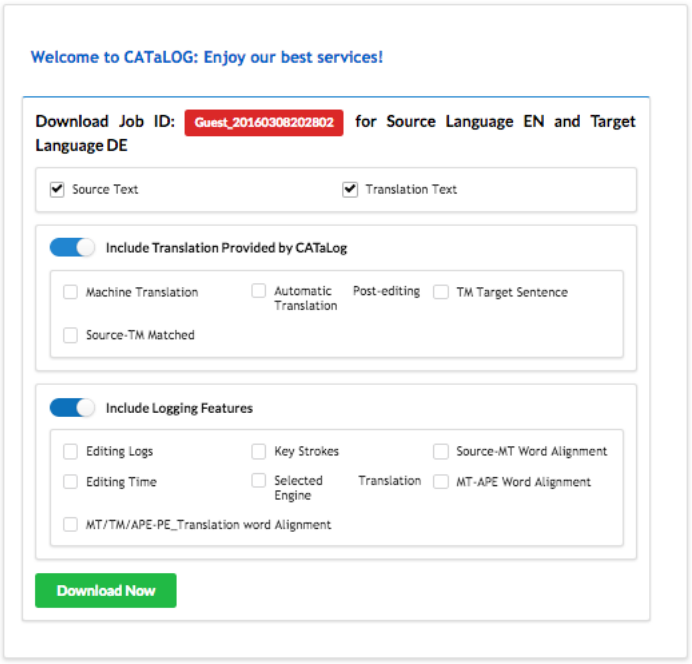


FIGURE A.9: Job download interface

A.4 APE and Translation Process Research using CATaLog Online

The post-editing logs collected during the translation process are a valuable source of information for translation process research as well as APE research and development. These user activity data logs not only help to assess the performance and understand the

behavior of the translators, they also provide crucial information about cognitive aspects of post-editing. The logs can be used to model APE to improve quality and productivity.

User Perspective: *CATaLog Online* generates a summary for every completed translation task which includes translator productivity in terms of number of words translated per minute and time taken per word. From the logs it is also possible to generate a report on translator style and behavior which can include, e.g., number of keystrokes per (effective) character editing, repetitive typing, preference for certain function words, etc.

Research Perspective: *CATaLog Online* records word alignments between source–MT, MT–APE, source–APE and source–HPE. These alignments and related post-editing information are beneficial for incremental MT/APE. Moreover, the source–HPE word alignments gathered by the tool can serve as a potential source for terminology extraction.

A.5 Conclusions and Future Work

CATaLog Online is a novel and user-friendly online CAT tool offering new features developed with the objective of improving translation productivity and experience. The tool provides a wide range of logs and data which serve as important information to translation process researchers, MT developers, and APE developers. The success of the two editions of the APE shared task in WMT (Bojar et al., 2016) indicate that APE is one of the important directions that research in MT is moving to. Post-editing tools, such as CATaLog Online, are able to provide crucial information for APE development. We would like to further expand and improve the tool by including additional features, e.g., interactive translation prediction in the form of on-the-fly translation suggestion, terminology extraction, option for compiling corpora, auto-suggestion for words, on-click pop-up terminology view, etc. Finally, we would like to model user behaviour and implement incremental MT/APE using the edit logs provided by the tool.

Bibliography

E. Agirre, C. Baneab, C. Cardiec, D. Cerd, M. Diabe, A. Gonzalez-Agirrea, W. Guof, R. Mihalcea, G. Rigau, and J. Wiebeg. Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of SemEval 2014*, page 81, 2014.

Alfred. V. Aho and Jeffrey. Ullman. Translations on a Context Free Grammar. In *Proceedings of the First Annual ACM Symposium on Theory of Computing*, pages 93–112, 1969.

Jacob Andreas, Nizar Habash, and Owen Rambow. Fuzzy syntactic reordering for phrase-based statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, WMT '11, pages 227–236, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-12-1. URL `http://dl.acm.org/citation.cfm?id=2132960.2132991`.

P. V. S. Avinesh. A data mining approach to learn reorder rules for smt. In *Proceedings of the NAACL HLT 2010 Student Research Workshop*, HLT-SRWS '10, pages 52–57, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1858146.1858156`.

Necip Fazil Ayan, Bonnie J. Dorr, and Christof Monz. NeurAlign: Combining Word Alignments Using Neural Networks. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 65–72, Vancouver, British Columbia, Canada, October 2005. Association for Computational Linguistics.

Ibrahim Badr, Rabih Zbib, and James Glass. Syntactic phrase reordering for english-to-arabic statistical machine translation. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '09, pages 86–93, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1609067.1609076`.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv preprint arXiv:1409.0473*, 2015.

Timothy Baldwin and Su Nam Kim. Multiword expressions. In *Handbook of Natural Language Processing, Second Edition*. CRC Press, Taylor and Francis Group, Boca Raton, FL, 2010.

Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, 2005.

S. Bangalore, G. Bordel, and G. Riccardi. Computing consensus translation from multiple machine translation systems. In *Proceedings of IEEE ASRU, Madonna di Campiglio, Italy*, 2002.

Hanna Béchara, Yanjun Ma, and Josef van Genabith. Statistical Post-Editing for a Statistical MT System. In *Proceedings of MT summit XIII*, pages 308–315, 2011.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning Long-Term Dependencies with Gradient Descent is Difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A Neural Probabilistic Language Model. *J. Mach. Learn. Res.*, 3:1137–1155, 2003.

Luisa Bentivogli, Raffaella Bernardi, Marco Marelli, Stefano Menini, Marco Baroni, and Roberto Zamparelli. Sick through the semeval glasses. lesson learned from the evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. *Lang. Resour. Eval.*, 50(1):95–124, March 2016.

Nicola Bertoldi, Mauro Cettolo, and Marcello Federico. Cache-based online adaptation for machine translation enhanced computer assisted translation. *Proceedings of the XIV Machine Translation Summit*, pages 35–42, 2013.

Peter J. Bickel and Kjell A. Doksum. *Mathematical statistics : basic ideas and selected topics*. Prentice-Hall, 1977.

Arianna Bisazza and Marcello Federico. Chunk-based verb reordering in vso sentences for arabic-english statistical machine translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, WMT '10, pages 235–243, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 978-1-932432-71-8. URL `http://dl.acm.org/citation.cfm?id=1868850.1868885`.

D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003a.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, March 2003b.

Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, et al. Findings of the 2016 Conference on Machine Translation (WMT16). In *Proceedings of WMT*, 2016.

