English-Persian Phrase-based Statistical Machine Translation: Enhanced Models, Search and Training

A THESIS PRESENTED IN FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

COMPUTER ENGINEERING

at Massey University
Albany (Auckland), New Zealand

Mahsa Mohaghegh

2012

ABSTRACT

Machine translation (MT), as applied to natural language processing, has undergone substantial development over the past sixty years. While there are a number of different approaches to MT, there has been increasing interest in statistical machine translation (SMT) as the preferred approach to MT. Advances in computational power, together with the exploration of new methods and algorithms have enabled a general improvement in the output quality in a number of systems for various language pairs using this approach. However, there is a significant lack of research work in the area of English/Persian SMT, mainly due to the scarcity of data for this language pair, and the shortage of fundamental resources such as large-scale bilingual corpora. Several research studies have been published on work in the area of machine translation involving the Persian language; however, results producing fluent, usable output are rare.

This thesis shows how SMT was implemented with this language pair for the first time, and how we created a cutting-edge hybrid SMT system capable of delivering high-quality translation output.

We present the development of what is currently the largest English/Persian parallel corpus, constructed using a web crawler to source usable online data, together with the concatenation of existing parallel corpora. As yet another contribution of the research, we propose an improved hybrid corpus alignment method involving sentence length-based and word correspondence-based models to align words, phrases and sentences in the corpus. We also show the impact that the corpus domain can have on the translation output, and the necessity to consider domains of both bilingual and monolingual corpora where they are included in the training and language models.

Two open-source toolkits, Moses and Joshua, were modified to work with the Persian language, and their behaviour and performance results were compared to determine which performed better when implemented with the Persian language.

We present our work in designing, testing, and implementing a novel, three-level Transfer-based automatic post-editing (APE) component based on grammatical rules, which operates by analysing, parsing, and POS-tagging the output, and implements functions as transformers which perform corrections to the text, from lexical

transformation to complex syntactical rearrangement. We show that rule-based approaches to the task of post-editing are superior to the commonly-used statistical models, since they incorporate linguistic knowledge, and are strong in terms of syntax, morphology, and structural semantics – qualities which are very desirable when performing grammatical correction and syntactical restructuring.

We implement independent manual evaluation as well as standard automatic techniques, in order to assess more accurately the translation output. This evaluation shows that the use of the APE component is able to improve translation output significantly, that is, by at least 25%, resulting in high-quality translation output.

Our system performs well by using a combination of the capabilities of two main MT approaches – SMT and RBMT – in different areas of the system as a whole. SMT provides the main system with consistent, mathematical-based translation, and the Transfer-based algorithm in the APE component operates with comprehensive linguistic rules in order to improve incorrect sentences, and fine-tune translation output. This results in a robust, state-of-the-art system, which noticeably exceeds other currently available solutions for this language pair.

ACKNOWLEDGEMENTS

In the name of God who owns soul and wisdom. These are the best attributes of God.
*~ **Ferdowsi** (935 – 1020)*

Firstly, I would like to express my gratitude to my advisors Professor Dr.Hossein Sarrafzadeh, and Dr. Rezaul Hasan. This thesis would not have been possible without their support, advice, and valuable ideas and suggestions.

I would also like to thank my committee member Dr. Tom Moir for his constant support and valuable advice and critical comments. His advice and patience was much appreciated.

I wish to extend my heartfelt thanks to my parents, whose continuous encouragement lightens my path into higher education.

At this point, I would like to express my everlasting gratitude to my best friend for his patience, support, and love. This thesis would not have been possible without Mike's continuous encouragement and patience. Thank you for reminding me that there are things more important than this work, and for standing next to me during this time.

I also want to thank my best friend Dr. Mandana Arzaghi, who pointed out early that there are life and opportunities also outside of Iran. During these last years, geographical distances were against seeing more each of each other, but you were always there for me as a true friend and a good example of perseverance and success. I know you will always be there as a good influence for me.

I would also like to thank Mehdi Mohammadi who has been working at the University of Shikh Bahaee for the many fruitful discussions and joint experiments related to Automatic Post-Editing and Hybrid translation approaches.

My PhD studies turned out to be an unforgettable experience, mostly thanks to the support from my colleagues and friends. I was lucky to be part of a great group of scientifically ambitious, intelligent, and industrious researchers at Massey e-Centre.

Lastly, I offer my regards and appreciation to all of those who have supported me in many aspects during the course of the research, my friends, family, colleagues, reviewers, etc, and anyone else not mentioned here.

Dedicated to my parents

DECLARATION

The author declares that this is her own work except where due acknowledgement has been given. It is being submitted for a PhD in Engineering to Massey University, New Zealand.

This Thesis describes the research carried out by the author at the School of Engineering, Massey University, New Zealand, from June 2008 to December 2012, supervised by Dr Rezaul Hasan.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ACL	Complement Clause of Adjective
ADV	Adverb
ADVC	Adverbial Complement of Verb
AJCONJ	Conjunction of Adjective
AJPP	Prepositional Complement of Adjective
AJUCL	Adjunct Clause
AOL	America Online
APE	Automatic Post Editing
APOSTMOD	Adjective Post-Modifier
APP	Apposition
APREMOD	Adjective Pre-Modifier
ASR	Automatic Speech Recognition
AVCONJ	Conjunction of Adverb
BLEU	Bilingual Evaluation Understudy
CAT	Computer Assisted Translation
CFGs	Synchronous Context-Free Grammars
CNW	Canada Newswire
COMPPP	Comparative Preposition
C-STAR	International Consortium for Research on Speech Translation
DARPA	Defence Advance Research Project Agency
DG	Dependency Grammar
EBMT	Example-Based Machine Translation
EGIU	English Grammar in Use
ELRA	European Language Resource Association
EM	Expectation–Maximization Algorithm
En	English
ENC	Enclitic Non-Verbal Element
Fa	Farsi
FAMT	Fully Automatic Machine Translation
FLDB	Farsi Linguistic Database
FST	Finite State Transducer
FTD	US Air Force's Foreign Technology Division
GLP	Gross Language Product
HAMT	Human-Assisted Machine Translation
Hiero	Hierarchical
HMT	Hybrid Machine Translation
IBM	International Business Machines Corporation
IEEE	Institute of Electrical and Electronics Engineers, Inc.
IR	Information Retrieval
IRNA	Iranian News Agency
IWSLT	International Workshop on Spoken Language Translation
LG	Link Grammar
LV	Linking Verb
LVP	Light Verb Particle
MAHT	Machine-Assisted Human Translation
MAP	Maximum A-Posteriori
MERT	Minimum Error Rate Training
MESU	Measure
METEOR	Metric for Evaluation of Translation with Explicit Ordering
MLE	Maximum Likelihood Estimation
MOS	Mosnad

MOZ	Ezafe Dependent
MPEC	Modern Persian-English corpus
MST	Parser Maximum Spanning Tree Parser
MT	Machine Translation
NADV	Adverb of Noun
NCL	Clause of Noun
NCONJ	Conjunction of Noun
NE	Non-Verbal Element of Infinitive
NEZ	Ezafe Complement of Adjective
NIST	National Institute of Standards and Technology
NLP	Natural Language Processing
NPEC	News Persian English Corpus
NPOSTMOD	Post-Modifier of Noun
NPP	Preposition 0f Noun
NPREMOD	Pre-Modifier of Noun
NPRT	Particle of Infinitive
NSPEC	News Subtitle Persian-English Corpus
NVE	Non-Verbal Element
ODJ	Object
ODJ2	Second Object
OOV	Out of Vocabulary
PAHO	Pan American Health Organization
PARCL	Participle Clause
PART	Interrogative Particle
PB	Phrase Based
PCONJ	Conjunction of Preposition
PCTS	Parallel Corpus Test Set
PeEn-SMT	Persian-English Statistical Machine Translation
PEN	Parallel English-Persian News Corpus
POS	Part of Speech
POSDEP	Post-Dependent
PPL	Perplexity Threshold
PREDEP	Pre-Dependent
PREM	Pre-Modifier
PRO	Predicate
PROG	Progressive Auxiliary
PUNC	Punctuation Mark
RBMT	Rule-Based Machine Translation
RHS	Right Hand Side
ROOT	Root
SAMT	Syntax Augmented Machine Translation
SBJ	Subject
SCFG	Stochastic Context-Free Grammar
SDL	Scalable Enterprise Translation Server
SDL	SDL Language Weaver
SMT	Statistical Machine Translation
SOV	Subject-Object-Verb
SRILM	Sri Language Model
SVO	Subject–Verb-Object
TAM	Tamiz
TEP	Tehran English-Persian Corpus
TER	Translation Error Rate Te
TER	Translation Error Rate
TMC	Tehran Monolingual Corpus
U	Unicode

UN	United Nations
UTIRE	University of Tehran Information Retrieval Evaluation System
VCL	Complement Clause of Verb
VCONJ	Conjunction of Verb
VPP	Prepositional Complement of Verb
VPRT	Verb Particle
WER	Word Error Rate
WSD	Word Sense Disambiguation

LIST OF PUBLICATIONS

1. **Mahsa Mohaghegh**, Abdolhossein Sarrafzadeh, Mehdi Mohammadi, ***GRAFIX: Automated Rule-Based Post Editing System to Improve English-Persian SMT*** (Short paper- COLING 2012, the 24th International Conference on Computational Linguistics. Mumbai, India, December 2012)
   http://aclweb.org/anthology-new/C/C12/C12-2085.pdf

2. **Mahsa Mohaghegh**, Abdolhossein Sarrafzadeh, ***A Hierarchical Phrase-Based Model for English-Persian Statistical Machine Translation*** (Full Paper - Innovations 12, 8th International Conference on Innovations in Information Technology. AL AIN ,UAE, March 2012)
   http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6207733

3. **Mahsa Mohaghegh**, ***Advancements in English-Persian Hierarchical Statistical Machine Translation.*** (Short paper - NZCSRSC New Zealand Computer Science Research Student Conference. Otago, April 2012)
   https://sites.google.com/a/nzcsrsc.ac.nz/nzcsrsc2012/programme/posters

4. **Mahsa Mohaghegh**, Abdolhossein Sarrafzadeh, Tom Moir, ***Improving Persian-English Statistical Machine Translation: Experiments in Domain Adaptation*** (Full paper – *In Proceedings of the 2nd Workshop on South and Southeast Asian Natural Language Processing (WSSANLP)*, IJCNLP 2011, pages 9–15,Chiang Mai, Thailand, November 2011)
   http://www.aclweb.org/anthology/W/W11/W11-3002.pdf

5. **Mahsa Mohaghegh**, Abdolhossein Sarrafzadeh, ***An Overview of the Challenges and Progress in PeEn-SMT: First Large Scale Persian-English SMT System*** (Full Paper - Innovations 11, 7th International Conference on Innovations in Information Technology. Abu Dhabi, April 2011)
   http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5893841

6. **Mahsa Mohaghegh**, Abdolhossein Sarrafzadeh, ***The Impact of Domain for Language Model in the PeEn-SMT: First Large Scale Persian-English SMT System***. (Short paper-NZCSRSC New Zealand Computer Science Research Student Conference. Palmerston North, April 2011)

   https://sites.google.com/a/maori.geek.nz/nzcsrsc2011/papers/paper-abstracts#TOC-Mahsa-Mohaghegh-The-Impact-of-Domain-for-Language-Model-in-the-PeEn-SMT:-First-Large-Scale-Persian-English-SMT-System-

7. **Mahsa Mohaghegh**, Abdolhossein Sarrafzadeh, ***Multilingual Information Service System for Tourists*** (Poster – *NZBio Conference*, Auckland, March 2011)

8. **Mahsa Mohaghegh**, Abdolhossein Sarrafzadeh, Tom Moir**.** ***Improved Language Modelling for English-Persian Statistical Machine Translation.*** (Full Paper - *In Proceedings of SSST-4 Workshop at COLING-2010*, Beijing, China, August 2010)

http://www.aclweb.org/anthology-new/W/W10/W10-3810.pdf

9. **Mahsa Mohaghegh**, Abdolhossein Sarrafzadeh. ***Performance Evaluation of Statistical English-Persian Machine Translation.*** (Full Paper – JADT2010 – *10th International Conference on Statistical Analysis of Textual Data.* Sapienza, University of Rome, June 2010)

   http://www.ledonline.it/ledonline/JADT-2010/allegati/JADT-2010-1091-1100_114-Sarrafzadeh.pdf

10. **Mahsa Mohaghegh**. ***A Statistical Approach to English-Persian Machine Translation.*** (Short Paper – NZCSRSC – *New Zealand Computer Science Research Student Conference.* Wellington, April 2010)

    http://ecs.victoria.ac.nz/Events/NZCSRSC2010/Papers

11. **Mahsa Mohaghegh**, Tom Moir, Abdolhossein Sarrafzadeh. ***A Statistical Approach to English-Persian Machine Translation.*** (Poster – *NZBio Conference*, Auckland, March 2010)

    http://www.academia.edu/238736/A_Statistical_Approach_to_English-Persian_Machine_Translation

12. **Mahsa Mohaghegh**, Abdolhossein Sarrafzadeh. ***Analysis of the Effect of Data Variation in a Statistical English-Persian Machine Translation.*** (Full Paper - Innovations '09 – *6th International Conference on Innovations in Information Technology*. Dubai, December 2009.

    http://dl.acm.org/citation.cfm?id=1802285
    http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=05413782

13. **Mahsa Mohaghegh**, Tom Moir. ***First English-Persian Machine Translation.*** (Poster – NZPGC – *New Zealand Postgraduate Conference*. Wellington, November 2009)

    http://www.academia.edu/238733/the_first_english-persian_statistical_machine_translation

Chapter 1. Introduction

*"The key to growth is the **introduction** of higher dimensions of consciousness into our awareness"*

~ Lao Tzu

Throughout history, and more specifically in the past century, advances in technology have enabled increasing interaction between peoples of different countries, cultures and languages. Technological developments resulting in ease of travel and communication alike have contributed to this. The arrival and development of the internet spelled a major breakthrough in communication techniques, and provided extensive new opportunities and possibilities on both personal and commercial levels that were previously inconceivable. With the development of Web 2.0 technologies, communication between international businesses became significantly easier, and concepts such as face-to-face conference calls became feasible. Even now in the early 21st century, recent advances in Artificial Intelligence such as emotional recognition through physical gestures continue to remind us that we are only on the brink of a new era, the technological bounds of which are yet to be discovered.

However, as mankind advances with new developments and inventions, and as globalisation and international travel and commerce increase as a result of these great feats, a restrictive barrier is encountered, somewhat limiting the extent to which communication technology can be applied. This challenge is the language barrier.

MT (machine translation) was one of the first applications of natural language processing, and involves translation from one human language to another. There are a number of different approaches to MT, one of which is SMT (statistical machine translation), an increasingly favoured approach which involves determining the maximum probability of a translation output by analysing patterns in previously translated text.

1.1 Problem Statement

SMT has seen very limited use with the English/Persian language pair, and at the time of this study's commencement, there was no documented research work. To date, aside from Google Translate, all other applications using this approach for

English/Persian are still very much at an early stage, and fall far short of yielding consistently accurate translation results. This is due to the fact that there are very few available resources necessary for the construction of an effective system.
This research seeks to address the following research questions and areas:

1. How might an English/Persian SMT system be constructed such that it is able to consistently provide fluent translation output, despite working with low data resources?
2. What methods could be used to obtain or generate more parallel data? What aspects of data domains are important in a parallel corpus?
3. What refining processes could be implemented in order to improve the quality of available resources?
4. Which decoding approach works best with morphologically-rich languages such as Persian?
5. What techniques may be developed to determine the meanings of unknown words based on their context?
6. How can the initial output of a baseline system be improved in terms of fluency and grammatical correctness?

1.2 Scope of the Study

The scope of this research was originally intended to cover the development of a handheld speech-to-speech translator for the Persian-English language pair. However, shortly after this venture was undertaken, it became clear that developing the translation engine alone was an immense task in itself. This was because very little work had been done in the area of SMT for this particular language pair, and what had been done had yielded very unsatisfactory results. Given the significant shortage of resources required for successful development, it became apparent that the development of the translation engine would easily cover the duration of time allotted to the whole research project. Because of this, it was decided that the focus needed to be shifted solely on the development and improvement of the central part of the system – the statistical machine translation engine.

The objectives of this thesis are outlined as follows:

1. Thoroughly review current research work to date on SMT systems for English-Persian (where available), as well as other MT approaches for this language pair. Review SMT systems involving languages similar to Persian, such as Arabic.
2. Determine the shortcomings of existing work and define possible solutions for the implementation of an effective SMT system for English-Persian, capable of delivering fluent translation in both translation directions.
3. Develop a large-scale parallel corpus using a web crawler to source bilingual web pages. Determine the effect corpus domain has on the output quality, and determine methods which can be implemented to monitor corpus domain effect. Test different phrase and word alignment methods, determining which is best for English-Persian.
4. Modify various open-source toolkits to work with the Persian language, determining which shows the best performance.
5. Explore and determine methods of achieving high-quality output, despite working with limited data resources.
6. Experiment with hybrid translation system architecture, and develop a hybrid SMT system coupled with an automatic post-editing method to further improve translation output in the English–Persian translation direction.

1.3 Research Challenges

This area of research presented a number of challenges. Persian and English are vastly different languages, both in terms of basic sentence structure, and in grammar, syntax, and morphology. This can cause great complications in any natural language processing task, and often tasks must be tuned and customised specific to that language pair.

One of the greatest challenges was the lack of data for system development. An acute shortage of large-scale bilingual corpora available for English-Persian means that any system working with this language pair is forced to operate with low resources. Since the parallel corpus is the most important component of the system, obstacles were

present right from the outset. This data had to be sourced both manually and via a web crawler, and a parallel corpus was constructed and developed in-house.

This data scarcity meant there was no documented research in this area at the time this study commenced, and there is still very little to date. This situation presented another challenge, in that grading the comparative performance of the system was close to impossible. It was only after Google Translate released support for the Persian language that we had a system for reliable comparison of performance.

1.4 Contributions to Knowledge

In summary, we have developed the first SMT system for the English/Persian language pair that is able to consistently produce fluent output, despite being limited by data scarcity. This was accomplished through:

1. Compiling and aligning what is currently the largest English/Persian parallel corpus. This corpus is soon to be made publicly available for future research, which will be of great benefit to this study area, since the shortage of bilingual data is a challenge for NLP tasks dealing with this language pair.
2. Developing a hybrid-architecture SMT approach focussed on achieving the highest possible performance using limited resources. Certain techniques and processes that have been developed for this system are able to be implemented in other systems dealing with low-resource languages (e.g., Maori), and can significantly improve their performance. This entire MT system can also be implemented as the translation engine in any speech-to-speech translation system.
3. Development of an APE (automatic post-editing) module to automatically correct grammar and syntax errors in the translation output. The techniques and algorithms developed for the APE component of the system can be used to improve the translation of a number of languages with similar grammatical structure to Persian, such as Dari or Tajik.

In the final stages of development, our system significantly outperformed Google Translate for the English/Persian language pair in both directions of translation. Development of the system during the study period can be found in the list of publications.

1.5 Thesis Outline

The rest of this thesis is structured as follows:

Chapter 2 gives a brief overview of the history of MT as a whole, outlining the general difficulties encountered, its advantages and disadvantages, and giving examples of areas in which MT has been implemented today. It also gives a comprehensive review of the work in the area of English/Persian MT, as well as other language pairs with similar methods. The different approaches to MT are discussed, showing increased favour towards SMT and why this approach has proven to be advantageous over others in recent years. It also gives a brief history of the Persian language, and an overview of the syntax, grammatical features, and general characteristics of the modern language, including reasons why Persian presents significantly greater challenges for SMT than other languages. Finally, it presents a review of related MT works for the English/Persian language pair, both SMT and other approaches, and also covers a summary of available English/Persian parallel corpora and Persian monolingual corpora.

Chapter 3 covers the details behind a statistical machine translation system, showing the different operating models, how language models and training models are generated from monolingual and bilingual corpora, how the corpora themselves are aligned, and how the baseline process works. Also detailed are the evaluation metrics and methods used to automatically evaluate translation output.

Chapter 4 gives details of the initial baseline tests that were undertaken using Moses, and working with bilingual and monolingual corpora developed in-house. The test output results were evaluated using automatic metrics such as BLEU, NIST and TER, and analysis and interpretation of these results were made in order to determine areas that would require focused development. The next section of the chapter presents the work carried out to increase the size of both the monolingual and parallel corpora, and in particular the development of what is currently the largest bilingual English/Persian corpus available. The issue of alignment quality in the parallel corpora is addressed, and a hybrid method for alignment is proposed. It then goes on to detail the experiments that were undertaken in both translation directions using training and language models generated with the new corpora. The effect of corpus size, domain, and overall quality are determined, based on the evaluation metric scores.

Chapter 5 compares the use of standard phrase-based decoders with hierarchical phrase-based decoders, and, in particular, the differences between the open-source toolkits, Moses and Joshua. Tests are run using both decoders and working with identical data sets, and the results evaluated. Evaluation shows that the performance of each decoder was affected by translation direction. Also covered in this chapter is the use of multiple translation referencing to yield a more accurate output evaluation.

Chapter 6 introduces the concept of hybrid translation systems, and shows the development of a method using a novel combination of phrase-based SMT, and rule-based MT (RBMT) in an automatic post-editing (APE) method. The motivation for a post editing component is discussed, and some literatures review of hybrid systems is presented.

Chapter 7 shows the details of our APE approach, including POS-tagging translated text, parsing, and three-level text transformers. It is shown that this hybrid system approach not only achieves much better translation output than previously, but the evaluation metrics score a number of instances at levels similar to that of high-resource language pairs (i.e., German/English) by existing commercial MT systems.

Chapter 8 concludes the thesis by outlining the achievements accomplished and contributions to this field, and possible directions for further development.

Chapter 2. Literature Review

"Literature can remind us that not all life is already written down: there are still so many stories to be told."

~ Colum McCann

2.1 Introduction

This chapter gives an introduction to MT, covering general advantages and disadvantages, and examples of use. The current different approaches to MT are reviewed, specifically SMT. Related work in the area of Persian English MT is presented in detail, together with an overview of the Persian language, addressing the shortage of available digital data necessary for SMT, and reviewing corpora that are available.

2.2 Machine Translation Systems

Machine Translation (MT) has a long history. Concepts of machine-like translation can be traced as far back as the 17th century. In the mid-1600s, René Descartes conceived the theory that it could be possible to equate ideas and thought between languages through intermediate symbols. Real work in this area did not begin until the mid-1930s, where the idea of machines being used as translation tools was first introduced in detail with a revolutionary concept that was pioneered by the French-Armenian *Georges Artsrouni* and the Russian *Petr Troyanskii*. They had developed a "translating machine," and consequently applied for a patent (Hutchins & Somers, 1992).