Dhouha Bouamor, Nasredine Semmar, and Pierre Zweigenbaum. Automatic construction of a multiword expressions bilingual lexicon: A statistical machine translation evaluation perspective. In *Proceedings of the 3rd Workshop on Cognitive Aspects of the Lexicon (CogALex-III)COLING 2012*, pages 95—108, Mumbai, India, December 2012a.

Dhouha Bouamor, Nasredine Semmar, and Pierre Zweigenbaum. Identifying bilingual multiword expressions for statistical machine translation. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 674–679, Istanbul, Turkey, may 2012b.

Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational linguistics*, 19(2):263–311, 1993.

Michael Carl and Andy Way. *Recent advances in example-based machine translation*, volume 21. Springer Science & Business Media, 2003.

Marine Carpuat and Mona Diab. Task-based evaluation of multiword expressions: A pilot study in statistical machine translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 242–245, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. ISBN 1-932432-65-5. URL `http://dl.acm.org/citation.cfm?id=1857999.1858028`.

Marine Carpuat, Yuval Marton, and Nizar Habash. Improving arabic-to-english statistical machine translation by reordering post-verbal subjects for alignment. In *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort '10, pages 178–183, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1858842.1858875`.

Mauro Cettolo, Christophe Servan, Nicola Bertoldi, Marcello Federico, Loic Barrault, and Holger Schwenk. Issues in incremental adaptation of statistical MT from human post-edits. In *Proceedings of the Workshop on Post-editing Technology and Practice (WPTP-2)*, Nice, France, 2013.

Tanmoy Chakraborty, Santanu Pal, Tapabrata Mondal, Tanik Saikh, and Sivaji Bandyopadhyay. Shared task system description: Measuring the compositionality of bigrams using statistical methodologies. In *Proceedings of the Workshop on Distributional Semantics and Compositionality*, pages 38–42, 2011.

Tanmoy Chakraborty, Dipankar Das, and Sivaji Bandyopadhyay. Identifying bengali multiword expressions using semantic clustering. *International Journal of Linguistics and Language Resources (Lingvisticæ Investigationes)*, 2014.

A. J. Chaney and D. M. Blei. Visualizing topic models. In *International AAAI Conference on Social Media and Weblogs*, Department of Computer Science, Princeton University, Princeton NJ, USA, March 2012.

Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27, 2011.

Rajen Chatterjee, Marco Turchi, and Matteo Negri. The fbk participation in the wmt15 automatic post-editing shared task. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 210–215, Lisbon, Portugal, September 2015a. Association for Computational Linguistics. URL `http://aclweb.org/anthology/W15-3025`.

Rajen Chatterjee, Marion Weller, Matteo Negri, and Marco Turchi. Exploring the Planet of the APEs: a Comparative Study of State-of-the-art Methods for MT Automatic Post-Editing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 156–161, Beijing, China, July 2015b. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P15-2026`.

Boxing Chen, Min Zhang, Haizhou Li, and Aiti Aw. A comparative study of hypothesis alignment and its improvement for machine translation system combination. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 941–948, 2009.

J. Chen and J-Y. Nie. Parallel web text mining for cross-language IR. In *Proceedings of RIAO 2000: Content-Based Multimedia Information Access, Volume 1*, pages 62–78, Paris, 2000.

Wenhu Chen, Evgeny Matusov, Shahram Khadivi, and Jan-Thorsten Peter. Guided alignment training for topic-aware neural machine translation. *arXiv preprint arXiv:1607.01628*, 2016.

Wei-Chen Cheng, Stanley Kok, Hoai Vu Pham, Hai Leong Chieu, and Kian Ming Adam Chai. Language modeling with sum-product networks. In *INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14-18, 2014*, pages 2098–2102, 2014.

Colin Cherry and George Foster. Batch Tuning Strategies for Statistical Machine Translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436, 2012.

David Chiang. A Hierarchical Phrase-based Model for Statistical Machine Translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 263–270, 2005.

David Chiang. Hierarchical Phrase-Based Translation. *Computational Linguistics*, 33(2):201–228, 2007.

Yun-Chuang Chiao and Pierre Zweigenbaum. Looking for candidate translational equivalents in specialized, comparable corpora. In *Proceedings of COLING-2002, the 19th International Conference on Computational Linguistics, Volume 2*, pages 1–5. Association for Computational Linguistics, 2002.

KyungHyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *CoRR*, abs/1409.1259, 2014a. URL `http://arxiv.org/abs/1409.1259`.

Kyunghyun Cho, Bart Van Merriënboer, Çağlar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, 2014b. Association for Computational Linguistics.

Junyoung Chung, Çağlar Gülçehre, Kyunghyun Cho, and Yoshua Bengio. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. Technical Report Arxiv report 1412.3555, Université de Montréal, 2014.

Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. In *Proceedings of the 27th Annual Meeting on Association for Computational Linguistics*, ACL '89, pages 76–83, Stroudsburg, PA, USA, 1989. Association for Computational Linguistics.

I. Cicekli and H. A. Guvenir. Learning translation templates from bilingual translation examples. *Applied Intelligence*, 15(1):57–76, 2001.

Ilyas Cicekli and H Altay Güvenir. Learning Translation Templates From Bilingual Translation Examples. *Applied Intelligence*, 15(1):57–76, 2001.

J.P. Clark. System, method, and product for dynamically aligning translations in a translation-memory system, February 5 2002. URL `https://www.google.com/patents/US6345244`. US Patent 6,345,244.

Stephen Clark, Julia Hockenmaier, and Mark Steedman. Building deep dependency structures with a wide-coverage ccg parser. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 327–334, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

Jacob Cohen. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46, April 1960.

Michael Collins, Philipp Koehn, and Ivona Kučerová. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 531–540, 2005.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

I. Dagan and O. Glickman. Probabilistic textual entailment: Generic applied modeling of language variability. In *Proceedings of PASCAL Workshop on Learning Methods for Text Understanding and Mining*, page 6, Grenoble, 2004.

Han Dan, Katsuhito Sudoh, Xianchao Wu, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. Head finalization reordering for chinese-to-japanese machine translation. In *Proceedings of the Sixth Workshop on Syntax, Semantics and Structure in Statistical Translation*, SSST-6 '12, pages 57–66, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=2392936.2392946`.

Sandipan Dandapat, Sara Morrissey, Sudip Kumar Naskar, and Harold Somers. Mitigating problems in analogy-based ebmt with smt and vice versa: a case study with named entity transliteration. In *Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation, PACLIC 24*, pages 365–372, 2010.

Sandipan Dandapat, Sara Morrissey, Andy Way, and Mikel L Forcada. Using example-based mt to support statistical mt when translating homogeneous data in a resource-poor setting. *Proceedings of the 15th annual meeting of the European Association for Machine Translation (EAMT 2011)*, pages 201–208, 2011.

Dipankar Das, Santanu Pal, Tapabrata Mondal, Tanmoy Chakraborty, and Sivaji Bandyopadhyay. Automatic extraction of complex predicates in bengali. In *Proceedings of the workshop on Multiword expression: from theory to application (MWE 2010) (Coling 2010)*, pages 37–46, 2010.

Hal Daumé, III and Jagadeesh Jagarlamudi. Domain adaptation for machine translation by mining unseen words. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 407–412, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-932432-88-6. URL `http://dl.acm.org/citation.cfm?id=2002736.2002819`.

Marie-Catherine de Marneffe and Christopher D. Manning. The stanford typed dependencies representation. In *Coling 2008: Proceedings of the Workshop on Cross-Framework and Cross-Domain Parser Evaluation*, CrossParser '08, pages 1–8, 2008.

Donald A. De Palma and Nataly Kelly. Project Management for Crowdsourced Translation: How User-Translated Content Projects Work in Real Life. *Translation and Localization Project Management: The Art of the Possible*, pages 379–408, 2009.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 41(6):391–407, 1990.

John DeNero and Klaus Macherey. Model-based Aligner Combination Using Dual Decomposition. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 420–429, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

Michael Denkowski. *Machine Translation for Human Translators*. PhD thesis, Carnegie Mellon University, 2015.

Michael Denkowski and Alon Lavie. Meteor 1.3: Automatic Metric for Reliable Optimization and Evaluation of Machine Translation Systems. In *Proceedings of the EMNLP 2011 Workshop on Statistical Machine Translation*, 2011.

Jacob Devlin and Spyros Matsoukas. Trait-based hypothesis selection for machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 528–532, 2012.

Yuan Ding and Martha Palmer. Machine Translation Using Probabilistic Synchronous Dependency Insertion Grammars. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 541–548, 2005.

George Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, pages 138–145. Morgan Kaufmann Publishers Inc, 2002.

Bonnie J. Dorr, Eduard H. Hovy, and Lori S. Levin. Machine translation: Interlingual methods, 2006.

Jinhua Du, Yifan He, Sergio Penkale, and Andy Way. MATREX: The DCU MT System for WMT 2009. In *Proceedings of the 4th Workshop on Statistical Machine Translation*, pages 95–99, March 2009. URL `http://www.aclweb.org/anthology/W/W09/W09-0416`.

Jinhua Du, Pavel Pecina, and Andy Way. An augmented three-pass system combination framework: Dcu combination system for wmt 2010. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 296–301, Uppsala, Sweden, July 2010.

Ted Dunning. Accurate methods for the statistics of surprise and coincidence. *Comput. Linguist.*, 19(1):61–74, March 1993. ISSN 0891-2017.

Nadir Durrani, Helmut Schmid, and Alexander Fraser. A joint sequence translation model with integrated reordering. In *Proceedings of ACL*, 2011.

Nadir Durrani, Helmut Schmid, Alexander M. Fraser, Philipp Koehn, and Hinrich Schütze. The operation sequence model - combining n-gram-based and phrase-based statistical machine translation. *Computational Linguistics*, 41:185–214, 2015.

Andreas Eisele and Jia Xu. Improving machine translation performance using comparable corpora. In *Proceedings of the 3rd Workshop on Building and Using Comparable Corpora. Applications of Parallel and Comparable Corpora in Natural Language Engineering and the Humanities*, pages 35–39, 2010.

Asif Ekbal and Sivaji Bandyopadhyay. Named entity recognition using support vector machine: A language independent approach. *International Journal of Electrical, Computer, and Systems Engineering*, 4(2):155–170, 2010.

Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1535–1545, 2011.

Federico Fancellu and Bonnie L. Webber. Translating negation: Induction, search and model errors. In *Proceedings of the Ninth Workshop on Syntax, Semantics and Structure in Statistical Translation, SSST@NAACL-HLT 2015, Denver, Colorado, USA, 4 June 2015*, pages 21–29, 2015.

Marcello Federico, Alessandro Cattelan, and Marco Trombetti. Measuring User Productivity in Machine Translation Enhanced Computer Assisted Translation. In *Proceedings of AMTA*, 2012.

Marcello Federico, Nicola Bertoldi, Mauro Cettolo, Matteo Negri, Marco Turchi, Marco Trombetti, Alessandro Cattelan, Antonio Farina, Domenico Lupinetti, Andrea Martines, et al. The matecat tool. In *Proceedings of COLING*, pages 129–132, 2014.

Yang Feng, Yang Liu, Haitao Mi, Qun Liu, and Yajuan Lü. Lattice-based system combination for statistical machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1105–1113, Singapore, August 2009.

Rebecca Fiederer and Sharon O'Brien. Quality and Machine Translation: a Realistic Objective. *Journal of Specialised Translation*, 11:52–74, 2009.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

George Foster, Roland Kuhn, and Howard Johnson. Phrasetable Smoothing for Statistical Machine Translation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 53–61, 2006.

P. Fung and P. Cheung. Mining Verynon-parallel corpora: Parallel sentence and lexicon extraction via bootstrapping and EM. In *Proceedings of EMNLP*, pages 57–63, 2004.

Pascale Fung and Kathleen McKeown. Finding terminology translations from non-parallel corpora. In *Proceedings of the 5th Annual Workshop on Very Large Corpora*, pages 192–202, 1997.

Pascale Fung and Lo Yuen Yee. An ir approach for translating new words from nonparallel, comparable texts. In *Proceedings of COLING-98, the 17th International Conference on Computational Linguistics, Volume 1*, pages 414–420. Association for Computational Linguistics, 1998.

Michel Galley and Christopher D. Manning. A Simple and Effective Hierarchical Phrase Reordering Model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 848–856, 2008.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. What's in a translation rule? In *HLT-NAACL 2004: Main Proceedings*, pages 273–280, Boston, Massachusetts, USA, May 2 - May 7 2004. Association for Computational Linguistics.

P. Gamallo Otero. Learning bilingual lexicons from comparable English and Spanish corpora. In *Proceedings of MT Summit XI*, pages 191–198, 2007.

Mercedes García-Martínez, Loïc Barrault, and Fethi Bougares. Factored neural machine translation. *CoRR*, abs/1609.04621, 2016. URL `http://arxiv.org/abs/1609.04621`.

Dmitriy Genzel. Automatically learning source-side reordering rules for large scale machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 376–384, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1873781.1873824`.

D. Giampiccolo, B. Magnini, I. Dagan, and B. Dolan. The Third PASCAL Recognizing Textual Entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, Prague, Czech Republic, 2007.

Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. *arXiv preprint arXiv:1302.4389*, 2013.