Broader interest in MT, and specifically the use of computers in this field, began after World War II. The impact of these ideas first became a reality in 1946 when the American mathematician Warren Weaver developed them further (Weaver, 1949b). The Georgetown experiment took place in the early 1950s – a successful machine translation demonstration of approximately sixty sentences from Russian to English. Other ventures included English to French translation work in London, which resulted in the publication of several papers and journal articles. This sparked much more interest in the field and, as a result, research funding. However, in reality, advances were very slow and, in the absence of the expected return, the amount of funding was lowered. It was only around the early 1990s that interest in the field awoke, as

computing technology advanced and computers became faster and more powerful, and the idea of statistical translation methods became feasible.

Over a period of almost fifty years, researchers in this field of computer science have explored a number of different approaches to developing machine translation. Today, we find various kinds of machine translation arrangements: machine-assisted human translation (MAHT), computer-assisted translation (CAT), human-assisted machine translation (HAMT) and fully-automatic machine translation (FAMT). These all have different applications. Present translation technology, however, has not yet been able to deliver fully-automated high-quality translations. In practice, the output from these systems almost always needs to be edited to correct errors.

2.2.1 Machine Translation Difficulties

Spoken language sentences are long and complex, and often contain unpredictable grammatical constructions. They may even have unwanted noise and grammatical errors. These factors, together with the task of finding suitable ways to deal with names and technical terms across languages with different alphabets and sound inventories, make machine translation for natural language a challenging task. Developing techniques for finding meanings of unknown words in context is a challenging problem in both text and speech translation.

Many words have various meanings and different possible translations. In some languages such as Chinese or Japanese, not even the word boundaries are given. Certain grammatical relations in one language might not exist in another language, and sentences involving these relations may need to be significantly reformulated. In addition, there are non-linguistic factors that may need to be considered in order to perform a translation, such as knowledge of cultural history, and cultural etiquette.

To accurately perform machine translation, many dependencies have to be taken into account. Often, these dependencies are weak and vague, which makes it rarely possible to describe simple and relevant rules that hold without exception in the translation process. From a linguistic viewpoint, various types of dependencies must be considered: morphologic, syntactic, semantic and pragmatic dependencies (Jurafsky, Martin *et al.*, 2000).

More specifically, there are dependencies that relate source and target language words, which describe that certain words or phrases can be translations of each other. Some dependencies relate only target language words describing the well-formed

parts of the produced translation. To develop an MT system, a general framework must be found which is able to deal with the weak and vague dependencies. Once such a framework is acquired, certain methods that efficiently obtain the large amount of relevant dependencies must be developed (Och, 2002).

Large-scale natural language processing requires the integration of vast amounts of lexical, grammatical and conceptual knowledge. A robust generator must be able to operate well, even when pieces of knowledge are missing; it must also be resistant to incomplete or inaccurate input.

There are two main issues machines encounter in the area of natural languages. The first is related to context and cultural issues: Computers are unable to perceive the contextual and pragmatic information that humans can. Similarly, they are unaware of cultural differences which often surface in linguistic exchanges.

The second issue relates to the function of language. Conveying meaning is just one application of human language; there are many others in addition, such as humour, establishing solidarity, sharing emotions and feelings without needing to convey any actual information, as well as plays, poetry, advertising, and song lyrics, which are difficult to translate even for humans. Hence, computers encounter great difficulty providing quality translations for these pieces. Ambiguity, idioms, differences in vocabulary, collocations, and structural and lexical differences between the source and target languages are also difficulties which a machine translating system must deal with (Arnold, Balkan *et al.*, 1994; Gross, 1992).

We can identify the kinds of linguistic errors that might be expected in the raw output yielded by fully automated machine translation (FAMT), and classify them into two groups: vital errors (impeding accurate translation of meaning), and errors which merely affect the general fluency and readability of the text, without actually changing or subtracting from the intended meaning.

Despite some of the negative aspects of machine translation, machines also hold many qualities that make machine translation very attractive. Machines are usually unchanging in interpretation and vocabulary; they do not omit words or paragraphs by accident, and do not make the erroneous conclusions that can be made even by competent human translators. According to (Gross, 1992), machines, for the most part, have potential to be faster, more economical, and provide translations with a greater degree of accuracy than human translators. This is particularly so if the machines are limited to a specific subject domain, just as human translators are.

People's overall mistrust and uncertainty about computers and technological advances may, he states, be the main cause behind their scepticism towards machine translation, rather than a legitimate criticism of MT.

2.2.2 Examples of Use

The English-Spanish translation system used by the Pan American Health Organization (PAHO) is one of the earliest machine translation engine examples. This MT engine was built to specialise in medical data. The primary purpose of the PAHO study was to assess the quality of machine translations in comparison to human translations. The evaluation showed machine translation to be a faster and easier method (Vasconcellos & Bostad, 1992).

A significant user of MT tools is the US Air Force's Foreign Technology Division (FTD). A partially-edited text which was translated from Russian into English was assessed by independent sources. Their conclusions regarding the comparison between the machine's translation and the human translation were summed up in the following: "While the [human] translation read somewhat more smoothly, it seemed to use inappropriate or erroneous terminology more often than the [machine] translation did. Consequently, we relied primarily on the [machine] translation, using the [human] translation mainly for reference" (Vasconcellos & Bostad, 1992).

According to Hutchins and Somers (1992), by far the most successful machine translation has been used in the translation of French weather predictions. "The METEO system, which translates daily more than 30,000 words of weather bulletins from English into French at a cost of less than 0.5¢ (Canadian) per word, with an accuracy rate of 95 per cent" performs a "boring" job which human translators might be unwilling to do," explains Somers (Hutchins & Somers, 1992).

A further application of MT is for web-searches and translation of web pages. As the internet is flooded with new users from various countries and language backgrounds, linguistic demands are also increasing and languages become barriers to communication. As a consequence, many search engines (Google, Bing, AltaVista, Yahoo, etc.) provide machine translation services to counter this problem.

2.2.3 Machine Translation Advantages and Disadvantages

Machine translation has some advantages over traditional professional human translation. MT systems are usually very simple and easy to use. Since translation is

performed quickly and usually on-demand, it is much more convenient to use than a human translator. On top of this, there is the cost factor: professional human translation is usually costly, not available on-demand, and when it is, requires much more time to complete.

Disadvantages of general MT systems include the fact that translation output is usually lacking in accuracy to some degree, especially when focussed on a particular domain. Translations made for technical or scientific writings are usually inaccurate, unless (in the case of Statistical MT) the system has been trained using data from that particular domain. The accuracy of machine translation is not guaranteed. If a poor quality translation is generated, there is usually no real way of knowing, unless someone who speaks both target and source languages is able to compare and evaluate them. This can pose some problems, particularly if translating sensitive or private documents.

2.3 Machine Translation Approaches

Over the years, researchers have applied different techniques to approach the challenge of machine translation; the most significant of these are outlined in the following subsections.

One approach to machine translation is the rule-based approach. This method is based on dictionary entries, which means that each word will be translated as a dictionary translates – word by word. The meanings of these words are not always interchangeable (hence the name “rule-based”) (Carbonell, Cullinford *et al.*, 1978).

Another type of machine translation is transfer-based MT. Transfer-based MT is a type of machine translation based on the idea of Interlingua and is currently one of the most widely-used methods of machine translation. Both transfer-based and Interlingua-based MT approaches use an intermediate representation which captures the "meaning" of the original sentence in order to generate the correct translation (Shirko, Omar *et al.*, 2000).

Although transfer-based machine translation systems work in different ways, they generally pursue the same configuration. They refer to a set of linguistic rules which compare the syntax structure in the source language with that in the target language to generate a result. The first stage of this process involves analyzing the input text for morphology and syntax (and sometimes semantics) to create an internal

representation. The translation is produced using this representation, together with bilingual dictionaries and grammatical rules.

In the Direct Approach, words are translated directly without passing through an additional representation. In the transfer approach, the source language is transformed into an abstract, less language-specific representation. Linguistic rules which are specific to the language pair then transform the source language representation into an abstract target language representation and, from this representation, the target sentence is generated.

Interlingua Machine Translation is one of the more classic approaches to machine translation. In this approach, the source language is transformed into an Interlingua, that is, an abstract language-independent representation. The target language is then generated from the Interlingua. Within the rule-based machine translation paradigm, the Interlingua approach is an alternative to the direct approach and the transfer approach (Leavitt, Lonsdale *et al.*, 1994). The Interlingua approach to machine translation has advantages and disadvantages. One advantage in multi-lingual machine translations is that no transfer component has to be created for each language pair. The obvious disadvantage is that the definition of an Interlingua is difficult, and, in some cases, may be even impossible for wider domains (Levin, Gates *et al.*, 1998).

In a Rule-based Machine Translation system, the original text is first analysed morphologically and syntactically in order to obtain a syntactic representation. This representation can then be refined to a more abstract level, putting emphasis on the parts relevant for translation and ignoring other types of information. The transfer process then converts this final representation (still in the original language) to a representation of the same level of abstraction in the target language.

The Example-based Machine Translation (EBMT) approach is often characterised by its use of a bilingual corpus at run time, with parallel texts as its main knowledge base. It is essentially a translation by analogy and can be viewed as an implementation of the case-based reasoning approach of machine learning. Example-based machine translation was first suggested by Nagao Makoto in 1994 (Nagao, 1994). At the foundation of example-based machine translation is the idea of translation by analogy. When applied to the process of human translation, the idea that translation takes place by analogy is a rejection of the idea that people translate sentences by performing deep linguistic analysis. Instead, it is founded on the belief that people translate first

by decomposing a sentence into certain phrases, then by translating these phrases, and finally by properly composing the fragments into one long sentence.

2.3.1 Statistical Machine Translation

The statistical machine translation approach was first proposed by Warren Weaver in 1949, but it has only seen significant development in the past three decades. During this time, it has become increasingly popular through successful implementation.

There was interest in statistical approaches to statistical machine translation in the early 1950s, and around that time some research work was proposed and begun (Weaver, 1949a). However, it seems that the overall vastness of tasks involved with SMT was not fully realised at first, and as certain research projects in this area were faced with the need for huge amounts of digital text, not to mention the computational power to process them, funding for these projects was reduced. The late 1980s and early 1990s saw progress in technology and increases in computational capacity, and, as a result, more serious interest was rekindled in the statistical approach, as the technologies made the tasks involved more feasible. Increased research and use of statistical machine translation yielded promising results, and made the funding of such projects more attractive.

Statistical machine translation is based on the theory that statistical models for translating between two languages can be learned from large parallel corpora of translated text. In the following pages, we will introduce some of the basic concepts and techniques.

The alignment of words and phrases in a parallel corpus turns out to be the most difficult problem statistical machine translation faces. Words and phrases in the source and target languages normally differ in where they are placed in a sentence. Words that appear on one language side may be dropped on the other. Concepts may be expressed by means of different syntactical categories. One English word may have as its counterpart a longer German phrase, and vice versa. Certain words, phrases and expressions while being common in one language, might not even exist in the other. Statistical machine translation models assume the approach that every sentence in the target language is a translation of the source language sentence with some level of probability. Therefore, the best translation is, of course, the sentence that has the highest probability for occurrence. Due to this, the key problems in

statistical machine translation lie in estimating the probability of a translation, and efficiently finding the sentence with the highest probability (Ramanathan, 2008).

Statistical machine translation is a machine translation paradigm where translations are generated on the basis of statistical models whose parameters are derived from the analysis of bilingual text corpora. The statistical approach is different compared to both the rule-based and example-based approaches to machine translation.

Unfortunately, even for simple translation models, the search problem in SMT is NP-complete. Various research groups have attempted to extend IBM's work to develop more efficient search algorithms by using suitable simplifications and applying better optimization methods. Beam search and dynamic programming-based monotone search with a time complexity linear to the input length has been suggested by (C Tillmann, Vogel *et al.*, 1997). In (Vogel, Och *et al.*, 2000) this was extended to also handle word reordering. (Ney, Nießen *et al.*, 2000) suggested a simplified recombination rule in dynamic programming search to obtain a polynomial time search algorithm, even in the case of general reordering. Various researchers have suggested greedy or perturbation search approaches (A. Berger, Brown *et al.*, 1994).

SMT was introduced by the seminal work of a research group at IBM (PF Brown, Della Pietra *et al.*, 1990). They introduced the concept of alignment models to describe the dependencies between source and target language words (PF Brown, Della Pietra *et al.*, 1993). A. Berger, Brown *et al.* (1994) developed a search algorithm for these models based on the paradigm of stack decoding (A. L. Berger, Brown *et al.*, 1996).

One ground-breaking publication which first described the aforementioned techniques to MT in the early 1990s was by PF Brown, Della Pietra *et al.* (1993). While they used a purely word-based approach, the currently best-performing SMT systems are of the phrase-based type (Koehn, Och *et al.*, 2003b), that is, they use phrases instead of words as the smallest translation unit.

A recent innovative approach has been integer programming as the framework for an optimal search algorithm for (Germann, Jahr *et al.*, 2001). Here, the search problem is reformulated as an integer programming optimization problem and a standard toolkit is used to solve it. However this approach is only applicable to very short sentences.

2.3.2 Advantages of the Statistical Approach for Machine Translation

There are a number of significant benefits SMT holds in comparison to traditional paradigms. These benefits alone cannot exclusively conclude that SMT is a superior system for a certain language pair. Systematic evaluations and testing must be carried out to determine this.

One benefit of the statistical approach is that, generally speaking, SMT systems are not language-pair specific. The linguistic rules in rule-based translation systems require manual development, and a significant amount of work must be done defining vocabularies and grammar. These rules and language vocabularies and grammar are not easily mirrored to other languages, if at all (PF Brown, Della Pietra *et al.*, 1990).

Most other machine translation approaches rely on linguistic rules in order to analyse the source sentence, mapping the semantic and syntactic structure into the target language. The statistical approach employs algorithms to obtain data from existing translation compilations called bilingual corpora. These corpora are effectively huge aligned banks of phrases and words. Algorithms statistically determine the best translation output based on the phrases in the corpora. Hence, it can be seen that since the SMT approach is, in reality, based on the use of pre-existing aligned language pairs, its output should in theory be more reliable.

Machine translation is a decision problem, in that once given the source language words and phrases as input, the target language words and phrases must be decided upon. This being the case, it is logical to solve the problem with the methods from statistical decision theory leading to the suggested statistical approach.

The relationships between linguistic objects such as words, phrases or grammatical structures, are often weak and vague. To model those dependencies, we need a formalism, such as offered by probability distributions, that is able to deal with these dependencies.

To perform machine translation, it is typically necessary to combine many knowledge sources. In statistical machine translation, we have a mathematically well-founded system to perform an optimal combination of these knowledge sources.

In SMT, translation knowledge is learned automatically from example data, and, as a result, the development of an MT system based on statistical methods is very fast compared to a rule-based system. SMT is well-suited for embedded applications where machine translation is part of a larger application.

The ‘correct’ representation of syntactic, semantic and pragmatic relationships is not known. Hence, where possible, the formalism should not rely on constraints induced by such hypothetical levels of description. Instead, in the statistical approach, the modelling assumptions are empirically verified on training data.

One aspect of statistical machine translation that is an indisputable advantage over rule-based approaches lies in SMT’s adaptability to different domains and languages. In general, once a functional system exists, all that has to be done in order to implement it on other language pairs or text domains is to train it on new data.

While SMT has been shown to yield promising results for large amounts of language, it can be shown that rule-based systems are more effective where the source sample is shorter.

2.4 Related Work in Statistical Machine Translation

The work of IBM’s group in the late 1980s reawakened more serious work in statistical machine translation (P. Brown, Cocke *et al.*, 1988; PF Brown, Della Pietra *et al.*, 1990; PF Brown, Della Pietra *et al.*, 1993). In these early approaches, single word probabilities were used (single-word based lexicon), and word alignment (aligning source and target language words) was first proposed. In a 1999 John Hopkins University workshop, IBM implemented open-source training software for their models.

GIZA tool, the main component in IBM’s software, used EM-trained (Expectation–Maximization) word alignment models. This tool was later extended to GIZA++ (Och & Ney, 2003). Stack decoding, multi-stack decoding, greedy techniques and dynamic programming were all decoding techniques on which the search algorithms were based (Germann, Jahr *et al.*, 2001; Ney, Nießen *et al.*, 2000; C. Tillmann & Ney, 2003; C Tillmann, Vogel *et al.*, 1997; Y. Y. Wang & Waibel, 1997).

Since those early days, the most popular translation approach tends to be phrase-based. Most of them are adaptations of the alignment template approach, where alignment templates describe the alignment of source and target phrases, defined at the word-class level (Och, 2002; Och & Ney, 2004; Och, Tillmann *et al.*, 1999). These templates are sourced out of word-aligned parallel corpora. Tests showed that this alignment approach performed better than single-word based approaches (Och, 2002; Och, Tillmann *et al.*, 1999).

Word classes are not used in the phrase-based approach. As stated before, phrases are defined at word-class level, and are extracted from aligned corpora using an algorithm identical to that used for alignment templates.

Tom´as and Casacuberta use a similar approach, as shown in Tom'as and Casacuberta (2001). They use the EM algorithm to determine the phrase translation probabilities, while constraining monotonic phrase segmentation. The experiments they performed were on the Spanish/Catalan language pair, where constraining monotonic phrase segmentation is suitable. However, such segmentation may not be successful for dissimilar language pairs. IBM model 1 lexicon was used for phrase identification, and later on they removed monotonic decoding from the approach.

Marcu and Wong (2002) perform phrase-based translation using a joint probability model. Their approach generates alignment of phrases directly, instead of employing word alignment for extraction. Again, in their approach, Marcu and Wong use the EM algorithm for translation training, and a modified greedy decoder for deciding single-word based models. However, this approach does not operate well with tasks of large data.

2.5 Related Work in Persian MT

Bakhshaei, Khadivi, Riahi *et al.* (2010) use a statistical phrase-based system to translate the English/Persian language pair, and investigate how different parameters within the system affect the translation output accuracy, as measured by the evaluation metric BLEU. [More details about evaluation metrics such as BLEU, NIST and TER are covered in chapter 3.] Their best improvement is 1.84% relative to the baseline accuracy (from 0.1697 improved to 0.1881). In their study they identified a number of parameters significantly affecting the output accuracy in their system. The quality of the translation is dependent on both the quality of the phrase table, and the quality of tuning in the decoding parameters.

In parameters relating to the language model, they show that increasing the language model order gives an increase in the volume model, and, as a result, will decrease system perplexity. This can lead to an increase in the test set BLEU score. They also show that the phrase table may be improved by increasing the allowed maximum length of extracted phrases, which leads to an improved BLEU score. Their research

results show that a phrase length of seven words is optimum for their English-Persian system.

They showed that an increase in search space can be accomplished by modifying several parameters. First, they show that increasing the distortion limit (allowed displacement of phrases in a sentence) will increase the output score, due to the differences in the POS order between English and Persian. Based on their results, they give a value of eight as the best distortion limit for this language pair.

Another factor found to positively influence the BLEU score was increasing the translation table limit. They show that a number of choices for input phrase translation may be affected by limiting the candidate hypothesis. The authors also show that stack size can influence output quality. During the translation process, a stack of the best possible translation choices for each input word is kept by the decoder. If this stack is increased in size, the number of hypotheses is also increased. This tends to increase output accuracy.

They compare their results with that of Google Translate, and also a human translation. They encounter a number of out-of-vocabulary (OOV) occurrences, probably due to the relatively small corpus size. They consider their best result of 0.1881 to be reasonable considering the nature of their corpus, its small size, and the general difficulties that ensue when performing machine translation with largely different languages such as Persian and English. However, output with quality as such scored is still largely unsuitable for practical purposes.

Bakhshaei, Khadivi, and Riahi (2010) use the SMT approach to develop an MT system for the Persian/German language pair. Because of the significant shortage of Persian-German parallel data (even more so than English-Persian), they use the English language as a bridge language between German and Persian, and vice versa. In effect, they combine two SMT systems, Persian-English and English-German, into one. This involved the use of an existing English-German parallel corpus, and manual translation of a significant portion of the English into Persian to provide the necessary training data. In this paper they show that their particular combination outperforms the Persian-German baseline system by approximately 15%.

Other previous work with bridging languages includes development of word alignment (Kumar, Och *et al.*, 2007; H. Wang, Wu *et al.*, 2006) and work with cross-language retrieval (Gollins & Sanderson, 2001).

The authors perform translation using phrase-based SMT developed using Moses decoder (P. Koehn, H. Hoang *et al.*, 2007b), with a 5-gram language model. They effectively construct a usable Persian-German parallel corpus together with a new English-Persian corpus. They show that combination of their Persian-English and English-German SMT systems at phrase level gives an increase in the BLEU score from 0.181 to 0.208, or approximately 15%, in the Persian-German language direction.

The authors show that correct manipulation of a bridging language can improve translation output accuracy for low-resource languages involved in SMT. Their chosen bridging language was English because of the resources it had with both German and Persian. However, where the goal is improvement of a system for the English-Persian language pair itself, this approach is not useful, since English is already the best-resourced language with Persian, that is, there is no bridge language that will provide improved connection.

The idea of using a bridging language, or triangulated translation as it is also known, in order to deal with low-resource language pairs is relatively new. Certain language pairs involved in machine translation and other NLP applications are so low-resource that they cannot by themselves be used in any feasible SMT system. However, sufficient resources between these and other languages often exist. These 'in between' languages can often be used to bridge the low-resource pair.

Farajian (2011) documents work in the construction of an English-Persian corpus, an open-ended parallel corpus (able to be enlarged by more added data) and able to have other languages added to it. The corpus is aligned semi-automatically on sentence level. Farajian (2011) describes the design of the corpus and the alignment processes used, and identifies character encoding as a challenge when working with corpus construction in Persian. There is a range of Unicode characters set aside for Persian, but certain Arabic characters are also used interchangeably and this usually presents difficulties in NLP work involving the Persian language. An example of this is the letter "ک" "kaf", and "ی" "ye". This is encoded as Persian Unicode (U+06A9 and U+064A), but, alternatively, in Arabic Unicode (U+0643 and U+06CC or U+0649). Another example is in the Persian letter "ه": this is represented as Persian Unicode (U+0647), but can be replaced with the Arabic letter "*Teh Marbuta*", Unicode (U+0629).

As mentioned earlier, normal Persian sentence structure follows subject-object-verb (SOV) order. Although this is normally the case, it is possible for sentences to have relatively free word order (sentence structure). This case is referred to as *scrambled order*. As a language standing alone, this feature enables easy rhyming and ability for phrases and sentences to be put into verse form. However, while this may be the case, scrambled order also presents difficulties in automatic processing (Kiani, Akhavan *et al.*, 2009; K. Megerdoomian, 2000).