Fabrizio Gotti, Philippe Langlais, Elliott Macklovitch, Benoit Robichaud Didier Bourigault, and Claude Coulombe. 3GTM: A Third-Generation Translation Memory. In *Proceedings of the CLiNE Workshop*, 2005.

Edward Grefenstette and Mehrnoosh Sadrzadeh. Experimental support for a categorical compositional distributional model of meaning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1394–1404, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics. ISBN 978-1-937284-11-4. URL `http://dl.acm.org/citation.cfm?id=2145432.2145580`.

G. Grefenstette. *Cross-Language Information Retrieval*. Kluwer Academic Publishers, London, 1998.

Ana Guerberof. Productivity and quality in the post-editing of outputs from translation memories and machine translation. *Localisation Focus*, 7(1):133–140, 2009.

Ana Guerberof. *Productivity and Quality in the Post-Edition of Outputs from Translation Memories and Machine Translation*. PhD thesis, Rovira and Virgili University Tarragona, 2012.

Çaglar Gülçehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loïc Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. On using monolingual corpora in neural machine translation. *CoRR*, abs/1503.03535, 2015. URL `http://arxiv.org/abs/1503.03535`.

Deepa Gupta, Mauro Cettolo, and Marcello Federico. Pos-based reordering models for statistical machine translation. In *Proceedings of the MT Summit XI*, pages 207–213, 2007.

Rajdeep Gupta, Santanu Pal, and Sivaji Bandyopadhyay. Improving mt system using extracted parallel fragments of text from comparable corpora. In *Proceedings of 6th workshop of Building and Using Comparable Corpora (BUCC)*, pages 69–76, Sofia, Bulgaria, 2013. ACL.

Rohit Gupta and Constantin Orăsan. Incorporating Paraphrasing in Translation Memory Matching and Retrieval. In *Proceedings of EAMT*, 2014.

Rohit Gupta, Constantin Orasan, and Josef van Genabith. Reval: A simple and effective machine translation evaluation metric based on recurrent neural networks. In *EMNLP*, pages 1066–1072, 2015a.

Rohit Gupta, Constantin Orăsan, Marcos Zampieri, Mihaela Vela, and Josef van Genabith. Can Translation Memories afford not to use paraphrasing? In *Proceedings of EAMT*, 2015b.

Nizar Habash. Syntactic preprocessing for statistical machine translation. In *Proceedings of the MT Summit XI*, pages 215–222, 2007.

Xiaodong He and Kristina Toutanova. Joint optimization for machine translation system combination. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1202–1211, Singapore, August 2009. Association for Computational Linguistics.

Xiaodong He, Mei Yang, Jianfeng Gao, Patrick Nguyen, and Robert Moore. Indirect-HMM-based hypothesis alignment for combining outputs from machine translation systems. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 98–107, Honolulu, Hawaii, October 2008. Association for Computational Linguistics.

Yifan He, Yanjun Ma, Josef van Genabith, and Andy Way. Bridging SMT and TM with translation recommendation. In *Proceedings of ACL*, pages 622–630, 2010.

Kenneth Heafield. KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, 2011.

Almut Silja Hildebrand and Stephan Vogel. Cmu system combination via hypothesis selection for wmt'10. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, pages 307–310, Uppsala, Sweden, July 2010.

Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Comput.*, 9(8): 1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL `http://dx.doi.org/10.1162/neco.1997.9.8.1735`.

Maria Holmqvist, Sara Stymne, Lars Ahrenberg, and Magnus Merkel. Alignment-based reordering for smt. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7.

Liang Huang. Statistical syntax-directed translation with extended domain of locality. In *In Proc. AMTA 2006*, pages 66–73, 2006.

W. John Hutchins. Machine translation: A brief history. In *Concise history of the language sciences: from the Sumerians to the cognitivists, Pergamon*, pages 431–445. Oxford: Pergamon Press, 1995.

Ray Jackendoff. *The architecture of the language faculty.* Number 28. MIT Press, 1997.

Shyamsundar Jayaraman and Alon Lavie. Multi-engine machine translation guided by explicit word matching. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 101–104, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

Rafal Józefowicz, Wojciech Zaremba, and Ilya Sutskever. An Empirical Exploration of Recurrent Network Architectures. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2342–2350, 2015.

Marcin Junczys-Dowmunt and Arkadiusz Szał. SyMGiza++: Symmetrized Word Alignment Models for Statistical Machine Translation. In *Proceedings of the 2011 International Conference on Security and Intelligent Information Systems*, pages 379–390, 2012.

Nal Kalchbrenner and Phil Blunsom. Recurrent Continuous Translation Models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, 2013.

Panagiotis Kanavos and Dimitrios Kartsaklis. Integrating Machine Translation with Translation Memory: A Practical Approach. In *Proceedings of the Second Joint EM+/CNGL Workshop Bringing MT to the User: Research on Integrating MT in the Translation Industry*, 2010.

Damianos Karakos, Jason Eisner, Sanjeev Khudanpur, and Markus Dreyer. Machine translation system combination using itg-based alignments. In *Proceedings of ACL-08: HLT, Short Papers*, pages 81–84, Columbus, Ohio, June 2008. Association for Computational Linguistics.

Martin Kay. The proper place of men and machines in language translation. *Machine Translation*, 12(1-2):3–23, 1997.

Vlado Kešelj, Fuchun Peng, Nick Cercone, and Calvin Thomas. N-gram-based author profiles for authorship attribution. *Proceedings of the conference pacific association for computational linguistics, PACLING*, 3:255–264, 2003.

Alexandre Klementiev, Ann Irvine, Chris Callison-Burch, and David Yarowsky. Toward statistical machine translation without parallel corpora. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 130–140, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

Reinhard Kneser and Hermann Ney. Improved backing-off for n-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, volume I*, pages 181–184, 1995.

Kevin Knight. Decoding complexity in word-replacement translation models. *Computational Linguistics*, 25:607–615, December 1999.

Kevin Knight and Ishwar Chander. Automated Postediting of Documents. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (Vol. 1)*, pages 779–784, 1994.

Rebecca Knowles and Philipp Koehn. *Neural Interactive Translation Prediction*, pages 107–120. 2016.

Philipp Koehn. A Process Study of Computer-aided Translation. *Machine Translation*, 23(4): 241–263, 2009.

Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, New York, NY, USA, 1st edition, 2010.

Philipp Koehn and Hieu Hoang. Factored translation models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 868–876, 2007. URL `http://www.aclweb.org/anthology/D/D07/D07-1091`.

Philipp Koehn and Jean Senellart. Convergence of translation memory and statistical machine translation. In *Proceedings of AMTA Workshop on MT Research and the Translation Industry*, pages 21–31, 2010.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical Phrase-based Translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54, 2003.

Philipp Koehn, Amittai Axelrod, Ra Birch Mayne, Chris Callison-burch, Miles Osborne, and David Talbot. Edinburgh system description for the 2005 iwslt speech translation evaluation. In *In Proc. International Workshop on Spoken Language Translation (IWSLT*, 2005.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180, 2007.

Maarit Koponen. Is machine translation post-editing worth the effort? a survey of research into post-editing and effort. *Journal of Specialised Translation*, 25, 2016.

Shankar Kumar and William Byrne. Minimum Bayes Risk Decoding for Statistical Machine Translation. In *Proceedings of the North American Association for Computational Linguistics (NAACL)*, pages 169–176, March 2004.

Elina Lagoudaki. The value of machine translation for the professional translator. In *Proceedings of AMTA*, pages 262–269, Waikiki, Hawaii, 2008.

Patrik Lambert and Rafael E. Banchs. Data Inferred Multi-word Expressions for Statistical Machine Translation. In *Proceedings of Machine Translation Summit X*, pages 396–403, Phuket, Thailand, September 2005.

J. Richard Landis and Gary G. Koch. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1):159–74, 1977.

Alon Lavie and Abhaya Agarwal. METEOR: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, 2007.

Yves Lepage and Etienne Denoual. Purest ever example-based machine translation: Detailed presentation and assessment. *Machine Translation*, 19:251–282, 2005.

Gregor Leusch and Hermann Ney. The rwth system combination system for wmt 2010. In *ACL 2010 Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, pages 315–320, Uppsala, Sweden, 2010.

Abby Levenberg, Chris Callison-Burch, and Miles Osborne. Stream-based Translation Models for Statistical Machine Translation. In *Proceedings of Human Language Technologies*, pages 394–402, Stroudsburg, PA, USA, 2010.

Percy Liang, Ben Taskar, and Dan Klein. Alignment by Agreement. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 104–111, 2006.

Yang Liu, Qun Liu, and Shouxun Lin. Tree-to-string Alignment Template for Statistical Machine Translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 609–616, 2006.

Yang Liu, Tian Xia, Xinyan Xiao, and Qun Liu. Weighted alignment matrices for statistical machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1017–1026, Singapore, August 2009. Association for Computational Linguistics.

Anne-Marie Loffler-Laurian. Traduction automatique et style. *Babel*, 31(2):70–76, 1985.

Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September 2015a. Association for Computational Linguistics. URL `http://aclweb.org/anthology/D15-1166`.

Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 11–19, Beijing, China, July 2015b. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P15-1002`.

Wei-Yun Ma and Kathleen McKeown. System combination for machine translation through paraphrasing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1053–1058, 2015.

Elliott Macklovitch. Transtype2: The last word. In *In Proceedings of the 5th International Conference on Languages Resources and Evaluation (LREC 06*, pages 167–172, 2006.

Daniel Marcu. Towards a unified approach to memory and statistical-based machine translation. In *In Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL 2001)*, pages 386–393, Toulouse, France, 2001.

Daniel Marcu, Wei Wang, Abdessamad Echihabi, and Kevin Knight. SPMT: Statistical Machine Translation with Syntactified Target Language Phrases. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 44–52, Sydney, Australia, July 2006.

David Mareček, Rudolf Rosa, Petra Galuščáková, and Ondřej Bojar. Two-step Translation with Grammatical Post-processing. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 426–432, 2011.

Evgeny Matusov, Nicola Ueffing, and Hermann Ney. Computing Consensus Translation from Multiple Machine Translation Systems Using Enhanced Hypotheses Alignment. In *Conference of the European Chapter of the Association for Computational Linguistics*, pages 33–40, Trento, Italy, April 2006.

Haitao Mi and Liang Huang. Forest-based Translation Rule Extraction. In *Proceedings of EMNLP*, pages 206–214. ACL, 2008.

Haitao Mi, Liang Huang, and Qun Liu. Forest-Based Translation. In *Proceedings of ACL-08: HLT*, pages 192–199, Columbus, Ohio, June 2008. Association for Computational Linguistics.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K Weinberger, editors, *Advances in Neural Information Processing Systems, 26*, pages 3111–3119. Curran Associates, Inc, 2013a.

Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In *INTERSPEECH*, pages 1045–1048. ISCA, 2010.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013b.

George A. Miller. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11): 39–41, November 1995. URL `http://doi.acm.org/10.1145/219717.219748`.

Jeff Mitchell and Mirella Lapata. Composition in distributional models of semantics. *Cognitive Science*, 34(8):1388–1429, 2010.

D. S. Munteanu and D. Marcu. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31:477–504, 2005.

Makoto Nagao. A framework of a mechanical translation between japanese and english by analogy principle. In *Proc. Of the International NATO Symposium on Artificial and Human Intelligence*, pages 173–180, 1984.

Tapas Nayak, Santanu Pal, Naskar Sudip, and Josef van Genabith. Beyond translation memories: Generating translation suggestions based on parsing and pos tagging. In *2nd Workshop on Natural Language Processing for Translation Memories (NLP4TM 2016)*, pages 12–20, 2016.

Tapas Nayek, Sudip Kumar Naskar, Santanu Pal, Marcos Zampieri, Mihaela Vela, and Josef van Genabith. CATaLog: New Approaches to TM and Post Editing Interfaces. In *Proceedings of the 1th Workshop on Natural Language Processing for Translation Memories (NLP4TM) collocated with RANLP 2015*, pages 36–42, Hissar, Bulgaria, September 2015. RANLP 2015 Organising Committee. fr.

Saul Ben Needleman and Christian Dennis Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3): 443–453, March 1970.

Laurent Nepveu, Guy Lapalme, Philippe Langlais, and George F. Foster. Adaptive language and translation models for interactive machine translation. In *EMNLP*, pages 190–197. ACL, 2004.

Graham Neubig. Travatar: A Forest-to-String Machine Translation Engine based on Tree Transducers. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 91–96, Sofia, Bulgaria, 2013. Association for Computational Linguistics.

Jan Niehues and Muntsin Kolss. A pos-based model for long-range reorderings in smt. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, StatMT '09, pages 206–214, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. URL `http://dl.acm.org/citation.cfm?id=1626431.1626472`.

Sharon O'Brien. Eye-tracking and translation memory matches. *Perspectives: Studies in Translatology*, 14:185–204, 2006.

Sharon O'Brien, Johann Roturier, and Roberto De Almeida. Researching and Teaching Post-Editing. *In Post-Editing MT Output - Views from the researcher, trainer, practitioner*, 2009. URL `http://mt-archive.info/MTS-2009-OBrien-ppt.pdf`.