There are numbers of different sources that may be used when constructing a parallel corpus. They include literature, movie subtitles, news articles, and Wikipedia articles.

Literary texts can present significant difficulties, as there are often large cultural differences presenting themselves between source and target languages. On top of this, it is normal for literary translations to be translated conceptually which requires a very good knowledge of source and target languages in order to effectively convey the correct emotions and feeling in the target language. These difficulties present themselves largely when alignment at sentence level is attempted (Qasemizadeh, Rahimi *et al.*, 2007).

As documented by M. T. Pilevar and Feili (2010), the use of movie subtitles in construction of a parallel corpus has certain benefits: Movie subtitles are publicly available, in large entries, and alignment is usually a simpler task since the entries are shorter and of similar text length. However the main issue with movie subtitles when used in corpus construction is the informality of the domains. This difficulty is worse for the Persian language in particular, since there is a vast difference between formal and informal text, and there is no mapping table available to be used, nor any software that will perform it. For these reasons and for Persian specifically, the use of movie subtitles as a source for a corpus is not commonly reported, since any system using subtitle-based corpora is usually prone to an unsatisfactory output.

Of all sources, news stories seem to lend themselves best as a source for corpus construction. There are vast numbers of news stories written and translated in a number of different languages, including Persian. In general, they are publicly available from a wide range of sources online. A particular advantage as a source for an English-Persian corpus is the fact that Persian news stories are always written in formal text. Because of this, news sources present themselves as one of the best sources for a number of NLP applications.

Recently, there has been increased interest in the use of news stories as sources for corpora construction. Examples of this work include J. Fry's construction of an English-Japanese parallel corpus based on RSS news feed of English translations of Japanese news articles. Links in the Japanese articles were used to obtain English equivalents (Fry & Center, 2005).

Nadeau and Foster published work in construction of an English-French parallel corpus using Canada Newswire CNW newsfeeds (Nadeau & Foster, 2004).

News publications have also been used in the construction of comparable corpora. Huang, Zhao *et al.* (2010) report development of an English-Chinese comparable corpus based on news stories. Another instance of comparable corpora construction is shown in Baradaran Hashemi, Shakery *et al.* (2010). However, this corpus needs further work in manual and automatic processing in order to be of any use in statistical machine translation.

In a paper addressing techniques for automatic text correction, Kukich (1992) shows that text errors may be classed into five different categories: 1) Isolated errors, or errors relating to spelling mistakes; 2) non-isolated (syntactic) errors; 3) real-word errors, both of which require syntactic and semantic analysis in order to be identified; 4) discourse structure errors; and 5) pragmatic errors, which are not able to be classified as spelling or grammatical.

According to Kies (2008), grammar checkers are unable to check the whole syntactic structure of text, but only deal with subject-verb disagreement and word order errors. Grammar checkers employ a number of NLP-related tasks, such as POS-tagging, tokenization, and matching grammatical rules.

Grammar checking may be classed into three different groups: syntax-based, statistical (corpus) based, and rule-based. In syntax-based approaches, the text is parsed, and if parsing is not successful (i.e., it yields an incorrect parse result) then the grammar of the sentence is deemed incorrect. The rule-based approach involves a list of rules defining errors most likely to be encountered. While this approach has been shown to be reasonably effective, it is time-consuming, and initialization of the system requires significant linguistic knowledge. In their system, Leacock, Chodorow *et al.* (2010) use an SMT-based framework, as the training of a statistical model is helpful in detecting and correcting grammatical errors, more so than a rule-based grammar checker (especially those needing contextual cues for recognition). They show their hybrid approach to a grammar checker to achieve an obtained recall of 0.5

for grammar correction. For grammar error detection, they achieve a score of 0.57, with 0.63 precision. They attribute these results to the merging of the two approaches. The SMT framework is able to correct some of the most probable errors in the text, while augmenting the system with the rule-based procedure is able to correct errors overlooked by the SMT framework.

Davis (2012) presents a transliteration system based on SMT techniques to transliterate text between Tajik and Persian. This work is motivated by the need to make the significantly greater computational linguistic resources of Persian available to Tajik. Transliteration, which differs from translation in that it converts text from one form of writing into another, is normally used to represent foreign words in the script of a local language, or words adopted into a local language where the original language uses a different writing system. Tajik, or Tajik-Persian as it is also known, is a dialect of the Persian language, spoken mainly in Tajikistan. As a spoken language it is very similar to Persian, and fluent speakers of each language may understand each other with little difficulty. However, the written language is another matter, with different writing systems making the two largely incompatible.

Certain differences between the two dialects make transliteration a challenging task. Tajik is written left-to-right with individual letters, in a modified Cyrillic alphabet, whereas Persian is written from right-to-left with connected letters that change form depending on their place in a word. Persian usually omits most vowels in words, as they are implied by the context of a sentence. However, in Tajik, vowel use is normal, as the Cyrillic alphabet possesses a full set of vowels. Despite this, the two dialects share similar syntax and grammar, and, for the most part, use the same word order. As it is uncommon to find usable bilingual corpora for this “language pair”, Davis presents a system that is based mainly on translating lexical representations of morphemes and phonemes of each dialect. This system relies on areas of similarity between the two dialects.

The following are some of the issues that this system must deal with, as listed by the author:

One issue is that Persian can use a number of different letters to represent one sound, but in Tajik, they are mostly represented by one letter. Obviously, this can present issues with alignment of letters. Another covers word “versions”: Persian often has multiple spellings for certain words, whereas, for the most part Tajik will only have

one. Again, this presents difficulties with alignment. This is a problem that is shared by other languages involved with Persian in SMT.

Another issue that must be dealt with concerns the ezafe, a phoneme in Persian which is suffixed to a noun to show modification by another noun, adjective or pronoun. This is normally not shown in Persian, whereas it always appears in Tajik.

The system uses a training corpus of 3503 commonly-used Tajik-Persian words, aligned with GIZA++ (Ney, Nießen *et al.*, 2000), and a language model based on *n*-grams of letters. The corpora used were the Bijankhan corpus on the Persian side, and a manually-constructed Tajik corpus based on data from Asia-Plus, a Tajikistan news website. Decoding was performed using Phramer, a beam search decoder by Olteanu, Davis *et al.* (2006).

The author shows that the system is able to achieve almost 90% transliteration accuracy between Persian and Tajik, and attributes errors in the system to the presence of foreign-origin words in the Tajik vocabulary that are non-existent in Persian, together with place and people's names specific to Tajik.

An example of earlier work in Persian-English SMT is that by Kathol and Zheng (2008). Sponsored by DARPA TransTac program, their surprise challenge was to develop an effective translation system for this language pair in the Persian-to-English translation direction in a 100-day timeframe. Since Persian-English is a low-resource language pair, and given the very short timeframe, the authors were not able to focus on the addition of larger amounts of data in order to improve their system, but instead had to explore other methods of improvement. They identify and apply three things that yielded improvement in their system: use of a hierarchical phrase-based SMT approach, addition of domain-unspecified resources, and application of morphological segmentation. They show an improvement to the baseline system of almost 25% when applying these system modifications. Their goal was to determine methods to significantly improve translation output without extensive linguistic knowledge. Their initial data consisted of 85,400 aligned sentences, and was supplied by DARPA. This initial corpus was then modified and improved to be used in the system. First, paired sentences containing ASR fragments in either language were removed. Then pauses, punctuation and other disjointed words in the text were also removed. Replacement rules such as those removing contractions were run on the English side of the corpus. Similar rules were applied to the Persian side. The authors then used USCPers

transliteration scheme (Ganjavi, Georgiou *et al.*, 2003) to convert Persian text to pure ASCII-based format.

For their decoder, the authors used SRI's SMT decoder. This decoder supports both standard phrase-based (PB) and also hierarchical PB models. The standard PB model runs on a bilingual phrase-pair translation model. While this approach performs well when modelling local word reordering, it encounters difficulties when it comes to long-distance relationships. When involved in language pairs with significantly different word orders (such as Persian-English used here), this can pose trouble for the system. On the other hand, hierarchical PB-SMT systems, based on the extraction of lexical synchronous context-free grammars (SCFGs), are much more effective when it comes to dealing with long-distance relationships. [Kathol and Zheng, 2008] clearly show this fact in their results. The first improvement seen necessary was to employ a hierarchical PB-SMT approach. Using this approach gave a 16.6% improvement in the BLEU score on the initial approach. The authors state that this improvement is due to the word order differences between Persian and English – they are better handled with a hierarchical PB system rather than the standard PB system.

The next improvement they make is the addition of more data to the corpus. They used the corpus developed at New Mexico State University in the Shiraz project (Amtrup, Rad *et al.*, 2000), while it did not provide more data, could result in a 3.26% improved BLEU score.

The final method of improvement explored was the use of unsupervised morphological segmentation. The authors identified the issue with the morphological differences between English and Persian, and the general difficulty with any language pair where this is the case. Morphology can be a challenge to any SMT system: different languages contain different amounts of information in each word. Where one word in one language may describe an entity, situation or action, a paired language may use several words to convey the same meaning. This can specifically pose problems in the alignment process. To help deal with this, the authors employed the Morfessor Categories-MAP algorithm (Creutz & Lagus, 2007) o split certain words into morphological segments, depending on the morphological structure of their equivalents. After testing different perplexity threshold (PPL) settings, they found the greatest improvement was achieved using a PPL setting of 4. Their best system is a combination of a hierarchical phrase-based SMT engine, with the added data of the Shiraz corpus, and the use of morphological segmentation with Morfessor at a

perplexity threshold of 4. This yielded a BLEU score of 0.353 in the Persian-to-English translation direction, which was considered a reasonable output at the time. Kathol and Zheng's main focus was in development and improvement of the Persian-to-English translation direction. However, the other direction was explored, and while the scores were significantly lower (0.216 – 0.225 BLEU), the system modified with the same components (hierarchical PB-SMT, Shiraz corpus addition, etc.) behaved in the same way as the original translation direction, with the modifications yielding an improved output compared to the baseline PB-SMT system.

Mansouri and Faili (2012) compared several different machine translation systems for the Persian-English language pair. Among those covered are baseline SMT systems, factor-based SMT systems, and rule-based MT with a statistical APE component. They propose what they name a "verb-aware SMT" system. This system comprises a hybrid MT model, which uses a rule-based detection module that is run to identify composition verbs in the text, and post-edits the output using an SMT-based APE system. The open-source toolkit Moses was used to train English phrasal verbs and Persian compound verbs, which were identified by the Verb-aware detection module. The system is run on two test sets, PCTS and EGIU. PCTS (Parallel Corpus Test Set) is a test set based on a parallel corpus, while EGIU is based on *English Grammar in Use*, an English educational book presenting structured formal English sentences. BLEU scoring shows that the best results come from the Verb-aware SMT system run on the PCTS test set, which gave a score increase of 2.78% on the baseline SMT system. The authors show that this improvement, for the PCTS corpus at least, is due to the use of a Verb-aware module to identify composition verbs in the text. Where Persian compound verbs and English phrasal verbs both appear on each side of an aligned sentence, the "Verb-aware" SMT system is capable of achieving better results compared to the baseline SMT system. In the case of the EGIU test set, the Verb-aware module failed to detect certain English phrasal verbs that appeared in the test set, and, therefore, the decoder was unable to align equivalent Persian phrases from the training data.

Technology used in the development of automated grammar checkers may also be used to improve translation output quality, in Automatic Post-Editing applications. Ehsan and Faili (2010) present both rule-based and statistical grammar checkers. The rule-based grammar checker outperformed the statistical checker by 0.57 in Precision

and 0.23 in Recall. In the rule-based checker, text is POS-tagged, and a set of rules based on this text is manually developed.

2.6 Existing Machine Translation Tools and Services

There are several existing machine translation tools, mostly available online. The most well-known of these are Google Translate, Bing Translator, Systran, Babel Fish, and Language Weaver.

Google Inc.'s language translation service, Google Translate, is a system based on statistical machine translation. It is currently probably the best-known online language translation service provider, performing hundreds of millions of translations every day. It is able to translate text selections, whole documents, and also web pages between a number of languages. Speech recognition software used in conjunction with its translation engine makes the service able to translate the spoken word also, and release the output as speech using text-to-speech software (Henderson, 2010).

Google Translate's SMT approach originally only supported English and Arabic, and was released in 2006. Until October 2007 the earlier versions of the service used Systran-based software for languages other than Arabic, Russian and Chinese. Currently, it offers full support for translation between 64 different languages, and also partial support for 11 "alpha" languages, which are still in the earlier stages of development (Aiken & Balan, 2011) .

Like many of Google's services, Translate is free for the public to use. Google also incorporates user input into its service, enabling users to contribute a better translation in order to improve the efficiency of the service. Users are also asked to submit alternative words or phrases, where necessary, when dealing with technical terms. Statistics from these user inputs are taken and the system modified to continually update and provide more accurate translation. The ability for a SMT-based system to improve itself with use was one of the main driving reasons for Google's shift to SMT. The benefits of this are enormous – with the public's constant use and contribution, the system is guaranteed to be constantly improving.

Additional features in the system, such as automatic language detection, default English translation, and automatic web page translation in Google Chrome's browser, make Translate an attractive tool to use. Programs and applications using the main system are available on systems such as iOS and Android, and can perform tasks such

as real-time chat translation (using GChat chatbot), and speech-to-speech translation for 14 different languages.

Google Translate's fully supported languages include the following:

Afrikaans, Albanian, Arabic, Belarusian, Bulgarian, Catalan, Chinese (simplified), Chinese (traditional), Croatian, Czech, Danish, Dutch, English, Estonian, Esperanto, Filipino, Finnish, French, Galician, German, Greek, Hebrew, Hindi, Hungarian, Icelandic, Indonesian, Irish, Italian, Japanese, Korean, Latvian, Lithuanian, Macedonian, Malay, Maltese, Norwegian, Persian, Polish, Portuguese, Romanian, Russian, Serbian, Slovak, Slovenian, Spanish, Swahili, Swedish, Thai, Turkish, Ukrainian, Vietnamese, Welsh and Yiddish.

The 11 "alpha" languages still in the earlier stages of development currently include:

Armenian, Azerbaijani, Basque, Georgian, Gujarati, Haitian, Creole, Kannada, Latin, Tamil, Telugu, Urdu. This means that although they are available to use, they produce less reliable translation result than those languages that are fully supported, and not all features are available for those languages (such as speech input/output, and applications using the main system).

The initial SMT system was researched and developed by Franz-Josef Och, the head of Google's machine translation department. According to Och, if developing an SMT system from square one, a bilingual text corpus of over a million words, and two monolingual corpora with over a billion words each, would be needed to form a sound base from which to work. Statistical models are then taken from this data and used to translate between the language pair. Google was able to use United Nations' documents to obtain this immense amount of data, since the same document is usually written in each of the six official UN languages - Arabic, Chinese, English, French, Russian and Spanish. This means that Google Translate now manages a huge multi-lingual corpus of twenty billion words for these languages.

Microsoft also provides a translation service enabling users to translate selections of text and even whole web pages into other supported languages. Bing Translator was previously called Windows Live Translator, and used Systran as its backend translation software. Microsoft Research has now developed its own translation software, called Microsoft Translation, which powers the language pairs currently offered by the service. Where computer-related translation is required (including technical computer terms), Microsoft uses its own syntax-based SMT technology.

Systran is a machine translation company founded in 1968 by Dr Peter Toma. One of the longest-standing machine translation companies, Systran has performed a significant amount of work for the United States Department of Defence. Translation services Yahoo, Babel Fish and AOL use Systran as the software base for their systems. Apple Mac's OS X operating system uses Systran in its Dashboard Translation widget.

Systran employs a sentence-by-sentence approach to translation, focusing on and processing individual words and their dictionary definitions before parsing the sentence to generate a translation output. The three main groups of modules composing Systran's framework are: Dictionary, Systems Software, and Linguistic Software. These groups work together to create an automatic machine translation system (Senellart, Dienes *et al.*, 2001).

Babel Fish is a web-based automatic machine translation program built by AltaVista and used by Yahoo. It is comically named after the fictitious translating animal from Douglas Adam's book *The Hitchhiker's Guide to the Galaxy*. Babel Fish uses Systran's translation system as a software base, and can translate among English, Simplified and Traditional Chinese, Dutch, French, German, Greek, Italian, Japanese, Korean, Portuguese, Russian, Swedish and Spanish .

The translations provided by Babel Fish are not as reliable as those given by other translation services, and Babel Fish is considered to be only a minor contributor to the language translation industry.

In 2002, Kevin Knight and Daniel Marcu of the University of Southern California founded Language Weaver (now known as SDL Language Weaver), a company commercializing a statistical approach to language translation and spoken language processing.

The software systems used by SDL Language Weaver give an example of slightly more recent progress in the statistical approach to machine translation. It implements learning algorithms to obtain statistical models from bilingual corpora. Since these models originate from pre-existing aligned language pairs, the output is statistically more likely to be accurate (Soricut, Bach *et al.*, 2012).

Another feature of Language Weaver is its ability to be customised to translate technical material. The software's learning capabilities aid it in specialising in different subjects or styles.

Language Weaver has incorporated the product of recent progress in statistical machine translation systems and, with some degree of success, is now able to create translation systems for language pairs that have limited amounts of bilingual text.

Language Weaver currently offers translation for English to and from French, Italian, German, Greek, Danish, Spanish, Dutch, Portuguese, Swedish, Russian, Czech, Romanian, Polish, Arabic, Persian, Simplified and Traditional Chinese, Korean and Hindi. It also offers Arabic/Spanish, Arabic/French, Spanish/French, and French/German.

Although Language Weaver currently translates using phrase-based SMT, their researchers are currently studying how to incorporate syntax-based statistical machine translation, as this approach can be used to improve translation quality for certain language pairs.

Though its main service area is in machine language translation, Language Weaver also offers several other service products, such as Alignment Tool, and Customiser. Alignment tool is a translation memory generator, and takes an input of a translated document, aligns it at segment level, and saves a translation memory file. Customiser is a tool which aids in fine-tuning machine translation output, helping to specify translation to a narrow domain.

2.7 Online vs. Installable Software

With machine translation, there is the option of using online software or computer-based software. Web-based machine translation systems and installable software differ in the kind of service they can offer. While installable software is flexible in that it can be customised and trained on specific data, online systems are limited to the domain they have been trained on, and cannot be trained on new data. Because of this, web-based systems will always be limited in the accuracy they can provide, since they cannot be customised for specific use; they are largely general purpose systems. Another issue with web-based machine translation is the user's vulnerability when translating sensitive or private documents. People take a risk with the submission of any sensitive material in any system online. Using a computer-based system is much more secure.

2.8 Persian Language

The Persian language is also known as Farsi. Some believe the two to be different, and terms such as “Farsi Iran” and “Afghan Farsi” have arisen. Still others refer to the Persian as spoken in Iran as “Western Farsi”, and to Dari (widely spoken in Afghanistan) as “Eastern Farsi”. However, to be precise, the correct formal name of the language is Persian, but the name “Farsi” is commonly used to refer to the same language.

Persian is an Indo-European language, spoken mostly in Iran, but also in parts of Afghanistan, India, Tajikistan, the United Arab Emirates, and also in large communities in the United States. Worldwide there are approximately 60-110 million people who speak Persian as a first language (Windfuhr, 2009). Figure 2-1 shows the Persian language position in the Indo-European language tree (Short, 2008).

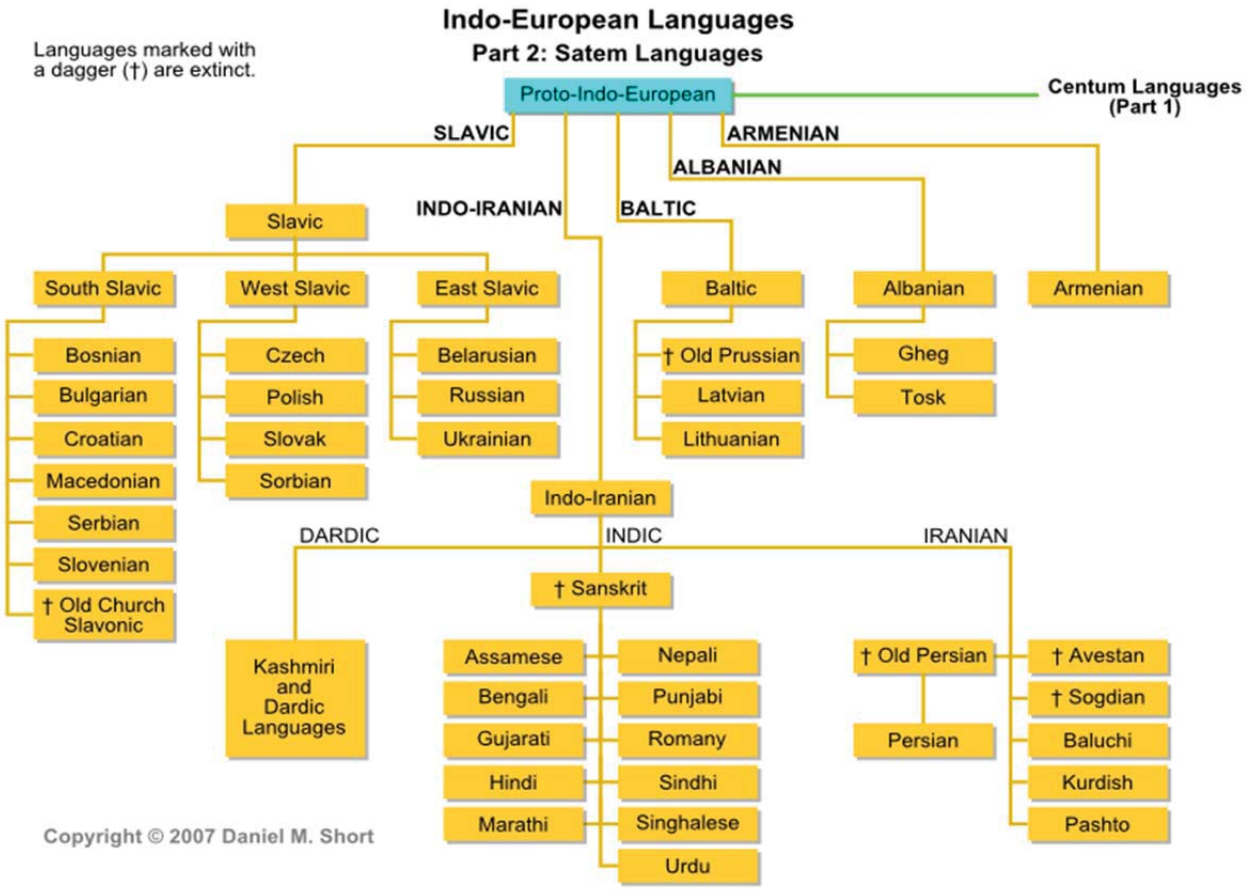


Figure 2-1: Indo-European Languages

Persian is among the top-affluence languages of the world, as ranked by GLP (Gross Language Product – total market value of goods and services produced by speakers of that language per year). Persian is ranked at 43rd place (Figure 2-2, Hammarström, 2009).