F. J. Och. *Statistical Machine Translation: From Single-Word Models to Alignment Templates*. PhD thesis, RWTH Aachen University, Computer Science Department, RWTH Aachen University, Aachen, Germany, October 2002.

Franz Josef Och. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, pages 160–167, 2003.

Franz Josef Och and Hermann Ney. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51, 2003a.

Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51, 2003b.

Franz Josef Och, Christoph Tillmann, and Hermann Ney. Improved alignment models for statistical machine translation. In *University of Maryland, College Park, MD*, pages 20–28, 1999.

Kemal Oflazer and Ilknur Durgar El-Kahlout. Exploring Different Representational Units in English-to-Turkish Statistical Machine Translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 25–32, Prague, Czech Republic, June 2007.

Ahmed Hamza Osman, Naomie Salim, Mohammed Salem Binwahlan, Rihab Alteeb, and Albaraa Abuobieda. An improved plagiarism detection scheme based on semantic role labeling. *Appl. Soft Comput.*, 12(5):1493–1502, 2012.

P. Pakray, S. Pal, S. Bandyopadhyay, and A. Gelbukh. Automatic answer validation system on english language. In *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE)*, volume 6, pages 329–333, 2010a.

Partha Pakray. *Answer Validation through Textual Entailment*. PhD thesis, Jadavpur University, 2013.

Partha Pakray, Santanu Pal, Soujanya Poria, Sivaji Bandyopadhyay, and Alexander Gelbukh. Ju_cse_tac: Textual entailment recognition system at tac rte-6. In *System Report, Text Analysis Conference Recognizing Textual Entailment Track (TAC RTE) Notebook*, 2010b.

Santanu Pal. Statistical Automatic Post Editing. In *The Proceedings of the EXPERT Scientific and Technological workshop*, 2015.

Santanu Pal, Sudip Kumar Naskar, Pavel Pecina, Sivaji Bandyopadhyay, and Andy Way. Handling Named Entities and Compound Verbs in Phrase-Based Statistical Machine Translation. In *In Proceedings of the of Multiword Expression Workshop (MWE-2010)*. The 23rd International conference of computational linguistics (Coling 2010), 2010.

Santanu Pal, Tanmoy Chakraborty, and Sivaji Bandyopadhyay. Handling Multiword Expressions in Phrase-Based Statistical Machine Translation. *Machine Translation Summit XIII*, pages 215–224, 2011.

Santanu Pal, Sudip Kumar Naskar, and Sivaji Bandyopadhyay. A Hybrid Word Alignment Model for Phrase-Based Statistical Ma-chine Translation. *ACL 2013*, pages 94–101, 2013a.

Santanu Pal, Sudip Kumar Naskar, and Sivaji Bandyopadhyay. MWE Alignment in Phrase Based Statistical Machine Translation. In *Proceedings of the XIV Machine Translation Summit*, pages 61–68, 2013b.

Santanu Pal, Sudip Kumar Naskar, and Sivaji Bandyopadhyay. Word Alignment-Based Reordering of Source Chunks in PB-SMT. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 3565–3571. European Language Resources Association (ELRA), may 2014a.

Santanu Pal, Partha Pakray, and Sudip Kumar Naskar. Automatic building and using parallel resources for SMT from comparable corpora. In *Proceedings of the 3rd Workshop on Hybrid Approaches to Translation (HyTra) @ EACL 2014*, pages 47–56, Gothenburg, Sweden, April 27 2014b. Association for Computational Linguistics.

Santanu Pal, Ankit Srivastava, Sandipan Dandapat, Josef van Genabith, Qun Liu, and Andy Way. USAAR-DCU Hybrid Machine Translation System for ICON 2014. In *Proceedings of the 11th International Conference on Natural Language Processing*, Goa, India, 2014c.

Santanu Pal, Sudip Naskar, and Josef van Genabith. UdS-Sant: English–German Hybrid Machine Translation System. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 216–221, Lisbon, Portugal, September 2015a. Association for Computational Linguistics.

Santanu Pal, Partha Pakray, Alexander Gelbukh, and Josef van Genabith. *Mining Parallel Resources for Machine Translation from Comparable Corpora*, pages 534–544. Springer International Publishing, 2015b.

Santanu Pal, Mihaela Vela, Sudip Kumar Naskar, and Josef van Genabith. USAAR-SAPE: An English–Spanish Statistical Automatic Post-Editing System. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 216–221, Lisbon, Portugal, September 2015c. Association for Computational Linguistics.

Santanu Pal, Sudip Kumar Naskar, and Josef van Genabith. Forest to string based statistical machine translation with hybrid word alignments. In *17th International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 2016a.

Santanu Pal, Sudip Kumar Naskar, and Josef van Genabith. Multi-engine and multi-alignment based automatic post-editing and its impact on translation productivity. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2559–2570, Osaka, Japan, December 2016b. The COLING 2016 Organizing Committee.

Santanu Pal, Sudip Kumar Naskar, Mihaela Vela, and Josef van Genabith. A Neural Network based Approach to Automatic Post-Editing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 281–286, Berlin, Germany, 2016c. URL `http://anthology.aclweb.org/P16-2046`.

Santanu Pal, Sudip Kumar Naskar, Marcos Zampieri, Tapas Nayak, and Josef van Genabith. Catalog online: A web-based cat tool for distributed translation with data capture for ape and translation process research. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 98–102, Osaka, Japan, December 2016d. The COLING 2016 Organizing Committee.

Santanu Pal, Marcos Zampieri, Sudip Kumar Naskar, Tapas Nayak, Mihaela Vela, and Josef van Genabith. Catalog online: Porting a post-editing tool to the web. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 599–604, may 2016e.

Santanu Pal, Marcos Zampieri, and Josef van Genabith. Usaar: An operation sequential model for automatic statistical post-editing. In *Proceedings of the First Conference on Machine Translation*, pages 759–763, Berlin, Germany, August 2016f. Association for Computational Linguistics.

Santanu Pal, Sudip Kumar Naskar, Mihaela Vela, Qun Liu, and Josef van Genabith. Neural automatic post-editing using prior alignment and reranking. *EACL 2017*, page 349, 2017.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, 2002.

Carla Parra Escartín and Manuel Arcedillo. Machine translation evaluation made fuzzier: A study on post-editing productivity and evaluation metrics in commercial settings. In *Proceedings of the MT Summit XV*, Miami (Florida), October 2015a. International Association for Machine Translation (IAMT).

Carla Parra Escartín and Manuel Arcedillo. Living on the edge: productivity gain thresholds in machine translation evaluation metrics. In *Proceedings of the Fourth Workshop on Post-editing Technology and Practice*, pages 46–56, Miami, Florida (USA), November 2015b. Association for Machine Translation in the Americas (AMTA).

Soma Paul. Representing compound verbs in indo wordnet, 2010.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL `http://www.aclweb.org/anthology/D14-1162`.

Álvaro Peris, Luis Cebrián, and Francisco Casacuberta. Online learning for neural machine translation post-editing. *CoRR*, abs/1706.03196, 2017. URL `http://arxiv.org/abs/1706.03196`.

Ari Pirkola, Turid Hedlund, Heikki Keskustalo, and Kalervo Järvelin. Dictionary-based cross-language information retrieval: Problems, methods, and research findings. *Information retrieval*, 4(3-4):209–230, 2001.

Mirko Plitt and François Masselot. A productivity test of statistical machine translation post-editing in a typical localisation context. *The Prague Bulletin of Mathematical Linguistics*, 93: 7–16, 2010.

Maja Popović and Hermann Ney. Pos-based reorderings for statistical machine translation. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*, pages 1278–1283, 2006.

Chris Quirk, Arul Menezes, and Colin Cherry. Dependency treelet translation: Syntactically informed phrasal SMT. In *n Proceedings of ACL*, pages 271–279, 2005.

Chris Quirk, Raghavendra Udupa U, and Arul Menezes. Generative models of noisy translations with applications to parallel fragment extraction. In *In Proceedings of MT Summit XI, European Association for Machine Translation*, 2007.

Sumathi Ramaswamy. Engendering language: the poetics of tamil identity. *Comparative Studies in Society and History*, 35(4), Oct 1993.

Reinhard Rapp. Identifying word translations in non-parallel texts. In *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics*, ACL '95, pages 320–322, Stroudsburg, PA, USA, 1995. Association for Computational Linguistics.

Reinhard Rapp. Automatic identification of word translations from unrelated English and German corpora. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*, pages 519–526. Association for Computational Linguistics, 1999.

R. Rehurek and P. Sojka. Software framework for topic modelling with large corpora. In *Proceedings of LREC 2010 workshop New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA.

Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, 2010.

Zhixiang Ren, Yajuan Lü, Jie Cao, Qun Liu, and Yun Huang. Improving statistical machine translation using domain bilingual multiword expressions. In *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications*, MWE '09, pages 47–54, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-60-2. URL `http://dl.acm.org/citation.cfm?id=1698239.1698249`.

Rudolf Rosa, David Mareček, and Ondřej Dušek. DEPFIX: A System for Automatic Correction of Czech MT Outputs. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, Stroudsburg, PA, USA, 2012.

Antti-Veikko Rosti, Necip Fazil Ayan, Bing Xiang, Spyros Matsoukas, Richard Schwartz, and Bonnie J. Dorr. Combining outputs from multiple machine translation systems. In *In Proceedings of the North American Chapter of the Association for Computational Linguistics Human Language Technologies*, pages 228–235, 2007a.

Antti-Veikko Rosti, Spyros Matsoukas, and Richard Schwartz. Improved word-level system combination for machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 312–319, June 2007b.

Antti-Veikko Rosti, Xiaodong He, Damianos Karakos, Gregor Leusch, Yuan Cao, Markus Freitag, Spyros Matsoukas, Hermann Ney, Jason Smith, and Bing Zhang. Review of hypothesis alignment algorithms for mt system combination via confusion network decoding. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 237–245, Montreal, Canada, June 2012. Association for Computational Linguistics.

Johann Roturier. Deploying Novel MT technology to Raise the Bar for Quality: a Review of Key Advantages and Challenges. In *Proceedings of the twelfth Machine Translation Summit*, 2009.

Ivan A Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. Multiword Expressions: A Pain in the Neck for NLP. In *Computational Linguistics and Intelligent Text Processing*, pages 1–15. Springer, 2002.

X. Saralegui, I. San Vicente, and A. Gurrutxaga. Automatic generation of bilingual lexicons from comparable corpora in a popular science domain. In *LREC 2008 workshop on building and using comparable corpora*, 2008.

Carolina Scarton, Marcos Zampieri, Mihaela Vela, Josef van Genabith, and Lucia Specia. Searching for context: a study on document-level labels for translation quality estimation. *Proceedings of EAMT*, 2015.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016a.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany, August 2016b. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P16-1009`.

Fei Sha and Fernando Pereira. Shallow parsing with conditional random fields. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 134–141, 2003.

Libin Shen, Jinxi Xu, and Ralph Weischedel. A new string-to-dependency machine translation algorithm with a target dependency language model. In *In Proceedings of Association for Computational Linguistics*, pages 577–585, 2008.

Yongzhe Shi, Wei-Qiang Zhang, Jia Liu, and Michael T. Johnson. RNN language model with word clustering and class-based output layer. *EURASIP Journal on Audio, Speech, and Music Processing*, 2013, 2013.

Nakatani Shuyo. Language Detection Library for Java, 2010. URL `http://code.google.com/p/language-detection/`.

Michel Simard and Atsushi Fujita. A Poor Man's Translation Memory Using Machine Translation Evaluation Metrics. In *Proceedings of the 10th Biennial Conference of the Association for Machine Translation in the Americas (AMTA)*, San Diego, California, USA, 2012.

Michel Simard and Pierre Isabelle. Phrase-based machine translation in a computer-assisted translation environment. *Proceedings of the Twelfth Machine Translation Summit (MT Summit XII)*, pages 120–127, 2009.

Michel Simard, Cyril Goutte, and Pierre Isabelle. Statistical Phrase-based Post-editing. In *In Proceedings of NAACL*, 2007a.

Michel Simard, Nicola Ueffing, Pierre Isabelle, and Roland Kuhn. Rule-based translation with statistical phrase-based post-editing. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 203–206, 2007b.

Mittul Singh, Clayton Greenberg, Youssef Oualil, and Dietrich Klakow. Sub-word similarity based search for embeddings: Inducing rare-word embeddings for word similarity tasks and language modelling. In *COLING*, pages 2061–2070. ACL, 2016.

James Smith and Stephen Clark. Ebmt for smt: a new ebmt-smt hybrid. In *Proceedings of the 3rd International Workshop on Example-Based Machine Translation*, pages 3–10, 2009.

R. Jason Smith, Chris Quirk, and Kristina Toutanova. Extracting parallel sentence from comparable corpora using document level alignment. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 403–411. Association for Computational Linguistics, 2010.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of association for machine translation in the Americas*, pages 223–231, 2006a.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *In Proceedings of Association for Machine Translation in the Americas*, pages 223–231, 2006b.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of AMTA*, AMTA, 2006c.