GLP is calculated by:

$$GLP(L) = \sum CSC(L) \times GNPperCapita(C) \qquad \textbf{2.1}$$

#	Language	iso-639-3	GLP	Total Pop.	GNP-per-capita
1	English	eng	14 570 119 604 622	326 959 888	44 562.4
2	Japanese	jpn	4 162 642 000 000	121 000 000	34 402.0
3	Spanish	spa	4 117 723 821 500	327 380 860	12 577.7
4	German, Standard	deu	3 408 803 154 660	84 959 210	40 122.8
5	Arabic, Standard	arb	2 807 780 000 000	206 000 000	13 630.0
6	French	fra	2 775 416 019 700	67 661 960	41 018.8
7	Chinese, Mandarin	cmn	2 146 387 252 580	845 033 030	2 540.0
8	Italian	ita	2 061 118 900 800	56 638 620	36 390.6
9	Russian	rus	1 809 937 945 460	125 102 940	14 467.5
10	Portuguese	por	1 345 089 888 980	174 307 980	7 716.7
11	Dutch	nld	966 883 126 740	21 309 290	45 373.7
12	Korean	kor	885 367 820 000	64 739 000	13 675.9
13	Bavarian	bar	570 287 492 000	13 259 000	43 011.3
14	Turkish	tur	445 343 958 400	47 777 700	9 321.1
15	Swedish	swe	412 626 274 000	8 206 000	50 283.4
16	Polish	pol	412 358 750 120	36 998 360	11 145.3
17	Catalan-Valencian-Balear	cat	404 298 076 000	11 351 000	35 617.8
18	Norwegian	nor	392 520 800 000	4 640 000	84 595.0
19	Chinese, Min Nan	nan	382 848 478 700	46 915 100	8 160.4
20	German, Swiss	gsw	337 411 118 000	6 469 000	52 158.1
21	Lombard	lmo	336 656 463 000	9 133 000	36 861.5
22	Greek	ell	334 090 819 150	11 526 360	28 984.9
23	Danish	dan	312 161 238 200	5 478 830	56 975.8
24	Vlaams	vls	267 949 458 000	6 132 000	43 696.9
25	Napoletano-Calabrese	nap	255 217 050 000	7 050 000	36 201.0
26	Finnish	fin	230 401 980 000	4 934 100	46 695.8
27	Mainfränkisch	vmf	197 946 650 000	4 910 000	40 315.0
28	Chinese, Wu	wuu	189 918 445 380	77 201 820	2 460.0
29	Hindi	hin	177 285 437 200	180 469 200	982.3
30	Sicilian	scn	174 850 830 000	4 830 000	36 201.0
31	Czech	ces	158 690 740 000	9 290 000	17 081.8
32	Javanese	jav	157 822 354 630	84 600 970	1 865.4
33	Romanian	ron	157 093 940 480	23 118 480	6 795.1
34	Hungarian	hun	156 132 196 260	12 253 140	12 742.2
35	Chinese, Yue	yue	148 402 526 750	54 471 530	2 724.4
36	Bengali	ben	121 632 915 000	180 624 200	673.4
37	Chinese, Hakka	hak	116 456 948 110	29 976 560	3 884.9
38	Galician	glg	113 090 635 000	3 185 000	35 507.2
39	Piemontese	pms	112 585 110 000	3 110 000	36 201.0
40	Hebrew	heb	112 330 850 000	4 850 000	23 161.0
41	Arabic, Najdi Spoken	ars	112 117 150 000	9 450 000	11 864.2
42	Chinese, Jinyu	cjy	110 655 000 000	45 000 000	2 459.0
43	Farsi, Western	pes	109 305 988 000	22 455 000	4 867.7

Figure 2-2: The Top-Affluence Languages of the World

The Persian language has evolved over three main periods of time. The language is classified with respect to each time period as Old, Middle and New Persian. Old Persian refers to the language used by the Achaemenians, from 650–350 B.C. During the Parthian period, circa 350 B.C.–230 A.D., right through to the Sassanian period, circa 230 A.D.–650 A.D., the language is classified as Middle Persian. This begins to develop into what is classified as New Persian after 650 A.D., around the time when Iran was invaded by the Arabian armies. New Persian itself is further classified into two different eras, Classical and Modern Persian, and although the exact boundary between these two is uncertain, it is known that approximately 300 years passed before Modern Persian as it is classified was widely used as a lingua franca between the two languages.

Persian, like other languages, is still constantly changing with use. Change can come about for a number of reasons, but the main cause is younger generations speaking slightly differently compared to older generations, with certain areas of vocabulary changing. Influences of other countries, interaction between people, and also the internet age are other factors which have caused changes.

A number of difficulties are encountered when the Persian language is used in a SMT system. First, SMT of the Persian language is only recently being exploited, so there is not a great deal of information available on prior work (and, therefore, the challenges and difficulties encountered) by other researchers. In the task itself, probably the largest difficulty encountered is the fact that there is very limited digital data available in the form of bilingual corpora.

The best language to pair with Persian for MT is English, since the English language is best supported by resources such as large corpora, language processing tools, and syntactic tree banks, not to mention it is the most widely-used language online, and in the electronic world in general.

When compared to English, however, Persian has many differing characteristics. There are several grammatical characteristics in written Persian which differ from English. There is no use of articles in Persian, as the context shows where these would be present. There are no upper or lower case letters, and symbols and abbreviations are rarely used. There is no gender system or tones in the language. It has inflectional morphology, and inflectional synthesis of verbs – usually four to five categories per word (Haspelmath & Bibiko, 2005).

The subject in a Persian sentence is not always placed at the beginning of the sentence as a separate word. Instead, it is denoted by the ending of the verb in that sentence. Adverbs are usually found before verbs, but may also appear in other locations in the sentence. In the case of adjectives, these usually follow the nouns they modify, unlike English where they are usually found before the nouns. Persian is a morphologically rich language, with many characteristics not shared by other languages (K Megerdoomian, 2000). This can present some complications when it is involved with translation into *any* other language, not only English. Compared to English, the basic sentence structure is generally different in terms of syntax. In English, we usually find sentence structure in its most basic form following the pattern of “subject – verb – object” (SVO), whereas in Persian it is usually “subject – object – verb” (SOV).

Secondly, spoken Persian differs significantly from its written form, being heavily colloquial, to a much greater degree than English. Thirdly, many Persian words are spelled in a number of different ways, yet all are correct. This, in particular, poses difficulty for translation, since if one version of the spelling is not found in a bilingual corpus, such a word may be incorrectly translated, or remain as an OOV (out of vocabulary) word. Any SMT system designed for this language pair needs to take these details into consideration, and the specifics of the system developed to cater for these differences.

Many languages of the world, like English and Persian, are alphabetic in the sense that they represent their vowels and consonants in the form of letters in their orthography. In these languages, words are composed of one or more syllables.

When translating into this language, it should be noted that most of the English sentence structure cannot be preserved due to the huge difference between the two languages in terms of syntax. It is imperative on the part of the translator to deliver a fluent and natural-sounding translation as opposed to a literal copy of the original with less focus on the meaning.

Many Persian words are spelled differently, and yet all of them are correct. For instance the terms "میشود" and " می‌شود" are different spellings of the translation of the same word "to be", and assume "می" is "-ing" in Persian so many words in Persian can be spelled in three different variations. Some translators prefer one variation, and when proofreading a job, consider the other forms incorrect.

Persian is also a language that has adopted many Arabic words. Some translators attempt to invent new words or use more “Persian" words which are less common instead of accepting the Arabic words as part of the language. This can make the language vague and give readers trouble understanding the material.

2.9 Characteristics of the Persian Language

The most common sentence structures in Persian (compared to English) are the following:

Persian:	Subject	+	intransitive verb
English:	Subject	+	verb

Persian:	Subject	+	object	+	transitive verb
English:	Subject	+	verb	+	object

Often, in both formal and informal Persian, the subject does not appear at the beginning of the sentence as an individual word; instead, it is a pronoun attached to the verb. In other words, the subject appears as a part of the verb.

Persian:	Subject	+	subject complement	+	linking verb
English:	Subject	+	LV	+	subject complement

In Persian, adverbs are normally used before verbs, but can be placed in other locations within the sentence as well. Adjectives are almost always used after the nouns they modify. When working with a language that has an entirely different structure, its speakers have drastically different cultures from native English-speakers, and therefore the task is much harder. An example of this is where translation or localization mistakes have occurred with Persian, such as problems with text expansion, date/time formats, counting errors, character encoding, or mistakes with the translation itself.

2.10 Persian Alphabet and Pronunciation

The written Persian language uses an extended Arabic alphabet, and is written from right to left. There are numerous different regional dialects of the language in Iran, however, nearly all writing is in standard formal Persian.

There are 32 characters in the Persian alphabet. Vowels are not separate letters, but rather are written with diacritics and/or combinations of consonant letters. These vowels are not always indicated in Persian text. There are seven vowel sounds: â (/ɒ:/), a (/æ/), e (/e/), i (/i:/), o (/o/), u (/u:/), ow (/ou/). The "alef" has no particular sound, and can denote "â" (ٰا), "a" (ا), "e" (اِ), "o" (اُ) at the beginning of words by means of diacritics, but elsewhere it always denotes "â". However, usually only the diacritic of "â" (ٰا) is written and the pronunciation must just be memorized, for example: آب (âb) – water, اسب (asb) – horse, امید(omid) – hope, امشب (emšab) – tonight. The Persian alphabet and numbers are shown in Appendix I, sections 1 and 2.

2.11 Persian Corpora

The construction of modern human language corpora is essential for the development of a number of different research areas. Without large, good-quality corpora, tasks such as MT are impossible.

For optimum operation, an SMT language model requires a significant amount of data that must be trained in order to obtain proper probabilities. Parallel corpora are required to be balanced, as well as being large in size. All statistical translation models are based on the idea of word alignment and are trained using a large parallel corpus. Obtaining this parallel corpus is one of the most important, and often challenging, steps in the development of an SMT system. The difficulty of this task is amplified somewhat when dealing with low resource languages such as Persian.

Because the Persian language is rich with prefix and suffix morphology, there are large differences between Persian and English in terms of utterance length and observed unique words. This means that compared to many other language pairs, much more parallel data is required in order to learn translations accurately, and great difficulty is encountered when it comes to alignment. This, coupled with the relatively small amount of parallel data available for the task at hand, presents a significant challenge for the production of quality translations.

Experimentations with the Persian language have been quite recent and are limited when compared to work with other languages. Most researchers in NLP and IR construct their own databases which are typically small, collected manually, and are not investigated for quality or balance. Because of this it is unclear how well experimental findings would compare.

2.12 Available Persian Text Corpora

There are several Persian corpora available: Bijankhan, Hamshahri, TMC, and TEP.

2.12.1 Bijankhan corpus

Bijankhan corpus is a tagged corpus derived from text gathered from daily news. Originally it was developed in the Faculty of Literature and Human Science at the University of Tehran (Bijankhan, 2004). Later, the Database Research Group lab at

the University of Tehran prepared it for the automatic learning process (F. Oroumchian, S. Tasharofi *et al.*, 2006). The collection is categorized into sections and subsections such as cultural, scientific, political, literature (poetry), and so on, to make a total of approximately 4300 sections. The original version of Bijankhan had a tag set of 550 Persian POS tags. The processed version trained for automatic learning consists of about 2.6 million manually tagged words, using a tag set of 40 POS tags. There are 76,707 distinct words in total. This corpus is in Unicode text format, and is suitable for NLP research in Persian.

2.12.2 Hamshahri Corpus

Hamshahri corpus was constructed at the Database Research Group lab at the University of Tehran and is based on collections of articles derived from the Hamshahri daily newspaper in Iran, one of the most popular daily newspapers there, which has been in publication for over 20 years. Document categories cover politics, city news, economics, reports, editorials, literature, sciences, society, foreign news, sports, etc. The size of each document varies from short news (under 1 KB) to rather long articles (e.g., 140 KB), with an average size of 1.8 KB.

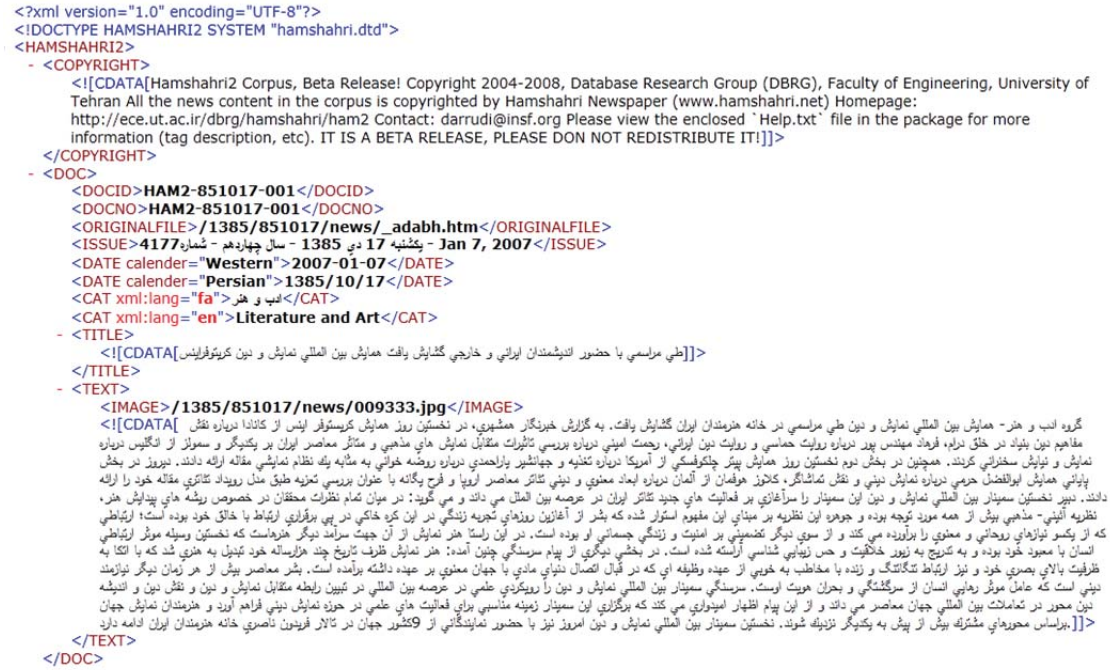


Figure 2-3: Hamshahri Corpus Version 1 sample

There are two versions of Hamshahri corpus, both in Unicode CLEF XML format. Version 1 is 700MB in size, and contains 160,000 Hamshahri news documents from

1996 to 2003. A sample of this corpus is shown in Figure 2-3. It consists of two sets of queries and judgments created at CLEF2008 and CLEF2009 for the evaluation of ad hoc information retrieval systems. Version 2 is twice as large, being 1400MB and consisting of 318,000 documents from 1996 to 2007. It was created with UTIRE (University of Tehran Information Retrieval Evaluation system) in 2009. This version also has links to pictures and web pages, making it suitable for some image retrieval tasks[1].

2.12.3 Shiraz Corpus

The Shiraz project (Zajac, Helmreich *et al.*, 2000) documents a first attempt at development of a Persian-English corpus. The authors constructed their parallel corpus by obtaining 3000 sentences in Persian from a monolingual Persian corpus of online material. After manual translation into English at New Mexico State University, it was used in testing the Shiraz system. This project was partially sponsored by DARPA (Defense Advance Research Project Agency), as the work was intended for use by the US Army, as well as for medical applications. Other sources of data used in this work originated from corpora for other language pairs, such as English-Iraqi, and also medical glossaries. Such sources need to be manually translated and aligned (Bach, Eck *et al.*, 2007; Belvin, May *et al.*, 2004; Ettelaie, Gandhe *et al.*, 2005; Georgiou, Sethy *et al.*, 2006).

2.12.4 MULTEXT-East Framework

Qasemizadeh et al. worked on the construction of a parallel corpus for Persian. This work used MULTEXT-East framework, and was based on Orwell's "1984" text for corpus construction. The resulting corpus consisted of 6,606 sentences, with approximately 110,000 tokens (Qasemizadeh, Rahimi *et al.*, 2007).

2.12.5 TEP – Tehran English-Persian Corpus (Parallel)

TEP corpus was created at the Natural Language and Text Processing lab at the University of Tehran in 2010. This corpus is based on extracted movie subtitles covering informal and conversational domains. It consists of 1600 aligned movie subtitles, with 613,000 bilingual sentences. The corpus comprises 4 million words (M. Pilevar, Faili *et al.*, 2011).

[1] http://ece.ut.ac.ir/dbrg/hamshahri/index.html

2.12.6 PEN: Parallel English-Persian News Corpus

PEN corpus was developed from 57,000 news documents that were available online. The result was a corpus with 30,000 sentence pairs. The documents on which the corpus was based cover a wide variety of news domains such as sport, politics, interviews, etc. Pairs of documents were pre-processed before being broken into sentences and aligned on sentence level with two similarity measures. Google Translate was used to verify alignment of sentence pairs (Farajian, 2011).

2.12.7 TMC – Tehran Monolingual Corpus

At 250 million words, TMC is the largest freely available monolingual corpus for Persian; it has 300,000 unique words of frequency > 1, and is suitable for language modelling[2].

2.12.8 ELRA

ELRA (European Language Resource Association) built a commercially-available parallel corpus, comprising 3,500,000 Persian-English words aligned at sentence level. This is a mixed-domain corpus, covering a number of different areas such as art, culture, idioms, law, literature, medicine, poetry, politics, proverbs, religion and science. The entire corpus consists of about 100,000 sentences over 5,021 entries[3].

2.12.9 Other Corpora

For European languages, the Europarl corpus has become somewhat standard for experimentation. Unfortunately, there is no similar resource available for Persian. Some examples of current corpora implemented in MT are the 25 MB corpora used by Taghiyareh, Darrudi *et al.* (2003) which was based on Iranian parliamentary laws and regulations; another is the FLDB (Farsi Linguistic Database) corpus (Assi, 1997), which is a well-structured and modern corpus of approximately 3 million words consisting of word lists, dictionary entries, and samples of both informal and formal spoken language. Despite the obvious advantages of its structure, and how new it is, it is still not large enough to be used for extensive IR tasks.

[2] http://ece.ut.ac.ir/nlp/resources.html

[3] http://catalog.elra.info/product_info.php?products_id=1111

In some of our experiments, we implemented several Persian monolingual corpora, which were concatenations of three different news sources – Hamshahri, IRNA, and BBC Persian (based on the parallel corpus developed in-house). IRNA has almost 5.6 million sentences, and the BBC corpus contained 7,005.

2.13 Summary

In summary, the advantages of machine translation over conventional human translators are becoming more numerous with research advances in NLP. MT, though by no means a new concept, has seen significant development in the past two decades, with the most popular approach now tending towards SMT because of the numerous advantages this approach holds over others. There are several open-source decoders that can be used for SMT, such as Moses and Joshua (Z. Li, C. Callison-Burch *et al.*, 2009). Individual differences in these decoders have certain effects on the system and its output as a whole, and will be covered in subsequent chapters. The literature review at the commencement of this study showed no other documented work in English/Persian SMT, and only several works have been reported, therefore there is potential to improve the performance.

There are a number of existing translation systems available, although not many that support Persian, and fewer still that are able to give any deal of output accuracy. Even prominent online systems such as Google Translate are still unable to provide satisfactory translation output for English/Persian. Many of the works reviewed show in-depth development of various aspects of MT systems, and useful advancements which, as shown, do improve the results somewhat. However, for the most part, the output reported in these works still does not provide English/Persian translation to a satisfactory degree from a language translation system.

The Persian language is a complex language with a long history. It possesses a number of characteristics which make it a significantly difficult language to pair with English in any machine translation approach. These characteristics and differences include syntax, morphology, word order, and great differences between formal and informal language.

There is a significant shortage of digital text for the Persian language, and, in the case of bilingual corpora specifically when paired with English, this is perhaps due to the difficulties encountered with aligning sentences and words. Despite this, there are

several mono- and bilingual corpora available such as Hamshahri, Shiraz, TEP, PEN, and ELRA, some of which were used in tests of this project.

Chapter 3. Statistical Machine Translation and Evaluation Metrics

"If you can't explain it simply, you don't understand it well enough."

~ Albert Einstein

3.1 Introduction

This chapter gives an overview of the statistical machine translation approach, with details of the noisy channel and log-linear models. Corpus alignment is discussed, together with how a training model and language model are constructed. A complete baseline decoding process is examined, showing the use of the training and language models. Finally, the automatic evaluation metrics used throughout this project, such as BLEU, NIST and TER are detailed, and their scoring methods are compared.

3.2 SMT Overview

Statistical Machine Translation (SMT) involves two individual processes known as training and decoding. In the training process, a statistical translation model is extracted from an aligned parallel corpus, and another separate statistical model is extracted from a monolingual corpus in the target language (PF Brown, Della Pietra *et al.*, 1990; PF Brown, Della Pietra *et al.*, 1993).

The decoding process is that which generates the translation. The input, a phrase, sentence or sentences, is passed to the decoder, which searches through all the possible translations of the input produced by the translation model. The translation with the highest probability produced by the language and translation models, is then designated as the most likely correct translation, and is output in the target language.

At a high level, SMT gives a view of MT expressed in a single formula. From this vantage point, how translations are generated is irrelevant. The only notable issue is that given the input string, it can be determined how likely any proposed translation is, and that consequently it is possible to determine the most probable (i.e., 'best', according to the system) translation from a set of proposed candidates.

3.3 Bayes Decision Rule

3.3.1 Noisy-Channel Model

In statistical machine translation, a source language string $f_1^J = f_1 \ldots f_j \ldots f_J$ is given as input, which is to be translated into a target language string $e_1^I = e_1 \ldots e_i \ldots e_I$.