Matthew Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. Fluency, Adequacy, or HTER? Exploring Different Human Judgments with a Tunable MT Metric. In *Proceedings of the WMT workshop*, EACL 2009, 2009.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 1201–1211, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

Harold Somers. *An overview of EBMT*, pages 3–57. Springer, 2003.

Mark Steedman. *The Syntactic Process*. MIT Press, Cambridge, MA, USA, 2000. ISBN 0-262-19420-1.

Andreas Stolcke. SRILM—an extensible language modeling toolkit. In *Proceedings of the international conference on Spoken Language Processing, Volume 2*, pages 901–904, 2002.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. *CoRR*, abs/1409.3215, 2014.

Liling Tan and Santanu Pal. Manawi: Using multi-word expressions and named entities to improve machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 201–206, Baltimore, Maryland, USA, June 2014. Association for Computational Linguistics.

TAUS Report. Post editing in practice. Technical report, TAUS, 2010. URL `http://www.translationautomation.com/reports/postediting-in-practice`.

TAUS/CNGL Report. Maschine Translation Post-Editing Guidelines Published. Technical report, TAUS, 2010. URL `http://www.cngl.ie/tauscngl-machine-translation-post-editing-guidelines-published`.

Zhaopeng Tu, Yang Liu, Qun Liu, and Shouxun Lin. Extracting Hierarchical Rules from a Weighted Alignment Matrix. In *In Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1294–1303, 2011.

Zhaopeng Tu, Yang Liu, Yifan He, Josef van Genabith, Qun Liu, and Shouxun Lin. Combining Multiple Alignments to Improve Machine Translation. In *The 24th International conference of computational linguistics (Coling 2012)*, pages 1249–1260, 2012.

Marco Turchi, Rajen Chatterjee, and Matteo Negri. WMT16 APE shared task data, 2016. URL `http://hdl.handle.net/11372/LRT-1632`. LINDAT/CLARIN digital library at Institute of Formal and Applied Linguistics, Charles University in Prague.

Hiroshi Uchida, Meiying Zhu, and Md. Anwarus Salam Khan. UNL explorer. In *Proceedings of COLING 2012: Demonstration Papers*, pages 453–458, Mumbai, India, December 2012. The COLING 2012 Organizing Committee.

Masao Utiyama, Graham Neubig, Takashi Onishi, and Eiichiro Sumita. Searching Translation Memories for Paraphrases. In *Machine Translation Summit XIII*, pages 325–331, 2011.

Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

Tony Veale and Andy Way. Gaijin: A Bootstrapping, Template-driven Approach to Example-based MT. In *Proceedings of the Recent Advances in Natural Language Processing*, 1997.

Sriram Venkatapathy and Aravind K. Joshi. Using information about multi-word expressions for the word-alignment task. In *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*, MWE '06, pages 20–27, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics. ISBN 1-932432-84-1. URL `http://dl.acm.org/citation.cfm?id=1613692.1613697`.

Lucas Vieira. Indices of cognitive effort in machine translation post-editing. *Machine Translation*, 3(28):187–216, 2014.

David Vilar, Maja Popović, and Hermann Ney. AER: Do we need to improve our alignments. In *In Proceedings of the International Workshop on Spoken Language Translation*, pages 205–212, 2006.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. HMM-based word alignment in statistical translation. In *Proceedings of the 16th conference on Computational linguistics-Volume 2*, pages 836–841. Association for Computational Linguistics, 1996.

Frane Šarić, Goran Glavaš, Mladen Karan, Jan Šnajder, and Bojana Dalbelo Bašić. Takelab: Systems for measuring semantic text similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, SemEval '12, pages 441–448. Association for Computational Linguistics, 2012.

Mingxuan Wang, Zhengdong Lu, Hang Li, Wenbin Jiang, and Qun Liu. *gen*cnn: A convolutional architecture for word sequence prediction. *CoRR*, abs/1503.05034, 2015.

Warren Weaver. Translation. In *Machine Translation of Languages*, pages 15–23. MIT Press, Cambridge, MA, 1949.

Xianchao Wu, Takuya Matsuzaki, and Jun'ichi Tsujii. Effective Use of Function Words for Rule Generalization in Forest-Based Translation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Techologies*, pages 22–31, Portland, Oregon, USA, June 2011.

Fei Xia and Michael McCord. Improving a statistical mt system with automatically learned rewrite patterns. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, 2004.

Peng Xu, Jaeho Kang, Michael Ringgaard, and Franz Och. Using a dependency parser to improve smt for subject-object-verb languages. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 245–253, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics. ISBN 978-1-932432-41-1. URL `http://dl.acm.org/citation.cfm?id=1620754.1620790`.

Kenji Yamada and Kevin Knight. A Syntax-based Statistical Translation Model. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, pages 523–530, 2001.

Lichi Yuan. Language model based on word clustering. In *Proceedings of the 20st Pacific Asia Conference on Language, Information and Computation*, 2006.

Marcos Zampieri and Mihaela Vela. Quantifying the Influence of MT Output in the Translators Performance: A Case Study in Technical Translation. In *Proceedings of the EACL Workshop on Humans and Computer-assisted Translation (HaCat)*, pages 93–98, 2014.

Anna Zaretskaya, Gloria Corpas Pastor, and Miriam Seghiri. Integration of machine translation in cat tools: State of the art, evaluation and user attitudes. *SKASE Journal of Translation and Interpretation*, 8(1):76–88, 2015a.

Anna Zaretskaya, Gloria Corpas Pastor, and Miriam Seghiri. Translators' requirements for translation technologies: a user survey. *New Horizons in Translation and Interpreting Studies*, pages 133–134, 2015b.

Anna Zaretskaya, Gloria Corpas Pastor, and Miriam Seghiri. Translators' requirements for translation technologies: Results of a user survey. In *Proceedings of the Conference New Horizons is Translation and Interpreting Studies*, 2015c.

Matthew D. Zeiler. ADADELTA: An Adaptive Learning Rate Method. *arXiv:1212.5701 [cs.LG]*, 2012.

Hui Zhang, Min Zhang, Haizhou Li, Aiti Aw, and Chew Lim Tan. Forest-based Tree Sequence to String Translation Model. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 172–180, 2009.

Bing Zhao, Matthias Eck, and Stephan Vogel. Language model adaptation for statistical machine translation with structured query models. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

Ventsislav Zhechev and Josef van Genabith. Maximising tm performance through sub-tree alignment and smt. In *Proceedings of the Ninth Conference of the Association for Machine Translation in the Americas (AMTA-2010)*, 2010.

Andreas Zollmann and Ashish Venugopal. Syntax Augmented Machine Translation via Chart Parsing. In *Proceedings on the Workshop on Statistical Machine Translation*, pages 138–141, New York City, June 2006.

Index