Statistical decision theory tells us that among all possible target language sentences, we should choose the sentence which minimizes the expected loss (Duda, Hart *et al.*, 1976):

$$\hat{e}_1^{\hat{I}} = \underset{I,\, e_1^I}{\operatorname{argmin}} \left\{ \sum_{I', e'^{I'}_1} Pr(e'^{I'}_1 \mid f_1^J) \times L\left(e_1^I, e'^{I'}_1\right) \right\} \qquad \textbf{3.1}$$

This is the Bayes decision rule for statistical machine translation. Here,

$$L_{0-1}\left(e_1^I, e'^{I'}_1\right) = \begin{cases} 0 & if \;\; e_1^I = e'^{I'}_1 \\ 1 & else \end{cases} \qquad \textbf{3.2}$$
$$= 1 - \delta\left(e_1^I, e'^{I'}_1\right)$$

$L\left(e_1^I, e'^{I'}_1\right)$ denotes the loss function under consideration. It measures the loss (or errors) of a candidate translation e_1^I assuming the correct translation is $e'^{I'}_1$. $\Pr(e'^{I'}_1 \mid f_1^J)$ denotes the posterior probability distribution over all target language sentences e_1^I given the specific source sentence f_1^J. Note that the Bayes decision rule explicitly depends on the loss function $L\left(e_1^I, e'^{I'}_1\right)$. In case we want to minimize the sentence or string error rate, the corresponding loss function is:

Here, equation 3.2 denotes the Kronecker-function. This loss function is called 0-1 loss as it assigns a loss of zero to the correct solution and a loss of 1 otherwise. Using the 0-1 loss, Bayes decision can be simplified to:

$$\hat{e}_1^{\hat{I}} = \arg\max_{I, e_1^I} \{ Pr(e_1^I \mid f_1^J) \} \qquad \textbf{3.3}$$

This decision rule is also called the maximum a-posteriori (MAP) decision rule. Thus, we select the hypothesis which maximizes the posterior probability $\Pr(e_1^I \mid f_1^J)$.

It is noteworthy that virtually all MT systems use the MAP decision rule although they are usually not evaluated using the 0-1 loss function. The most common evaluation metric nowadays is the BLEU score (K. Papineni, Roukos *et al.*, 2002),

which is also used in many other evaluation metrics such as NIST, TC-Star, and IWSLT. This results in a mismatch between the decision rule that is used to generate a translation hypothesis and the loss function that is used to evaluate it. Kumar, Och *et al.* (2007) presented a BLEU induced Bayes risk decoder and reported performance gains. A similar approach was taken in (Zens & Ney, 2007; Zollmann & Venugopal, 2006).

In the original work on statistical machine translation (PF Brown, Della Pietra *et al.*, 1990), the posterior probability was decomposed:

$$Pr(e_1^I | f_1^J) = \frac{Pr(f_1^J | e_1^I)}{P(f_1^J)} \quad \textbf{3.4}$$

Note that the denominator P (f_1^J) depends only on the source sentence f_1^J and, in case of the MAP decision rule, can be omitted during the search:

$$\hat{e}_1^{\hat{I}} = \arg\max_{I,e_1^I} \{ Pr(e_1^I) \times Pr(f_1^J | e_1^I) \} \quad \textbf{3.5}$$

This is the *noisy-channel model*, the so-called fundamental equation of statistical machine translation (PF Brown, Della Pietra *et al.*, 1993). The decomposition into two knowledge sources is known as the noisy-channel approach to SMT (PF Brown, Della Pietra *et al.*, 1990). The noisy channel model is a more traditionally-used model, but has been largely replaced with the log-linear model, as it has been shown to be advantageous over the noisy-channel model in a number of areas. In the noisy-channel model, there are two feature scores: $Pr(f_1^J | e_1^I)$ and $Pr(e_1^I)$. $Pr(f_1^J | e_1^I)$ is referred to as the translation model, and represents the probability of source sentence *f* and target translation *e* being linguistically equivalent. The feature $Pr(e_1^I)$ is known as the language model, and represents the probability of translation *e* being a valid sentence in the target language. The two features $Pr(f_1^J | e_1^I)$ and $Pr(e_1^I)$ are multiplied together. The noisy-channel model allows an independent modelling of the target language model $Pr(e_1^I)$ and the translation model $Pr(f_1^J | e_1^I)$. The target language model $Pr(e_1^I)$ describes the well-formed target language sentence. The

translation model $Pr(f_1^J \mid e_1^I)$ links the source language sentence to the target language sentence. The translation model score, generally based on lexical correspondences, shows how well the meaning of the source sentence is captured in the translation. The language model score is based on frequency of occurrence of substrings in a monolingual corpus of the target language, and is independent of whether the original meaning of the source language sample is captured. The score simply shows the likelihood of the translation being a valid sentence in the target language. In general, the translation model score has twice the influence on the final score than that of the language model, since it is a more important parameter. The final score is a combination of the translation model score and the language model score, and represents the best combination of scores to give the optimum target sentence.

3.3.2 Log-linear Model

The log linear model differs from the noisy-channel model in that it is able to express scoring based on an unlimited number of features. In this way, it can be described as a more general model. Log probabilities are used by converting standard probabilities with the log function and adding them together, rather than multiplying, following standard logarithmic rules (i.e. *log (A·B) = log (A) + log (B))*. The log-linear model can be derived by the direct modelling of the posterior probability $Pr(e_1^I \mid f_1^J)$. Using a log-linear model was proposed in (Och & Ney, 2002; K. A. Papineni, Roukos *et al.*, 1998).

$$Pr(e_1^I \mid f_1^J) = p_{\lambda_1}^M(e_1^I \mid f_1^J) \qquad \textbf{3.6}$$

$$p_{\lambda_1}^M(e_1^I \mid f_1^J) = \frac{exp\left(\sum_{m=1}^M \lambda_m h_m\left(e_1^I, f_1^J\right)\right)}{\sum_{e'_1{}^{I'}} exp\left(\sum_{m=1}^M \lambda_m h_m\left(e'_1{}^{I'}, f_1^J\right)\right)} \qquad \textbf{3.7}$$

Here, we have models and model scaling factors . Again, the denominator represents a normalization factor that depends only on the source sentence f_1^J. Therefore, we can omit it during the search process in case of the MAP decision rule. The result is a linear combination of the individual models $h\left(e_1^I, e'_1{}^{I'}\right)$:

$$\hat{e}_1^{\hat{I}} = arg\max_{I,e_1^I}\{\ Pr(e_1^I \mid f_1^J)\}$$

$$= arg\,max_{I,e_1^I}\left\{\frac{exp\left(\sum_{m=1}^{M}\lambda_m h_m\left(e_1^I, f_1^J\right)\right)}{\sum_{e'_1{}^{I'} exp\left(\sum_{m=1}^{M}\lambda_m h_m\left(e'_1{}^{I'}, f_1^J\right)\right)}\right\} \quad \textbf{3.8}$$

$$= arg\max_{I,e_1^I}\left\{\sum_{m=1}^{M}\lambda_m h_m\left(e_1^I, f_1^J\right)\right\} \quad \textbf{3.9}$$

where equation (3.9) is the final form of the log-linear model. In equation (3.9), *M* denotes the number of features to be added, and individual scoring is undertaken by multiplying λ_m and $h_m\left(e_1^I, f_1^J\right)$, λ_m being an importance-indicating weight, and $h_m\left(e_1^I, f_1^J\right)$ the assigned log probability of the source sample and target translation's linguistic equivalence. Thus, the noisy-channel model can be expressed exactly in the log-linear model by manipulating the features used in the model, or, in other words, the log-linear model as shown in (3.9) is merely a general solution expressed in the noisy-channel approach.

The log-linear model is superior to the noisy-channel model in that the importance of the features in the model can be adjusted in order to control the influence each feature has on the overall output. This is done, for instance, by controlling the values of λ_m and $h_m\left(e_1^I, f_1^J\right)$. The model scaling factors $\lambda_1^M 1$ are trained according to the maximum class posterior criterion, for example, using the GIS algorithm (Och & Ney, 2003). More features may be added to the model and the λ_m and $h_m\left(e_1^I, f_1^J\right)$ values defined to suit the particular features function within the model, such as modifying the level of operation (in terms of tokens) of either the translation or language model. Alternatively, these can be trained with respect to the final translation quality measured by an error criterion (Och, 2003). This is the so called minimum error rate training (MERT).

Because of its superiority and adaptability to different systems, the log-linear model was used in this system's development.

3.4 Translation Model

$Pr(e_1^I \mid f_1^J)$, referred to as the translation model, represents the probability of source sentence *f* and target translation *e* being linguistically equivalent, or in other

words, that the meaning of the source sentence *f* is accurately represented by the target sentence *e*. The translation consists of a model of the source-target training corpus (aligned on sentence level), and an algorithm used to calculate *f* and *e* equivalence. Table 3-1 shows examples of short English phrases associated with Persian phrases, and in each example the phrase pair is associated with a scored probability.

Table 3-1: English-Persian Probability Example

I need	من باید	0.1
I	من	0.7
To return	برگردم	0.05
Tomorrow	فردا	0.4
Return tomorrow by	فردا برگردم	0.0001

Figure 3-1 below shows how the sample sentence is broken down:

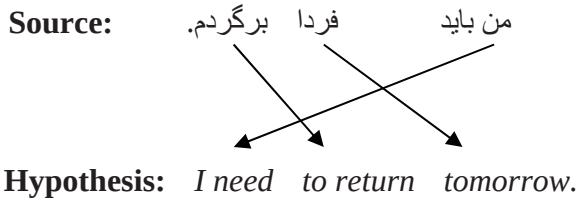


Figure 3-1: Example of Persian-English alignment (1)

The probability scoring of the sentence is shown in the equation below. Each term's probability (from Table 3-1) is multiplied in the equation to give the final score.

$$\Pr(\textit{Source} \mid \textit{Hypothesis}) = \Pr(\text{من باید} \mid \textit{I need}) \bullet \Pr(\text{برگردم} \mid \textit{to return}) \bullet$$

$$\Pr(\text{فردا} \mid \textit{tomorrow}) = 0.012$$

However, it is possible to reach a different probability score depending on the way the sample sentence is divided into phrases. Compare Figures 3-1 and 3-2, which show the different ways the sample sentence may be divided.

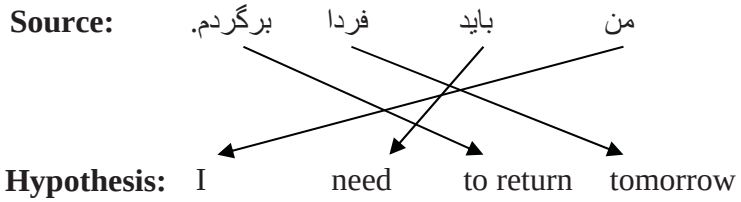


Figure 3-2: Example of Persian-English alignment (2)

Multiplication of the individual probability scores now gives the following:

Pr (*Source* |*Hypothesis*) = Pr (من | *I*) • Pr (باید | *need*) • Pr (برگردم | *to return*) • Pr (فردا | *tomorrow*) = 0.0042

The first step required to extract a translation model from a parallel corpus involves word-aligning the data using GIZA++ and extending those alignments to cover phrases. Phrase pairs are then extracted to give phrase lengths of 1 to *n* words, where *n* is chosen as a maximum such that the system is presented with phrases that are actually feasible to work with. In many cases the number of words in each aligned phrase may be different between the source and target language, depending on how each language represents the meaning of the phrase.

3.5 Training Model

The two main resources on which SMT relies are its parallel and monolingual corpora. The monolingual corpus, in the target language, is used in generating a language model, while the parallel corpus is needed to generate the training model, which is searched by the translation model $Pr(e_1^I | f_1^J)$ for aligned phrases and sentences, depending on the level of alignment. The parallel and monolingual corpora can be collectively referred to as training data. After the training process, the corpora themselves are no longer required for any further process.

3.6 Parallel Corpus Alignment

3.6.1 Word Alignment

In short, the process of word alignment refers to linking words, phrases or sentences of equivalence between the two sides of a parallel corpus. A parallel corpus must be aligned before a training model (which is based on the parallel corpus) can be generated. Alignment is generally classified by the level it is performed. For example, a parallel corpus aligned on sentence level refers to the alignment of sentences. The number of phrases and words may be different between the two languages, but the sentences themselves are linguistically equivalent. Alignment on phrase level refers to equating phrases, and word level to equating words.

Figure 3-3 gives an example of alignment on word level, showing paths of equivalence between the words in an English sentence and a Persian sentence. Word alignment is based on a dictionary approach, and since word equivalency alone is the

only parameter observed, the meaning of the phrase or sentence as a whole may be changed somewhat, or at best have its fluency greatly impaired.

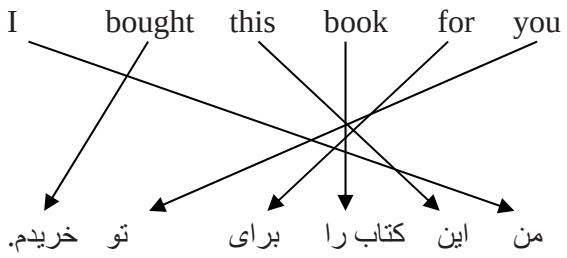


Figure 3-3: Example of Persian-English alignment (3)

In SMT, all possible alignments between sentence pairs are examined, and the most likely arrangement is determined. The most important factor in determining the probability of a certain alignment is to what degree the aligned words are linguistically equivalent. A significant amount of this information is contained within the sentence-aligned data. Dempster, Laird *et al.* (1977) developed the Expectation-Maximisation (EM) algorithm, an iterative algorithm which enables systematic identification of word alignments for which there is substantial evidence throughout the parallel corpus alone. Each iteration of the algorithm involves two steps defined as the Expectation (E) step, and the Maximisation (M) step. In the E-step, the alternative word alignment of each sentence pair in the corpus is assigned a probability based on the word pair probabilities defined in the model. The M-step involves using the probabilities of the corpus-specified word alignments to compute new probabilities for each word pair in the model. The model is then updated using these new probabilities and, in effect, the probabilities of the model are re-evaluated based on the number of occurrences of the word pairs in the set of word alignments. Iterations are repeated until estimates cease to be improved.

3.6.2 Phrase Alignment

The algorithm used in word alignment will give different results depending on the direction of alignment. An alignment operation with English as the source language and Persian as the target language will have a number of differences compared to Persian as source and English as target.

The alignment algorithm is able to produce alignments of single-to-single (single source word to single target word) and single-to-multi (single source word to multiple

target words). However, it is unable to align multiple-to-single or multiple-to-multiple. Word alignment takes place in both directions in the training process of an SMT system (i.e., English-to-Persian and Persian-to-English). In this way, single-to-multi alignments are extracted in both directions. Multiple-to-multiple alignments are extracted using phrase-alignment heuristics (Koehn, Och *et al.*, 2003b; Och, 2003; Och & Ney, 2003), which work with the word alignment algorithm output. In this operation, word alignment is first carried out on each training sentence in both directions, and the output represented in a bi-text grid (Figures 3-1 and 3-2). The word alignment sets are refined by removing alignments occurring only on one set. The resulting output is shown in Figure 3-3. Further alignments are iterated where alignments are adjacent and the source or target word is unaligned (Figure 3-4).

Table 3-2: Phrase alignment examples

didn’t buy	↔	نخريدم
I didn’t buy	↔	من نخريدم
this book	↔	اين كتاب
this book	↔	اين كتاب را
didn’t buy this book	↔	اين كتاب را نخريدم
didn’t buy this book yesterday	↔	اين كتاب را ديروز نخريدم
I didn’t buy this book yesterday	↔	من اين كتاب را ديروز نخريدم

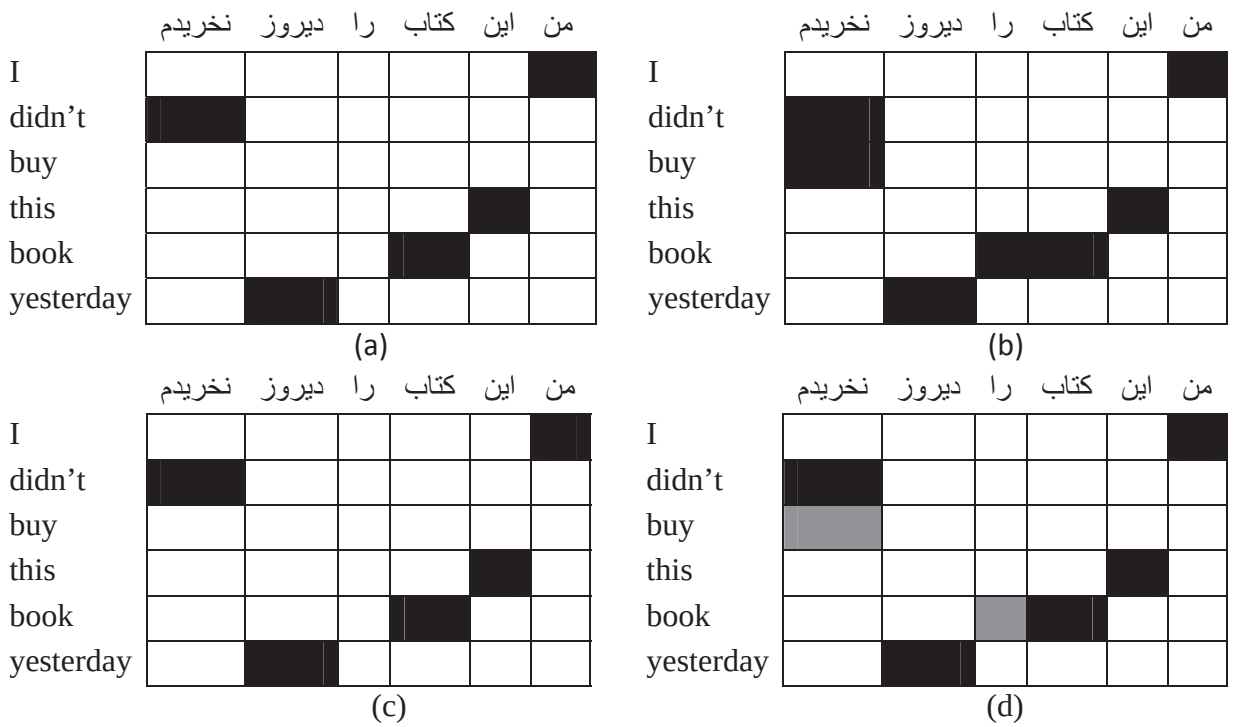


Figure 3-4: English – Persian Bi-text grid

3.7 Language Model

The language model is used by the decoder to determine the validity and fluency of a proposed target sentence or phrase. In this way, the probability $Pr(e_1^I)$ of even an unseen target sentence or phrase *e* can be checked, based on the language model. The language model, extracted from the corpus, gives the frequency of substrings in that corpus. When the input sentence's probability is determined, it is based on the substrings of that sentence compared to those in the model.

3.7.1 Uni-gram Model

One basic language model, known as the uni-gram model, may be simply composed of substrings alone, based on corpus word tokens. The probability of a word type is given by taking the total number of times that word occurs in the corpus and dividing it by the total number of word tokens found in the corpus. However, there are significant limitations to this model. Since it only operates on single word types, this leads to unwanted characteristics, such as the tendency to score shorter sentences higher than others. This is due to the fact that short sentences contain fewer probabilities. Incorrectly high probabilities are also generated when the model must deal with grammatically incorrect sentences, such as repeated (redundant) words. On the other extreme, a probability of zero is assigned to a sentence containing a word unknown to the model.

One simple method to improve the issue of unknown words is to increase the size of the parallel corpus the model is trained on, thus increasing the model's vocabulary. However, since there is no way of ensuring that all or even a high percentage of every word in a language is included in the parallel corpus used, this method alone is inadequate. For this reason, smoothing techniques (Bahl, Baker *et al.*, 1978) are also used. These techniques assign a small probability score to sentences and phrases with unknown words, but are able to determine sentences and phrases with greater numbers of unknown words than others, and can assign appropriate probabilities (i.e., less) to them. In this way each phrase and sentence is guaranteed a non-zero score.

3.7.2 Bi-gram Model

A bi-gram language model is a model consisting of all bi-grams (two-word substrings) found in the corpus. Such language models operate based on word

sequences. Probabilities are defined by determining the likelihood of the bi-gram's second word occurring, given the first word. The probability is calculated by determining the number of occurrences of a particular bi-gram in the corpus, and dividing that figure by the number of occurrences of the first word in the bi-gram.

3.7.3 N-grams

Larger models, known as n-grams, are based on the same logic as bi-grams, with *n*-length substrings, or n-grams. N-grams are strings of length *n* generated from words in texts. In traditional vector space approaches, dimensions of the document space for a given collection of documents are words or sometimes phrases that occur in the collection. By contrast, in the n-gram approach, dimensions of the document space are n-grams, namely, strings of *n* consecutive characters extracted from words. Since the number of possible strings of length *n* is a lot smaller than the number of possible single words in a language, n-gram approaches, therefore, have smaller dimensionality (Aleahmad, Hakimian *et al.*, 2007). So, the n-gram method is a remarkably pure statistical approach, one that measures statistical properties of strings of text in a given collection without regard to the vocabulary, or the lexical or semantic properties of natural language(s) in which documents are written. The n-gram length (*n*) and the method of extracting n-grams from documents vary from one author and application to another (Mustafa, 2005).

Both bi-gram or n-gram models operating on any string length, still encounter issues in this particular case of unknown n-grams. In general, the larger the n-gram model, the greater the issue becomes, as fewer occurrences are returned. Increasing the training corpus size helps slightly, and using smoothing techniques will aid the probability scoring somewhat, however, even with smoothing techniques there is no way to determine whether the individual words in a previously unseen n-gram have already occurred in the training corpus.

It can be seen, therefore, that there is a trade-off between flexibility and obtaining accurate word order. To make the best of this situation, various n-gram models are used, each with different weights, the scores of which are combined. In this way, a more accurate probability for a given sentence may be obtained.

A sample segment of a 5-gram language model is shown in Appendix II, section 1.

3.8 Translation and Evaluation for Training Purposes

In the previous section, it was noted in the log-linear model,

$$\arg\max_{I,e_1^I}\left\{\sum_{m=1}^{M}\lambda_m\, h_m\left(e_1^I, f_1^J\right)\right\} \qquad \textbf{3.10}$$

that λ_m is a weight showing the relative importance of each feature. The values of λ can be changed to control the relative importance of each feature used in the model. Many features are optional, depending on the language pair being used. As well as translation and language models, other features commonly added include the following:

- source-to-target and target-to-source phrase tables
- n-gram language model over target sequences
- phrase reordering model
- source-to-target and target-to-source lexical translation probabilities
- standard word/phrase penalty (controlling target sentence length)

Any feature added must be linked to a value of λ to define its importance relative to the other features in the model, and the influence it will have on the final output. Minimum Error Rate Training (MERT) is an approximation technique proposed by (Och, 2003), and is used to optimise system performance by determining the best weights for each feature used in the model. The MERT technique involves using the SMT system to translate a reference text, a set of source-target sentence pairs called the dev-set (development set). The output is then scored using a metric such as BLEU [see next section]. The λ values are then adjusted, and the same process is repeated, observing whether the change in λ values caused an improvement in the output. It is important that the dev-set is not part of, nor included in, the training set. However, the closer the dev-set is to the actual test set, the more the model's adjusted settings will be suited to it. MERT is limited to the number of parameters and features it can work with. When a large number of features require tuning, MERT cannot be relied on to determine the best feature weights (D. Chiang, Marton *et al.*, 2008).

3.9 Decoding Process

This section covers decoding, or the actual translation phase. Decoding is a search process, whereby the most likely translation is to be determined from all possible

translations, given the translation model. The decoding process may be represented as, given a source sentence and a set of possible translations, the process which determines the most probable translation. Instead of generating all possible translations for a given input, input sentence substrings are matched with translation model substrings, each individual translation is retrieved, and those translations are concatenated to produce the full translation. As only a certain number of hypotheses may be generated in a given amount of time, it is necessary to maximise the number of probable hypotheses generated, and avoid producing hypotheses unlikely to be chosen (Al-Onaizan, Curin *et al.*, 1999). In summary, it is necessary to find the most probable translations in the given amount of time.

Currently, the most advanced decoding methods are implemented in a *beam-search decoder* (Koehn, 2004; P. Koehn, H. Hoang *et al.*, 2007a). In this method, the runtime of the system is governed by setting a number of hypotheses to be generated, known as a *beam stack*. This number is maintained throughout the decoding process. As new hypotheses are generated, they are added to the beam stack, until the stack has reached the maximum number of hypotheses. At this point, if a new hypothesis has a higher score than the lowest scored hypothesis in the stack, it will be added to replace the lowest-scoring hypothesis, and the maximum number in the stack is maintained.
Scoring of hypotheses to determine whether they are added to the beam stack is based in part on the log-linear equation, and also by a cost estimation factor awarded to hypotheses, the value of which depends on the difficulty of translation of the parts of the sentence the hypothesis covers (Koehn & Senellart, 2010). In this way, sentences which are relatively easy to translate are not incorrectly awarded higher probability than those which were simply more difficult to translate. The final stage of decoding involves searching the beam stack containing *n*-best list of candidate translations, where *n* is the source sentence length. The final sentence with the highest probability is selected and output as the chosen translation.

An example of the test set, reference set, output set and BLEU scores for a baseline Moses-based system is given in Appendix II, section 2.

3.10 Evaluation Metrics

A significant amount of research has been done in the field of automatic machine translation evaluation. Human evaluation of machine translation is comprehensive and

generally considered to provide optimum fluency, but unfortunately, for the most part, it is expensive, time-consuming, and has difficulty sustaining consistency in the process. The main motivation behind automatic machine-based methods of evaluation is that they are fast, inexpensive, language independent, and are necessary in order to facilitate MERT techniques. The most commonly-used evaluation metric is BLEU.

3.10.1 BLEU

The most commonly used metric is BLEU (BiLingual Evaluation Understudy), which was developed by a team at IBM. The BLEU system awards a score between 0 and 1 depending on how close a machine translation output is to that produced by a professional human translator.

The BLEU scoring metric was developed by K. Papineni, Roukos *et al.* (2002) at IBM's Watson Research Lab in 2001–2002. BLEU evaluates machine translation performance by taking the output of the system's translation of a reference text, and comparing that output to the reference translations in terms of total translation length, word choice and word order. The main score, or *n*-gram precision p_n, is based on the number of n-word sequences in the MT output compared to the number in the reference translation. The following equation is used to calculate p_n:

$$p_n = \frac{|C_n \cap r_n|}{|C_n|} \qquad \textbf{3.11}$$

Where C_n and r_n are the multi sets of n-grams occurring in the candidate and reference translations, respectively. $|C_n \cap r_n|$ represents the number of n-grams present in C_n that are also present in r_n, such that the number of n-grams present in $|C_n \cap r_n|$ is not greater than those present in r_n, regardless of the number of the number in C_n. This is to ensure that if a reference sequence occurs a greater number of times in the MT output than in the reference translation, the additional occurrences in the MT output will not affect p_n.

N-gram precision scores can decrease rapidly as *n* increases, since the likelihood of longer word sequences occurring in both the MT output and the reference translation decreases. This can result in the p_n score for higher values of *n* being too small to have any reasonable effect on the final score. This can be offset by combining the

scores for all n-values into a single score (K. Papineni, Roukos *et al.*, 2002). The combined p_n score is determined by the following equation:

$$P_n = \exp\left(\sum_{n=1}^{N} \frac{1}{N} \log(p_n)\right) \quad \textbf{3.12}$$

where the sum of the log of each score is multiplied by weight 1/N.

Where an output translation is shorter than the reference translation, the final precision score is multiplied by a brevity penalty, *BP*, which is a decaying exponential based on the length of the reference sentence compared to the MT output sentence. In this way, single word occurrences such as 'the' will not incorrectly be scored highly. The brevity penalty is calculated using the following equation:

$$BP = e^{max\left(1-\frac{length\ (R)}{length(c)},0\right)} \quad \textbf{3.13}$$

where *R* is the reference set, and *C* is the candidate (MT output) set. The final score is given by:

$$BLEU = BP \bullet P_n \quad \textbf{3.14}$$

or, as suggested by K. Papineni, Roukos *et al.* (2002):

$$Log(BLEU) = \left(1 - \frac{length\ (R)}{length(c)}, 0\right) + \sum_{n=1}^{N} \frac{1}{N} log(P_n) \quad \textbf{3.15}$$

since the ranking behaviour is more clearly observed when shown in the log domain.

3.10.2 NIST

NIST evaluation metric is somewhat of an extension of BLEU, but differs by taking the weights of *n*-grams into account. The scoring process involves adding all the information counts of co-occurring *n*-grams, summing them separately and normalizing with the total *n*-gram count. As well as information-weighted *n*-gram counts, NIST differs from BLEU in other areas, such as text pre-processing, and a lower penalty for word length difference. Translation text is scored from 0–100 (Zhang, Vogel *et al.*, 2004).

3.10.3 Meteor

METEOR evaluates a translation by computing a score based on explicit word-to-word matches between the translation and a given reference translation. If more than one reference translation is available, the translation is scored against each reference independently, and the best scoring pair is used. An improvement in this metric is the high correlation with human judgment. The range of scores is between 0 and 1 (Banerjee & Lavie, 2005).

3.10.4 TER

TER (Translation Error Rate) is a metric based on determination of the number of editing operations required to change the output of the system into that of the reference texts. Changes to output can include deleting, inserting, or substituting, as well as shifting whole sections of text in the output.

Computation of TER, however, is relatively time-consuming, and cannot be implemented effectively at document level (Agarwal & Lavie, 2008).

3.11 Open-source Decoding Software

There are implementations of subtasks and algorithms in SMT and even software tools that can be used to set up a fully-featured state-of-the-art SMT system.

Moses is a fully-featured, open-source SMT system developed at the University of Edinburgh (P. Koehn, H. Hoang *et al.*, 2007a), which allows one to train translation models using GIZA++ for any given language pair for which a parallel corpus exists (Och & Ney, 2003). This toolkit was used to build the initial baseline system in this project.

Dyer, Weese *et al.* (2010) present the development of a new open-source framework called CDEC, used for decoding, aligning and training work with various SMT models, including rule-based, phrase-based and SCFG-based models. Several features of CDEC give it advantages over other open-source decoders. Being written in C++, it has the benefit of efficient memory usage and superior run time performance. It is not limited to extraction of just *k*-best translations, but is also able to extract alignments to references. Its use of gradient-based and gradient-free optimization allows CDEC to implement discriminative training.

Where most MT models use FSTs (phrase-based models such as that used in Moses (P Koehn, H Hoang *et al.*, 2007) or lexical models (PF Brown, Della Pietra *et al.*, 1993) or SCFGs – hierarchical phrase-based models such as that used in Joshua, or Jane (D. Chiang, 2007; Vilar, Stein *et al.*, 2010), CDEC implements both these classes and maximises on the benefits of each of them.

Dyer, Weese et al. propose that both phrase-based and hierarchical models are lacking significantly in certain areas, specifically in being unable to extend easily to new algorithms and models. They identify this to be because the translation, language model integration, and pruning algorithms are too closely linked, resulting in either difficulty or inability to examine different translation models. Another area identified is the limited number of dense features in phrase-based parameterisation. This has been improved in CDEC, with any parameterisation configuration supported, even up to millions of features.

These features, coupled with tight C++ coding, enable fast, efficient translation with low memory usage. In their experiments, Dyer, Weese *et al.* (2010) show that in other decoders there is a trade-off between run time and memory usage. They perform experiments decoding an English-Chinese test set, using Joshua (1x), Joshua (8x), Hiero and their own CDEC. Results of average run time per sentence, and memory usage are given. In Joshua (1x), written in Java, an average run time of 0.98 seconds and 1.5GB of memory are used. In Joshua (8x), an average run time of 0.35 seconds, but 2.5 GB memory, is used. Hiero, written in Python, although only using 1.1 GB of memory, is the slowest, with an average run time of 4.04 seconds. CDEC performs the best, with an average run time of 0.37 seconds per sentence, and 1.0 GB of memory used.

3.12 Summary

In summary, the task of SMT is based on Bayes decision theorem, and, in this case, the log-linear model form. A baseline SMT system consists of a training model (generated from the parallel corpus) aligned usually on phrase level, a language model (from the monolingual corpus in the target language), and a translation model. The translation model, $Pr(e_1^I | f_1^J)$, determines the probability of target sentence *e* being linguistically the equivalent of source (input) sentence *f*. This probability calculation is determined by searching the training model for the most likely target phrases and

sentences. These are then checked against the language model to determine their validity as sentences. Thus, the correct output with the highest probability is chosen as the output.

Two corpora are used in the process: bilingual and monolingual. The bilingual corpus is used in the construction of the training model, which is used to determine the most likely translation phrase. The monolingual corpus is used to construct the language model, which is used in determining if the proposed translation is a valid sentence.

Output is evaluated automatically with evaluation metrics, which score the output according to a number of parameters particular to that specific metric. The most commonly used metrics are BLEU and NIST. BLEU scores output by comparing parameters of translation length, word choice, and word order to a reference text. NIST differs from BLEU by taking into account weights of *n*-grams, and using a lower word length penalty.

Chapter 4. Initial Tests and Corpus Development

"Knowledge is of no value unless you put it into practice."

~ Anton Chekhov

4.1 Introduction

This chapter describes the first baseline system that was built using the open-source toolkit Moses. Tests are performed, and the results are collected and discussed. The chapter then shows the work in developing the corpora used for the training and language models, how they were aligned and tuned, and divided into different systems in order to determine the best possible combination of domains. Extensive experimentation is carried out, together with presentation and discussion of the results.

4.2 Initial Set-up and Testing

This system was trained and tested in the English-Persian translation direction, using a parallel corpus of data originating mainly from BBC's Persian News website[4] and United Nations' documents sourced by a website which collects political commentary in multiple languages. We modified and developed this corpus further in-house to suit the system. Training and testing was repeated as the corpus size grew. A training model was constructed using this parallel corpus. This was aligned using the Microsoft bi-lingual sentence aligner developed by Moore (2002). The language model was then manually prepared, with blank lines and other inconsistencies deleted. Alignment was also performed manually, with the aim of improving the results. Testing was performed using a model test sentence with a confirmed 100% accurate human translation. The key requirement of this test set was its absence from the training and language models. The human translation was used as the reference set, against which the accuracy of the MT output is measured. Different tests were performed as we continued to increase the training model size, using different language model sizes, as shown in Table 4-1.

[4] http://www.bbc.co.uk/persian/

Table 4-1: Training model and Persian language model sizes

Test No. (En/Pe)	1	2	3	4	5
Training Model Sentences	730	817	1011	1011	2343
Language Model Sentences	864	1066	864	5514	7005

Evaluation results from these experiments are presented in Figure 4-1, not surprisingly showing an improvement in BLEU scores with the increased corpus size. However, these initial tests yielded very unsatisfactory results when compared to those of other SMT systems for other languages. This, we believe, was due mainly to the size of the training model, which at 2343 sentences was miniscule compared to what is normally required to achieve any reasonable output.

Due to the significant differences between the Persian and English languages (as mentioned in Chapter 2) several problems were encountered, such as the large difference between the number of sentences in the source and target languages, as well as the differences in the types and symbols used for punctuation. These issues had to be taken into account in order to achieve the best alignment results. Needless to say, the better the alignment, the better the translation result.

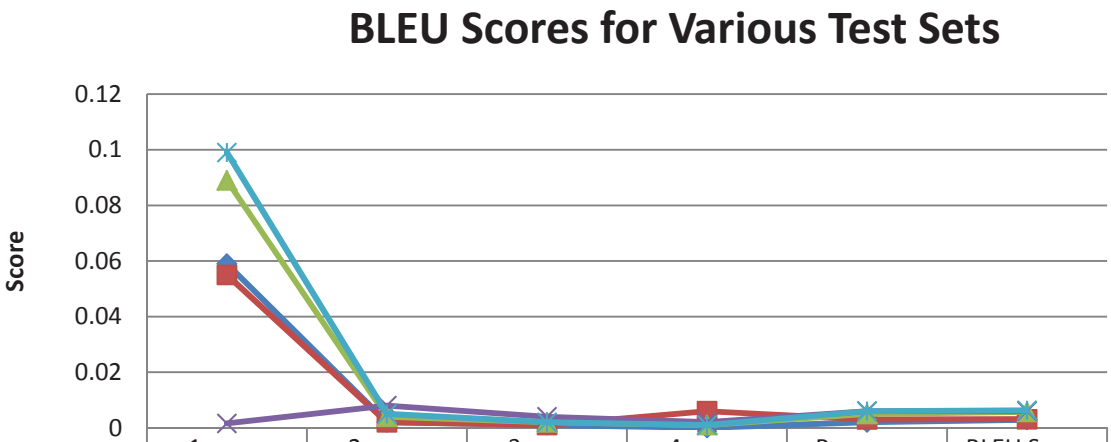
Table 4-2: BLEU scores for test with different sized models

Test Number	1	2	3	4	5
1-gram	0.059	0.055	0.089	0.016	0.099
2-gram	0.002	0.002	0.004	0.008	0.005
3-gram	0.001	0.001	0.002	0.004	0.002
4-gram	0.000	0.006	0.001	0.002	0.001
Pre-score	0.002	0.003	0.005	0.006	0.006
BLEU	**0.0029**	**0.0031**	**0.0057**	**0.0060**	**0.0063**

The first translation test was performed with a training model of 730 (parallel) sentences. The language model used consisted of 864 sentences in Persian, and the output result was evaluated using BLEU with 1 reference text, to 4-gram precision, and case-sensitivity (BLEU**r1n4c**).

In the second test, the training model was increased to 817 sentences, and the language model to 1066 sentences. As expected, the results improved slightly. The third test used a training model of 1011, but used the same language model as in the

first test, at 864 sentences. Tests 4 and 5 had training models of 1011 and 2343, respectively, and language models of 5514 and 7005, respectively. However, this time the results showed little improvement, and were fairly similar to test 3. There was a small increase in the BLEU score when a set of 2343 sentence pairs was used. The increase in the BLEU score with the increased training model size is shown in Table 4-2 and Figure 4-1. It must be noted that BLEU is only a tool to compare different machine translation systems. So, an increase in BLEU scores would not necessarily mean an increase in the accuracy of translation.



	1-gram	2-gram	3-gram	4-gram	Pre-score	BLEU Score
Test 1	0.059	0.002	0.001	0	0.002	0.0029
Test 2	0.055	0.002	0.001	0.006	0.003	0.0031
Test 3	0.089	0.004	0.002	0.001	0.005	0.0057
Test 4	0.0016	0.008	0.004	0.002	0.006	0.006
Test 5	0.099	0.005	0.002	0.001	0.006	0.0063

Figure 4-1: BLEU scores for various tests

4.3 Discussion and Analysis of Initial Results

After the initial tests, further experiments were carried out, this time with fixed training model sizes of 817, 1011, and 2343 sentences. For each training model size, three tests were performed with language model sizes of 864, 1066, and 7005 sentences, to give a total of nine tests. The results of these tests are shown in the following pages in Tables 4-3 to 4-5. The performance of the system was evaluated by computing BLEU and NIST (Zhang, Vogel *et al.*, 2004) scores for the translation outputs of each configuration. Tables 4-3 to 4-5 show the results obtained from these tests.

Table 4-3: Results for 817-sentence training model

Training model size (sentences) = 817			
Language model size (sentences)	864	1066	7005
BLEU	0.1061	0.0920	0.0805
NIST	1.8218	1.6838	1.6721

As evident from Table 4-3, an increase in language model size does not necessarily mean an improved translation. Initially, it was thought that this was due to the difference in size between the language model and the corpus, and that maintaining a similarity in size would improve translation output.

Table 4-4: Results for 1011-sentence training model

Training model size (sentences) = 1011			
Language model size (sentences)	864	1066	7005
BLEU	0.0882	0.0986	0.0888
NIST	1.5338	1.5301	1.5512

Table 4-4 seemed to confirm this assumption, since as shown, using training and language models of 1011 and 1066 sentences, respectively, yielded a better result. It was determined afterwards, however, that the size difference between training and language models has little effect; instead the unusual score difference is attributed to the exceedingly small amount of data the system was given to work with. The differences between the BLEU and NIST results here can be attributed to the actual differences between the operational parameters of the metrics themselves, for example NIST has a lower penalty for word length differences than BLEU, and the range of score is between 0 – 10 (10 being 100% accuracy). BLEU however compares the output to the reference translation based on the total translation length, word choice, and word order.

The best translation output was achieved in the final test, with the training model of 2343 sentences, and a language model of 7005 sentences. This indicated the improving influence of a corpus of greater size, as shown in Table 4-5 and Figures 4-2 and 4-3.

Table 4-5: Results for 2343-sentence training model

Training model size (sentences) = 2343			
Language model size (sentences)	864	1066	7005
BLEU	0.0806	0.1127	0.1148
NIST	1.7364	1.6961	1.7554

It was determined that an increase in the size of the corpus will improve the quality of translation, but that to construct a feasible system that is able to produce reasonable output, a parallel corpus of much greater size is needed. Given the characteristics of the Persian language, the tests conducted indicated that in applying SMT to this language, although the size of corpus affects the quality of the translation to some extent (as measured using the BLEU metric), the improvement of the output would be much more noticeable when corpora of a much greater size are used. At this stage, it was also supposed that even then, the size of the corpus is not the only key parameter, but the domain of both the parallel and monolingual corpora are also of significant concern. This observation is also made by (Ma & Way, 2009).

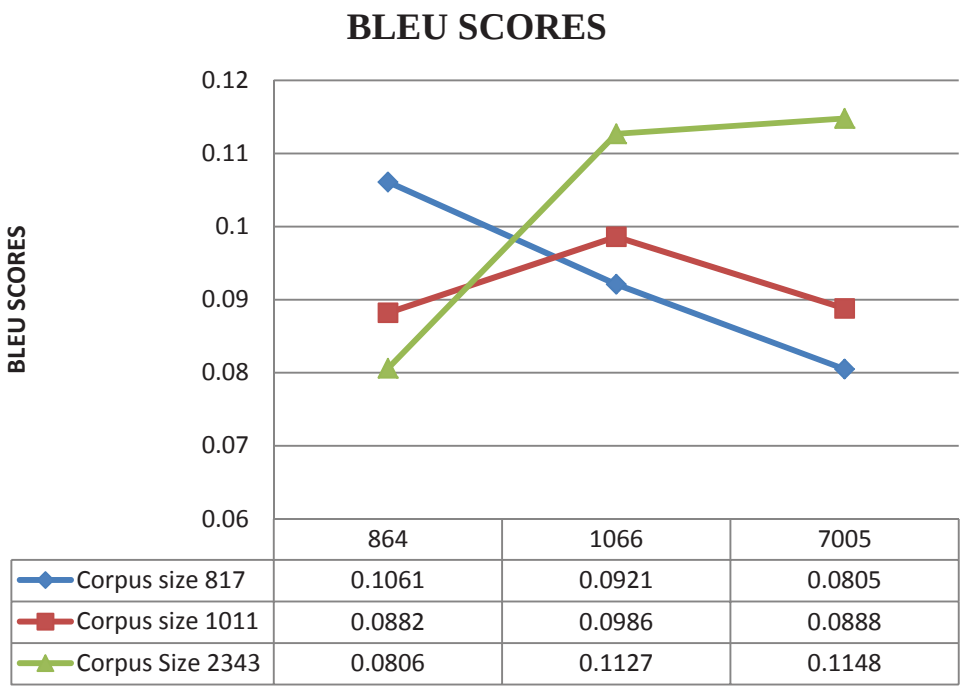
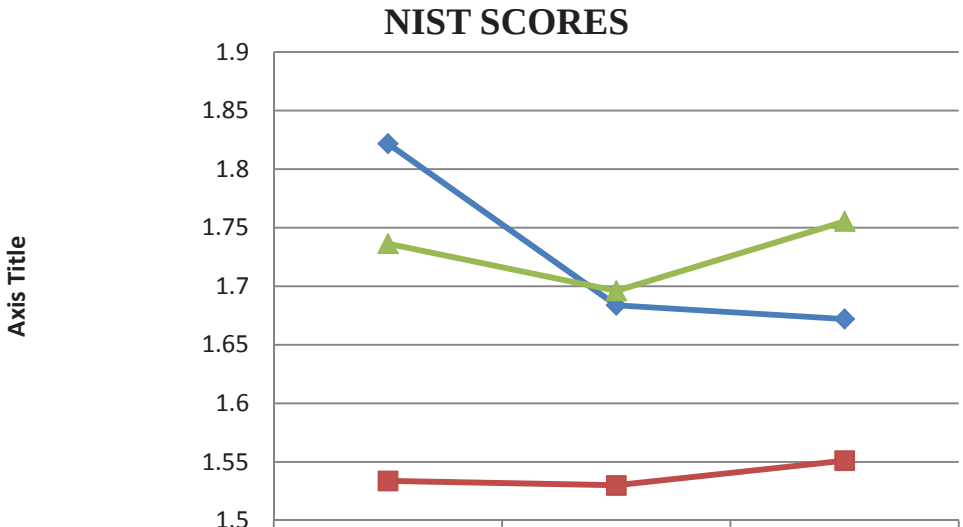


Figure 4-2: BLEU scores vs. language model sentences for each system configuration



	864	1066	7005
Corpus size 817	1.8218	1.6838	1.6721
Corpus size 1011	1.5338	1.5301	1.5512
Corpus Size 2343	1.7364	1.6961	1.7554

Figure 4-3: NIST scores vs. language model sentences for each system configuration

There were issues encountered in the process of parallel corpus alignment due to the major differences between English and Persian. This resulted in a difference between the number of sentences in the source and target languages, and the differences in the types and symbols used for punctuation. These issues had to be taken into account while performing an alignment of the corpus. As noted in Chapter 3, before any attempt at SMT can be made, accurate alignment of the parallel corpus in use is paramount, since the translation output is directly related to the accuracy and quality of the alignment.

At the time these initial tests were made, it was the earliest reported instance of SMT being used for the English/Persian language pair (Mohaghegh & Sarrafzadeh, 2009). The first objective of these tests was to determine with what success Persian could be translated to English using a statistical approach (the success of which was evaluated using metrics such as BLEU, NIST). After an output was produced and scored, it was necessary to determine which conditions of the system gave rise to lower scores, and which caused an increase in accuracy. Issues leading to low output accuracy needed to be determined, together with how they might be resolved. The second part of the work

was to repeat each process using different sized parallel corpora, comparing results and finding a relationship between the size of the parallel corpus and the quality of the output. Although the size of the language model and the training model both affect the translation output, the size of the training model is more influential.

At this stage, there were several issues surrounding sentence alignment which needed to be investigated further. It was believed that accuracy could be increased by categorising the corpus into different subject domains. At the time, it consisted of a mix of genres, such as news stories, poetry, scientific documents and other literature. After these tests and test analyses, it was proposed that incorporating linguistic inputs – such as POS (part-of-speech) tagging, parsing, morphological analysis, semantic modelling and a dictionary specific to the domain – would make such a system more robust in terms of accuracy and because of this they were suggested as an area of development. However, the biggest requirement, and incidentally what proved to be an ongoing challenge, was obtaining or concatenating a parallel corpus feasibly large enough to be used in an effective SMT system. Other successful language pairs use parallel corpora of sometimes up to billions of words. The translation accuracy we acquired from the tests was far from satisfactory, and when compared to other machine translation systems on other language pairs, the output left much to be desired.

4.4 Corpus Development

Earlier in this chapter, initial tests that were carried out in implementing an SMT system on a range of corpus sizes up to several thousand sentences, and determining areas in the system which needed immediate development, such as fine-tuning the alignment process, and the need to significantly increase the size of the bilingual corpus in use. This section shows significant achievements reached in the project, in particular the acquisition of a large amount of Persian/English bilingual text. Details are then given on the tests that were run using different selections of data in different quantities. High-performing system arrangements are examined, and details which lead to higher quality output are investigated. It is shown that increasing the size of the parallel corpus (and, therefore, training model), and using different sizes of monolingual data to build a language model, will affect the output of the SMT system. It had to be determined whether or not certain domains in the corpus would give the desired results and, if necessary, remove those which seemed to have an adverse

effect on the output. It is explained that improved results are due to two main factors: first, using an in-domain corpus is superior to a mixed domain corpus, even if it is smaller; secondly, focusing on stringent alignment of the parallel corpus prior to training.

For optimum operation, the training model requires a significant amount of data that must be trained to generate accurate probabilities. Several Persian monolingual corpora were obtained, completely adapted to news stories and originating from three different news sources – Hamshahri (AleAhmad, Amiri *et al.*, 2009), IRNA[5] and BBC Persian[6]. Hamshahri contained around 7.3 million sentences, IRNA almost 5.6 million, and the BBC corpus contained almost 10,000 sentences.

Certain common language pairs have multiple millions of sentences available. Unfortunately for Persian/English, there is a significant shortage of digitally stored bilingual texts, and obtaining a corpus of reasonable size is a challenge.

One English-Persian parallel text corpus that was obtained consisted of almost 100,000 sentence pairs of 1.6 million words, and like the monolingual corpora, originated mostly from bilingual news websites. There were a number of different domains covered in the corpus, but the majority of the text was in literature, politics, culture and science. It is believed that currently the only freely available corpus for the English-Persian language pair is the TEP corpus, which is a collection of movie subtitles consisting of almost 3 million sentences of 7.8 million words. This corpus and the first were concatenated together to form what was called the News Subtitle Persian English Corpus (NSPEC) a single corpus of 3.1 million sentences for use in one test, and will also be used in the future for further experiments. Figure 4.4 shows the composition of the corpus divided into separate domains**.**

[5] http://www.irna.ir/ENIndex.htm
[6] http://www.bbc.co.uk/persian/

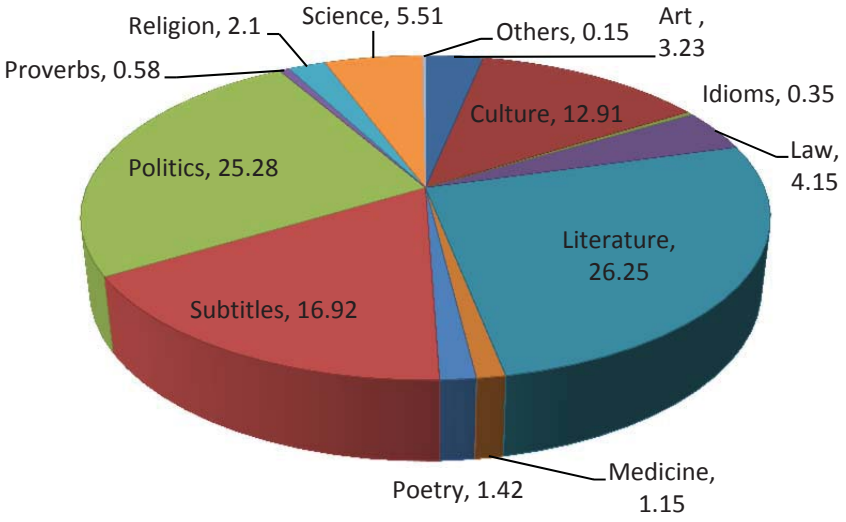


Figure 4-4: Domain percentages for NSPEC corpus

4.5 Alignment

The issues surrounding word alignment of Persian/English parallel corpora have been the subject of much attention. It has been shown that sentence-aligned parallel corpora are useful for the application of machine learning to machine translation, however, it is unfortunately not usual for parallel corpora to originate in this form, aligned at sentence level. The alignment of the corpus became a task of paramount importance, especially due to the shortage of bilingual text for English-Persian in the first place: it was necessary to ensure the corpus being used was of the highest possible quality, to make up for this. There are several methods available to perform alignment. Characteristics of an efficient sentence alignment method include speed and accuracy, but it is also beneficial if the alignment method is language independent and does not require prior knowledge of the corpus. For the experiments presented in this section, a hybrid sentence alignment method was used, with sentence length-based and word-correspondence based models that covered all these areas, only requiring the corpus to be separated into word and sentence. In each of the experiments the corpus was first manually aligned using this hybrid method, and then later using GIZA++ when the data was put through Moses.

4.6 Experiments and Results

4.6.1 Overview of earlier English-Persian experiments

The original tests performed using the SMT system produced unsatisfactory results, as published by Mohaghegh, Sarrafzadeh *et al.* (2010). It was determined that this was due mainly to the small corpora and training models used. As detailed in these papers, a number of preliminary tests were carried out, and each time the language model was increased in size to a maximum of 7005 sentences. The training model at its largest consisted of 2343 sentences. The language model in these tests consisted of text collected from BBC news stories, and the training model consisted of a bilingual corpus of mostly UN news. It was thought that the unsatisfactory test results achieved could be remedied by enlarging the language model and corpus, since the amounts of data in each model were far too small to achieve any reasonable success in SMT.

4.6.2 Further experiments in the English-Persian Translation Direction

4.6.2.1 Data Development

In order to develop the training model, an English-Persian parallel corpus was built, as explained in the initial set-up and testing in Section 4.1. The parallel corpus was divided into different sized groups for each test system. The details of the corpus size for each test are shown in Table 4-6. Table 4-7 shows the size of each test's corpus after the text was tokenized, converted to lower case, and stripped of blank lines and their correspondences in the corpora. This data was obtained after applying the hybrid sentence alignment method, as explained in Section 4.2.

The corpus was divided to construct five different systems, beginning from 10,000 sentences in the smallest corpus, and increasing in steps of approximately 10,000 sentences each time up to the fifth test system, with a final corpus of approximately 53,000 sentences. In addition to the news stories corpus as shown earlier, the only other corpus that was freely available for research consisted of movie subtitles in Persian and English. This was shown to be in a completely different domain to the main corpus, so, for most cases, it was preferred to run tests separately when using these corpora.

Table 4-6: Bilingual corpora used in the training model

Training Model	Data Genre	English Sentences	English Words	Persian Sentences	Persian Words
System 1	Newswire	10874	227055	10095	238277
System 2	Newswire	20121	353703	20615	364967
System 3	Newswire	30593	465977	30993	482959
System 4	Newswire	40701	537336	41112	560276
System 5	Newswire	52922	785725	51313	836709
TEP	Subtitle	612086	3920549	612086	3810734
NSPEC	Newswire + Subtitle	678695	5596447	665678	5371799

Table 4-7: Bilingual corpora after hybrid alignment method

Training Model	Data Genre	English Sentences	English Words	Persian Sentences	Persian Words
System 1	Newswire	9351	208961	9351	226759
System 2	Newswire	18277	334440	18277	362326
System 3	Newswire	27737	437871	27737	472679
System 4	Newswire	37560	506972	37560	548038
System 5	Newswire	46759	708801	46759	776154
TEP	Subtitles	612086	3920549	612086	3810734
NSPEC	Newswire Subtitle	618039	5370426	618039	5137925

Finally, in NSPEC, the two complete corpora were concatenated to give a combined corpus of over 600,000 sentences. This was done to ascertain the system's performance potential when using a combined corpus. The subtitle corpus was tested separately to observe how an out-of-domain corpus would affect the output result. Each parallel corpus was used to build a training model specific to that test set. In all cases, the reference text consisted of a news article covering a variety of different domains showing various grammatical aspects of each language. Transcriptions and corpora originating from newspaper stories were used to construct a language model. One source that was used was the Hamshahri corpus, extracted from the Hamshahri newspaper, one of the most popular daily newspapers in Persian which has been in publication for more than 20 years. Hamshahri corpus is a Persian text collection that consists of 700Mb of news text spanning from 1996 to 2003. This corpus is designed

for classification tasks and contains more than 160,000 news articles on a variety of topics. Another source used was the IRNA corpus, consisting of almost 6 million sentences collected from IRNA (Islamic Republic News Agency). Table 4-8 summarises the monolingual corpora used for the construction of the language model. SRILM toolkit (A. Stolcke, 2002) was used to create language models up to 5-gram precision using these resources. The baseline system was tested using different sizes of aligned corpora and different sized language models. Tables 4-9, 4-10 and 4-11 show the results obtained using the BBC, Hamshahri, and IRNA language models, respectively.

Table 4-8: Monolingual corpora used to train the language model

Monolingual Corpus	Data Genre	Sentences	Words
BBC	News	7005	623953
Hamshahri (V.1)	News	7288643	65937456
IRNA	News	5852532	66331086

4.6.2.2 Testing and Evaluation of Results

The first experiment was carried out with a training model of 10,000 sentences (System 1) in the English-Persian translation direction. This training model was used with the three different language models. As shown in Tables 4-9, 4-10 and 4-11, the best result was achieved when the machine was run with the IRNA language model. The next test set (System 2), consisted of a training model of almost 21,000 sentences, and tests were repeated for each different language model. Again, the output score showed that using IRNA resulted in the best translation, followed by BBC and then Hamshahri. Almost identical trends were observed in each test set, up to the set with the largest training model (53,000 sentences – System 5). We originally thought that the increase in the size of both models would yield a much higher metric score, since it gave the translation program more data to work with. However, these new tests proved that this was not necessarily always true, and increased corpus size alone was not synonymous with improved translation. For instance, in the case where the Hamshahri corpus was used to construct the language model, the output result was even worse than the initial baseline tests even with use of a far smaller corpus like

BBC. The tests with the IRNA-based language model, much larger than the original BBC corpus but still smaller than Hamshahri, yielded the best result of the three.

To establish a reason for the apparently illogical test results, the characteristics of each corpus were examined, together with their combinations in each test. After analysis, it was observed that there were a number of likely factors contributing to the poor results.

Table 4-9: Evaluation metric scores with Hamshahri-based language model

Hamshahri-based Language Model				
Training Model	**Evaluation**			
	BLEU	**NIST**	**METEOR**	**TER**
System 1	0.1081	2.1453	0.2526	0.8106
System 2	0.1229	2.4721	0.3078	0.7196
System 3	0.1325	1.2080	0.2215	0.7236
System 4	0.1945	2.4804	0.2970	0.7500
System 5	0.2127	3.6452	0.3040	0.8863
TEP	0.0127	1.2547	0.1377	0.9015
NSPEC	**0.0856**	**1.9871**	**0.2313**	**0.7825**

Table 4-10: Evaluation metric scores with BBC News-based language model

BBC News-based Language Model				
Training Model	**Evaluation**			
	BLEU	**NIST**	**METEOR**	**TER**
System 1	0.1417	2.4803	0.3104	0.7500
System 2	0.1700	2.5258	0.3347	0.6287
System 3	0.2385	3.4394	0.3654	0.6312
System 4	0.2645	3.6466	0.4466	0.6515
System 5	0.2865	3.8441	0.4479	0.8181
TEP	0.1312	2.6552	0.2372	0.8333
NSPEC	**0.2152**	**3.2643**	**0.3929**	**0.6824**

Table 4-11: Evaluation metric scores with IRNA-based language model

IRNA-based Language Model				
Training Model	**Evaluation**			
	BLEU	**NIST**	**METEOR**	**TER**
System 1	0.2472	3.5099	0.4106	0.6969
System 2	0.3287	4.0985	0.4858	0.5833
System 3	0.3215	4.1409	0.4838	0.5606
System 4	0.3401	4.2090	0.4833	0.5833
System 5	**0.3496**	**4.4925**	**0.5151**	**0.5236**
TEP	0.0535	1.8830	0.2021	0.8787
NSPEC	**0.1838**	**3.0264**	**0.3380**	**0.7234**

One such factor was the nature of the data contained in each corpus, and how this affected the match between the language model and the training model. For instance, in the case where an even lower score than the original tests was achieved, it was noted that the training model was constructed with a corpus comprising mainly movie subtitles, yet the language model based on the Hamshahri corpus was a collection of news stories. For the most part, movies consist of spoken, natural language in informal situations, filled with idioms, colloquial expressions and terms, and commonly incorrect grammar and sentence structure. These characteristics were heavily present in the training model. News stories on the other hand not only ideally consist of well-structured sentences, with correct grammar and little presence of colloquialism, but the very nature of this kind of literature is unique, and rarely found in natural spoken language. Another example showing this involved the subtitle corpus (TEP). This corpus was significantly larger in size (612,000 sentences) when compared to the other corpora that were available for use. However, when we performed the same tests using different language models, the result was unsatisfactory. It was concluded that this was due to the test sets being in a different domain than that of the movie subtitles. These results confirmed that using larger language and training models alone was not a reliable determining factor in satisfactory output.

For comparison, Google Translate was tested on the same test data with results shown in Table 4-12. The system output was compared with that of Google Translate, using the same evaluation metrics as before. Comparison shows the system significantly outperforms Google Translate in the English-Persian translation direction.

Table 4-12: Evaluation metric score comparison between Google Translate and System 5 with IRNA-based language model

Google Translate (English-Persian)				
	BLEU	**NIST**	**METEOR**	**TER**
Google	**0.2611**	**3.7803**	**0.5008**	**0.7272**
System 5	**0.3496**	**4.4925**	**0.5151**	**0.5236**

4.6.3 Experiments in the Persian–English Translation Direction

4.6.3.1 Data Development

Two news story-based monolingual English corpora were used to construct the English language model, both of them originating from Europarl Corpus (Koehn, 2005). The Europarl corpus is extracted from the proceedings of the European Parliament in 11 different languages: Romanic (French, Italian, Spanish, and Portuguese), Germanic (English, Dutch, German, Danish, and Swedish), Greek and Finnish. The parallel corpus was the same as that used in the English-Persian translation direction shown earlier in Section 4.1.

4.6.3.2 Testing and Evaluation of Results

To develop a training model, the English-Persian parallel corpus was divided, as explained earlier in Section 4.1. The parallel corpus was divided into different sized groups for each test set (see Tables 4-6 and 4-7 for details).

Table 4-13 summarises the monolingual corpora used for the construction of the language model. SRILM toolkit (A. Stolcke, 2002) was used to create language models of up to 5-gram precision. The baseline SMT system was tested against different sized aligned corpora and language models. Tables 4-14 and 4-15 show the results obtained using the Europal and News–Commentary language models, respectively.

Table 4-13: Monolingual corpora used to train the language model

Monolingual Corpus	Data Genre	Sentences	Words
Europarl	News	1658841	40624075
News Commentary	News	18911860	44904370

Table 4-14: Evaluation metric scores with News Commentary-based language model

News Commentary-based Language Model				
Training Model	**Evaluation**			
	BLEU	**NIST**	**METEOR**	**TER**
System 1	0.1318	2.8344	0.3809	0.7535
System 2	0.2655	3.2458	0.4470	0.6225
System 3	0.2910	3.4425	0.4138	0.6952
System 4	0.3056	3.7057	0.4414	0.6278
System 5	**0.3332**	**3.8085**	**0.4685**	**0.5231**
TEP	0.0621	2.2952	0.2978	0.8236
NSPEC	**0.1975**	**2.9907**	**0.3831**	**0.6429**

Table 4-15: Evaluation metric scores with Europarl v4-based language model

Europarl v4-based Language Model				
Training Model	**Evaluation**			
	BLEU	**NIST**	**METEOR**	**TER**
System 1	0.1208	2.5952	0.3841	0.7463
System 2	0.1277	2.5592	0.4033	0.6376
System 3	0.2005	3.5310	0.4410	0.6231
System 4	0.2415	3.2908	0.43271	0.6449
System 5	0.2576	3.1892	0.40149	0.6225
TEP	0.0414	2.1196	0.2880	0.8623
NSPEC	**0.1796**	**3.1622**	**0.3950**	**0.6325**

The first experiment was carried out with the smallest training model of 10,000 sentences (System 1) in the Persian-English translation direction. The two different language models tested were based on the Europarl v4 corpus and the News Commentary corpus (Tables 4-14 and 4-15).

The testing procedure was the same as in the English-Persian translation direction tests performed earlier, with different training models used NSPEC being the largest.

In these tests, two different sized language models were used, one with approximately 1,700,000 sentences, and the other with almost 19,000,000 sentences. The size of the language model is important. For instance, in System 5, the News Commentary-based language model is over 10 times larger than the Europarl-based model. The BLEU score for this arrangement is almost 30% better than when the Europarl-based language model is used (cf. 0.2576 to 0.3332). However, where the TEP-based training model is used (TEP System), although the size of the corpus is dramatically larger than System 5, the BLEU score was far from satisfactory. Upon examination, it was determined that this was because the domains of the language model and training model were completely different. The system with the NSPEC-based training model (a combination of movie subtitles and newswire domains) shows a score also much lower than what might be expected. Again, it was determined that constructing a larger overall corpus by combining corpora would not necessarily lead to better output results, especially where the corpora used were of entirely different domains. These tests proved again that output quality is closely related to the domain and quality of the corpora used in the training and language models. Again, these results were compared to Google Translate's output scores for this language pair and in the Persian-English direction. As shown below in Table 4-16, the best translation system

arrangement (System 5 – Table 4-14) is compared to that of Google Translate. Despite Google Translate's scores being slightly higher, the scored output results are very close. It was concluded that this was due to Google's accessibility to much larger amounts of monolingual (in this case, English) data usable in the construction and training of a system, together with more focus on the Persian-English direction due to political matters in the Middle East.

Table 4-16: Evaluation metric score comparison between Google Translate and System 5 with News Commentary-based language model

Google Translate (Persian-English)				
	BLEU	**NIST**	**METEOR**	**TER**
Google	**0.3453**	**4.9075**	**0.5987**	**0.5072**
System 5	**0.3332**	**3.8085**	**0.4685**	**0.5231**

4.7 Summary

In this chapter we show initial testing using a baseline system using the open-source toolkit, Moses. The tests were run with relatively small training and language models, and the output was evaluated by BLEU and NIST metrics. It is shown that to achieve results of any degree of usability, much more data must be used.

Also presented is the development and testing of new corpora for use in the baseline SMT system, based on the open-source decoder Moses. It was shown that increasing the size of the corpus alone does not necessarily lead to better results. Instead, more attention must be given to the domain of the corpus. There is no doubt that the parallel corpora used in these experiments are small when compared to other corpora used in training SMT systems for other languages, such as German and Chinese, or with Google, which has access to extensive resources. However, this was the greatest challenge from the outset, to develop an effective reliable system for this low-resource language pair.

Chapter 5. Hierarchical Phrase-Based Translation Model

"God, grant me the serenity to accept the things I cannot change, the courage to change the things I can, and the wisdom to know the difference."

~ *Reinhold Niebuhr*

5.1 Introduction

In this chapter we investigate other methods of decoding, in particular the hierarchical phrase-based method Joshua. A comparison between Moses and Joshua is made, the benefits and shortcomings of each with respect to translation output are shown and the advantages of a hierarchical approach for the English-Persian translation direction is discussed.

5.2 Hierarchical Phrase-Based Overview

Most recent research in the area of statistical machine translation has been targeted at modelling translation based on phrases in the source language, and matching them with their statistically-determined equivalents in the target language ("phrase-based" translation) (Koehn, Och *et al.*, 2003b; Marcu & Wong, 2002; Och & Ney, 2004; Och, Tillmann *et al.*, 1999). Many modern successful translation machines use this translation approach.

A critical task in a phrase-based MT system is the determination of a translation model from a word-aligned parallel corpus. A phrase table containing the source language phrases, their target language equivalents and their associated probabilities, in most systems, is extracted in a pre-processing stage before decoding a test set (Deng & Byrne, 2006; Koehn, Och *et al.*, 2003b).

Moses toolkit (P. Koehn, H. Hoang *et al.*, 2007a) is an open-source phrase-based toolkit, and uses such a pre-processing approach in its training scripts. The hierarchical approach does not detract from the strengths of phrase-based approaches, but instead uses them to its advantage. In a phrase-based decoder, phrases are used in order to learn word reordering. In a hierarchical approach, this principle is taken a step further, and phrases are used for phrase reordering, using synchronous context-free grammars (SCFGs) to compose the hierarchical phrases from words and sub-phrases.

Synchronous context-free grammars can be represented as a tuple:

$$(N, S, T_\sigma, T_\tau, G) \quad \textbf{5.1}$$

Where N represents a set of non-terminal symbols of the grammar, $S \in N$ the goal symbol, T_σ the source terminal symbol vocabulary, T_τ the target terminal symbol vocabulary, and G represents grammar production rules. In G, each rule is in the form

$$X \rightarrow <\propto, \boldsymbol{\gamma}, \sim> \quad \textbf{5.2}$$

Where $X \in N$ represents a non-terminal symbol, $\propto$ is a sequence of symbols from $N \cup T_\sigma$, γ is a sequence of symbols from $N \cup T_\sigma$, and $\sim$ is a one-to-one correspondence between the non-terminal symbols $\propto$ and γ.

An SCFG's language is a set of ordered pairs of strings. During decoding, the set of hypothesis translations of an input sentence *f* is the set of all *e* such that the pair (*f*, *e*) is governed by the translation model SCFG. Each hypothesis *e* is generated by applying a set of rules. The cost of implementing each rule is given as:

$$\omega(X \rightarrow < \alpha, \gamma >) = \prod \emptyset_i \, (X \rightarrow < \alpha, \gamma >)^{\lambda_i} \quad \textbf{5.3}$$

Where each $\emptyset_i$is a feature function and $\lambda_i$is the weight for $\emptyset_i$. The product of the rules used in the derivation of the translation model is the translation model score of the hypothesis *e*, which is then combined with other features, such as a language model score, in order to produce an overall score for each hypothesis translation.

Hierarchical phrase-based translation (D Chiang, 2005) expands on phrase-based translation by allowing phrases with gaps, modelled as SCFGs. In effect, it is grammars that are used, not phrase tables. The original hierarchical implementation trains its SCFG translation model in a pre-processing stage similar to standard phrase-based models. A subsample of occurrences of given source phrases is used to calculate translation probabilities. Phrase translation and their model parameters can be determined at run-time as the system accesses the target language corpus and word alignment data. A suffix array can also be used to obtain hierarchical phrases at run time (Lopez, 2008).

Joshua is another well-known open-source machine translation toolkit, based on a hierarchical approach (Z Li, C Callison-Burch *et al.*, 2009). Originally, Joshua (Z. Li,

C. Callison-Burch *et al.*, 2009) was a re-implementation of the Hiero MT system (D. Chiang, 2007), but was extended by Z. Li, C. Callison-Burch *et al.* (2009) in order to support formalisms such as SAMT (Zollmann & Venugopal, 2006). Joshua is written in Java, and employs n-gram language model integration, chart-parsing, unique *k*-best extraction and beam and cube-pruning algorithms, and is also scalable for use on large-scale systems due to the use of parallel and distributed computing in its construction. Using Joshua, sentences can be translated using an aligned parallel corpus without the need to extract an SCFG prior to decoding. This implementation enables any input sentence to be decoded, and data structures are not as large as full phrase tables, using less disk space. However, because of this, the decoder has a slower running time as phrase translations must take place while running. Running the decoder is done in both the tuning stage and the testing stage. In this stage, memory is critical to the decoding process. As a decoder, Joshua is very memory-intensive, in particular when decoding large grammars and language models. Memory usage is a major consideration in decoding with Joshua and hierarchical grammars. Many steps have been taken to reduce memory usage, including beam settings and test-set- and sentence-level filtering of grammars. However, memory usage can still be in the tens of gigabytes.

5.3 Thrax

Thrax is an open-source SCFG extractor built on Apache Hadoop. It is able to extract syntax-augmented (Zollmann & Venugopal, 2006) and hierarchical grammars (D. Chiang, 2007) and is easily extendable to support new grammars, output formats and feature functions. How well the extractor performs depends largely on the features and options used with the base extractor.

After running numerous tests with Moses, we decided to experiment with some modifications of Joshua toolkit, to see if a better score could be achieved. To our knowledge, this was the first time Joshua had been used for the Persian-English language pair. One motivation for this was the fact that since Persian is a morphologically rich language, word disordering is a common issue that we face. Joshua takes syntax into account to some extent, with phrases being used to learn word reordering. Below follows preparation of data, experiment results and evaluation.

5.4 Moses vs. Joshua

5.4.1 Syntax Models

Hierarchical phrase-based or syntactic grammar is used in most synchronous context-free grammar (SCFG)-based MT decoders, such as in open-source toolkits Joshua, Jane or CDEC.

5.4.2 String-to-tree Models

In hierarchical phrase-based grammar model sets, non-terminals are represented by 'X':

```
X --> [source] X1 ||| [target] X1
```

Where glue rules are required to ensure an output from the decoder, the non-terminals for glue rules are represented by 'S':

```
S --> <s> ||| <s>
S --> X1 </s> ||| X1 </s>
S --> X1 X2 ||| X1 X2
```

In syntactic models, the output from a sentence parser enables non-terminals to be labelled linguistically, such as "ADJ" or "NOUN".

```
ADJ --> [source] ||| [target]
NOUN --> [source] ||| [target]
```

Although, as in phrase and hierarchical phrase-based models, the decoder input and output are in conventional string form, it is possible also to obtain context-free grammar-tree derivation of the output. Non-terminals in this CFG tree are labelled linguistically. Such models are known as 'string-to-tree' models, and are generally used by most open-source decoders.

5.4.3 Text Rule Table Format

Rule table format differs between Moses and hierarchical systems like Joshua or CDEC. Moses format is based on the Pharaoh/Moses phrase-based format, and has the following differences:

For instance, consider the following translation rule:

```
[a b c --> d e f]
```

With word alignments and probabilities:

$[a_1, a_2, ...a_n], [p_1, p_2, ...p_n]$:

In Moses, the rule is formatted as:

$a\ b\ c\ |||\ d\ e\ f\ |||\ p_1\ p_2\ ...p_n\ |||\ a_1\ a_2\ ...a_n$

In a hierarchical phrase-based system, consider the following rule:

$[X \rightarrow a\ X_1\ b\ c\ X_2\ |||\ d\ e\ f\ X_2\ X_1]$

The hierarchical (i.e., Joshua/CDEC/Hiero) format is:

$X\ |||\ a\ [X,1]\ b\ c\ [X,2]\ |||\ d\ e\ f\ [X,2]\ [X,1]\ |||\ p_1\ p_2\ ...p_n$

While the Moses format is:

$a\ [X][X]\ b\ c\ [X][X]\ [X]\ |||\ d\ e\ f\ [X][X]\ [X][X]\ [X]\ |||\ p_1\ p_2\ ...p_n\ |||\ 1\text{-}3\ 4\text{-}4$

In a string-to-tree rule such as:

$VP \rightarrow a\ X_1\ b\ c\ X_2\ |||\ d\ e\ f\ NP_2\ ADJ_1$

The Moses format is:

$a\ [X][ADJ]\ b\ c\ [X][NP]\ [X]\ |||\ d\ e\ f\ [X][NP]\ [X][ADJ]\ [VP]\ |||\ p_1\ p_2\ ...\ |||\ 1\text{-}3\ 4\text{-}4$

For a tree-to-string rule:

$VP \rightarrow a\ ADJ_1\ b\ c\ NP_2\ |||\ X \rightarrow d\ e\ f\ X_2\ X_1$

The Moses format is:

$a\ [ADJ][X]\ b\ c\ [NP][X]\ [VP]\ |||\ d\ e\ f\ [NP][X]\ [ADJ][X]\ [X]\ |||\ p_1\ p_2\ ...\ |||\ 1\text{-}3\ 4\text{-}4$

In a Joshua/Hiero/CDEC file format, in order to enable large models to be used while decoding, the text rule table should be easily convertible to on-disk binary format. This enables the use of large models even on low-memory servers. Smooth and efficient conversion into this format depends on several things: first, the sequence of each rule's source column must have matching terminals and non-terminals with the sequence to be decoded. Secondly, it is necessary for the file to be arranged so that the entries in the first column are in alphabetical order. The RHS of each rule is then scanned by the decoder for target non-terminals, and these are added to the first

column, which will then consist of source terminals and source non-terminals, and the RHS target non-terminals. In order to preserve memory space, the counts used to calculate probabilities *P(T|S) = count(t, s)/count(s)*, and also *P(T|S) = count(t, s)/count(t)* are discarded immediately after they are used. The extract file must be arranged contiguously in order to be able to do this. While this file format may be used for hierarchical models, it does not support Moses' various syntax models.

5.5 Data Preparation

IRNA was used as a monolingual corpus for training SMT translation from English to Persian. For the Persian to English translation direction we used the news commentary monolingual corpus. The IRNA corpus consisted of about 6 million sentences and was derived from the Iranian News Agency. Table 5-1 shows the number of sentences and words in each monolingual corpus.

Table 5-1: Monolingual corpora composition

Monolingual	Data Genre	Sentences	Words
News-Commentary	News	18911860	44904370
IRNA	News	5852532	66331086

The News-Commentary corpus is based on text with smaller sentence size than that of IRNA. Because of this, IRNA has fewer words, even though it has more sentences than IRNA. The test set consisted of 2000 sentences with one human translation as a reference. This same test set was used in both directions of translation.

At the time of our experiments, the only large, freely available parallel corpus available for the English-Persian language pair was the TEP corpus, developed on slang words with public domain, extracted from movie subtitles, and consisting of about 7.8M words in 5.3M sentences. This corpus, and another corpus privately obtained (MPEC-Modern Persian-English Corpus) consisting of about 50K sentences, were concatenated together to form a single corpus of about 5.4M words.

This concatenated corpus we dubbed NSPEC (News-Subtitle-Persian-English Corpus), for use in one branch of tests.

The tests used the MPEC corpus divided into sections of 20K, 30K, 40K, and 50K sentences, the NSPEC corpus, and also the TEP corpus in a separate test. Each corpus

and corpus combination was used with both Moses and Joshua toolkits, in order to obtain an accurate comparison of output.

5.6 Experiment Results and Evaluation

5.6.1 System configuration

We perform and evaluate directions of translation. For both directions, we use the default settings of Moses, that is, we set the beam size to 200 and the distortion limit to 6. We limit the number of target phrases that are loaded for each source phrase to 20, and we use the same default eight features of Moses. For the translations using Joshua, default settings are also used. Our Joshua-based experiments used the Joshua implementation of the hierarchical phrase-based algorithms. Our maximum phrase length was set to 5, and maximum MERT iterations were set to 10, with the size of N-best list at 300. The language models used were 5-gram models.

As previously mentioned, the issue of word alignment in the parallel corpus is an area needing much attention. Sentence-aligned parallel corpora are useful for the application of machine learning in machine translation; however, unfortunately it is not usual for parallel corpora to originate in this form. Since there is a great shortage of bilingual text for Persian-English, great care needed to be taken to ensure that the text that was available was the best possible quality. Several different methods are able to perform alignment. Desirable characteristics of an efficient sentence alignment method include speed, accuracy and no need for prior knowledge of the corpus or the languages in the pair. In our experiments using the Moses toolkit, we used the Microsoft bilingual aligner and later Giza ++ (Och & Ney, 2000), whereas with the Joshua toolkit, we used the Berkeley aligner (Liang, Taskar *et al.*, 2006). All the corpora used in each test, in both the Moses and Joshua experiments, were aligned on sentence level, and tokenized.

5.6.2 Results

We trained and ran each system (Joshua and Moses) on five different corpora (Table 5-2). We also used news commentary for building a language model (Table 5-1). The language model in both systems was smooth, with a modified Kneser-Ney algorithm, and implemented in SLRIM (A Stolcke, 2002). We trained language models up to 5-grams. In our Joshua tests, we used N-best list of size 300.

Table 5-2: Parallel corpora composition

Language Pair En-Pe	Data Domain	English		Persian	
		Sentences	Words	Sentences	Words
20K	Newswire	20121	353703	20615	364967
30K	Newswire	30593	465977	30993	482959
40K	Newswire	40701	537336	41112	560276
50K	Newswire	52922	785725	51313	836709
NSPEC	Newswire -Subtitle	678695	5596447	665678	5371799
TEP	Subtitle	612086	3920549	612086	3810734

We start by comparing the translations yielding the best configuration generated by both Joshua and Moses. As seen in Tables 5-3 and 5-4, for the Persian-English direction of translation, we achieve the best score with system 50K, and the BLEU score for Moses shows a better result in comparison to Joshua. The comparison of BLEU scores between Moses and Joshua is shown in Figure 5-1.

Table 5-3: BLEU scores Pe-En Joshua vs. Moses

Parallel data	Joshua	Moses
20K	0.1817	0.2655
30K	0.1795	0.2910
40K	0.1672	0.3056
50K	**0.1836**	**0.3332**
NSPEC	0.1691	0.0621
TEP	0.0252	0.1975

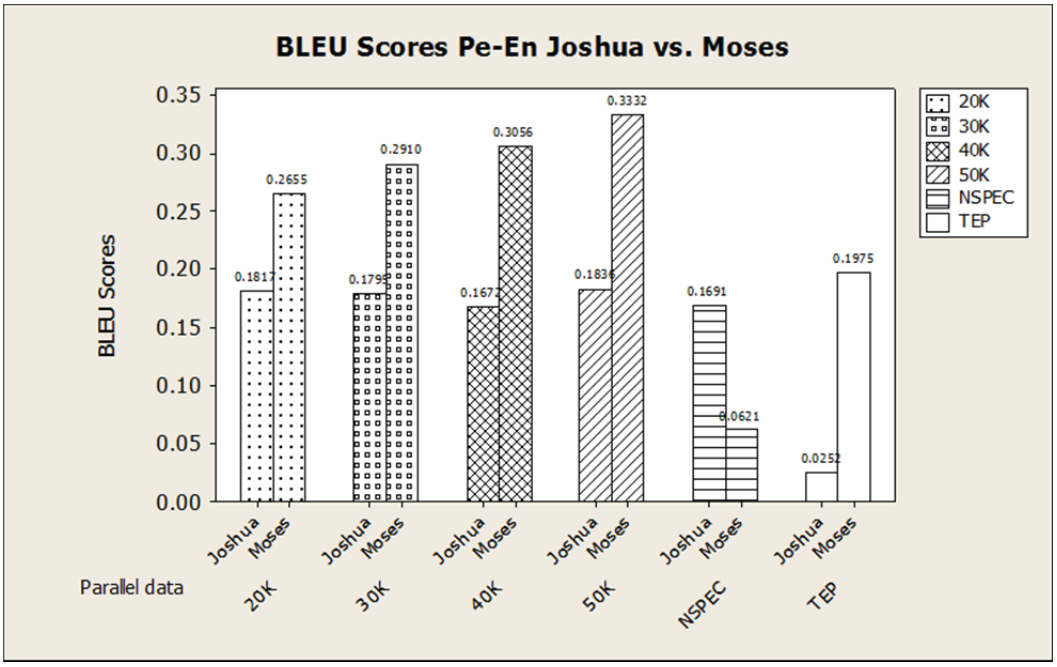


Figure 5-1: BLEU Scores Pe-En Joshua vs. Moses

The same trend is also observed in the NIST score. Table 5-4 and Figure 5-2 both show the NIST scoring differences between Moses and Joshua.

Table 5-4: NIST scores Pe-En Joshua vs. Moses

Parallel data	Joshua	Moses
20K	3.0927	3.2458
30K	3.0440	3.4425
40K	2.9694	3.7057
50K	**2.9135**	**3.8085**
NSPEC	2.8822	2.2952
TEP	1.8462	2.9907

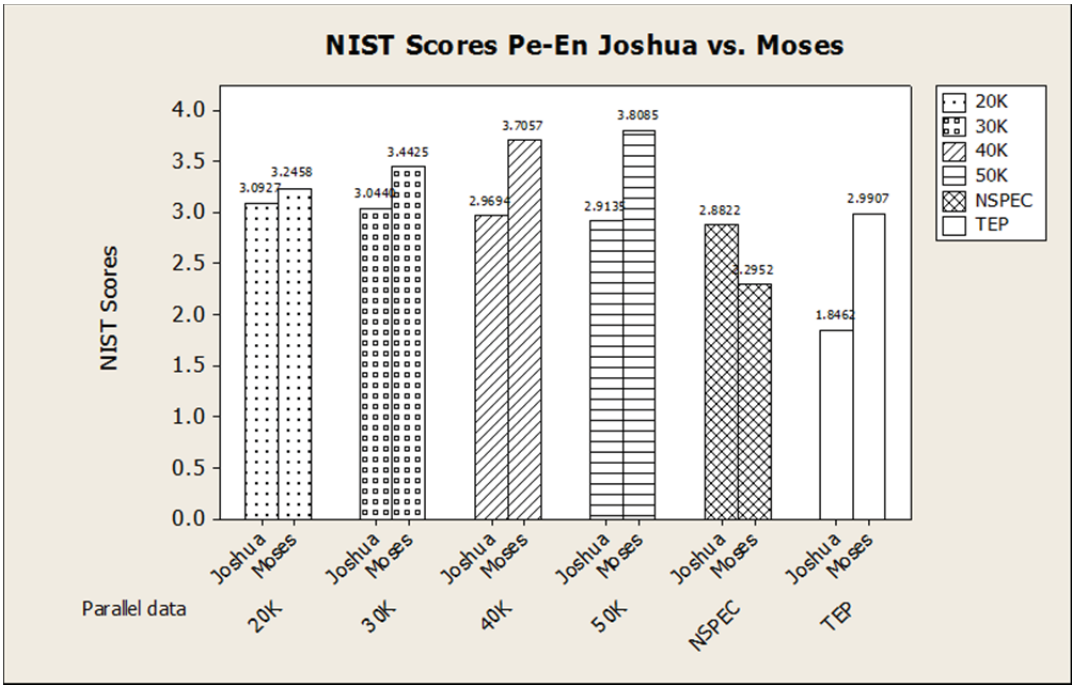


Figure 5-2: NIST scores Pe-En Joshua vs. Moses

For the English-Persian translation direction, shown in Tables 5-5 and 5-6, the NIST score for Joshua (4.5269) is slightly higher than the Moses score (4.4925). The trend is also shown similarly in the BLEU scores, with Moses scoring 0.3496 and Joshua 0.3708. These scores seem to suggest that for the Persian-English direction of translation, Joshua will yield a more accurate translation output. Table 5-6 and Figure 5-4 show the difference in the NIST score in the Persian –English direction in both Joshua and Moses.

Table 5-5: BLEU scores English-Persian

Parallel data	Joshua	Moses
20K	0.3239	0.3287
30K	0.3252	0.3215
40K	0.3411	0.3401
50K	**0.3708**	**0.3496**
NSPEC	0.2563	0.1838
TEP	0.1259	0.0535

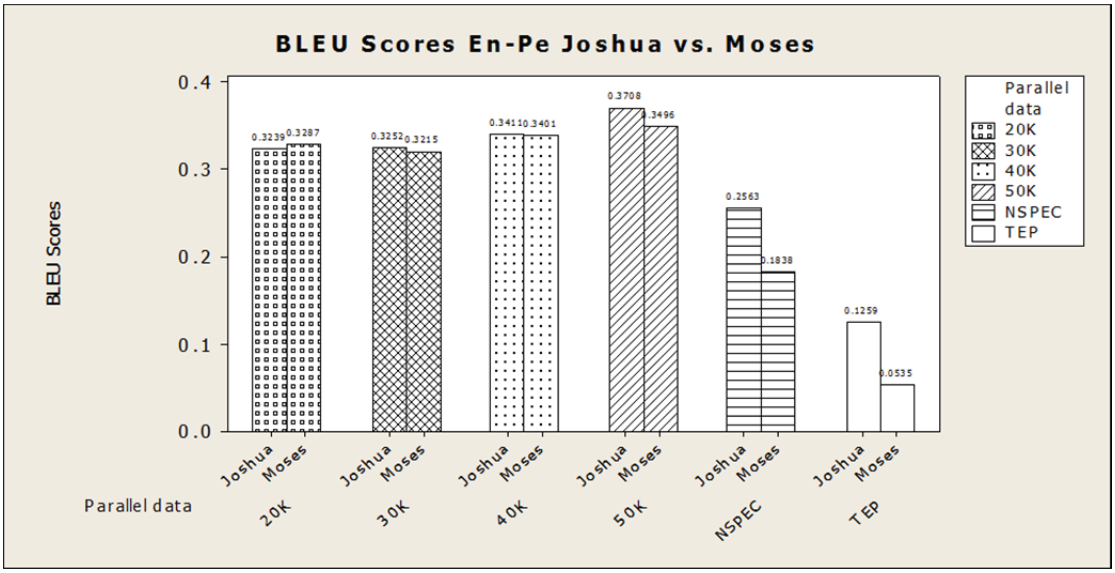


Figure 5-3: BLEU scores English-Persian

Table 5-6: NIST scores En-Pe Joshua vs. Moses

Parallel data	Joshua	Moses
20K	4.2892	4.0985
30K	4.0903	4.1409
40K	4.2362	4.2090
50K	**4.5269**	**4.4925**
NSPEC	3.1536	3.0264
TEP	2.1560	1.8830

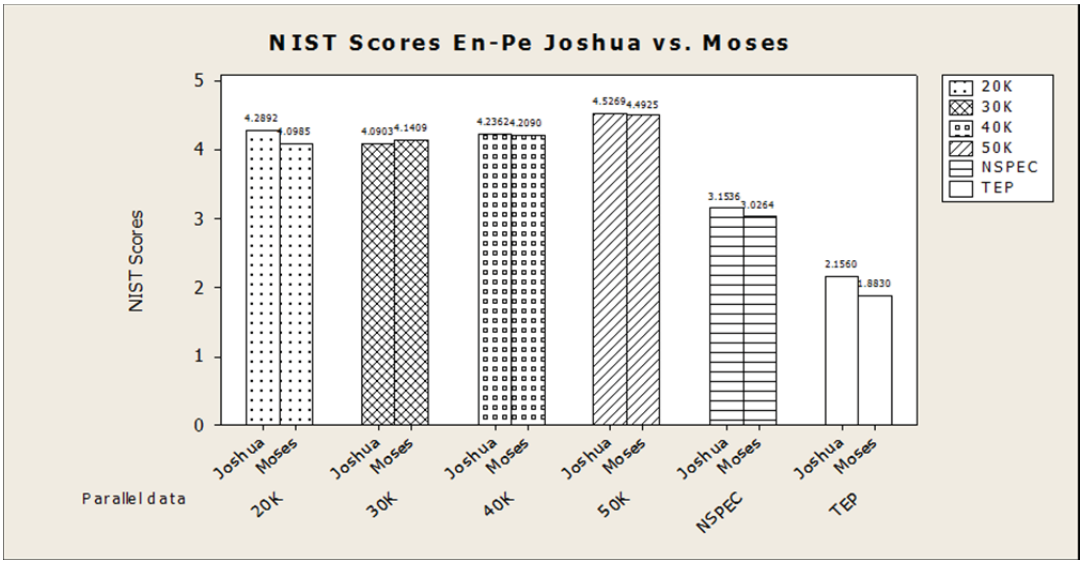


Figure 5-4: NIST scores English-Persian

One of the major differences between English and Persian is the word order. As previously mentioned, Persian as a target language possesses some features that negatively affect MT performance: it is rich in morphology, much more so than English, and there is greater noise in training data, and harder sparse-data problems due to vocabulary that combines words from various sources. Its richness in morphology means that if Persian is the target language, the SMT system must not only select a lexically correct Persian equivalent of an English word, but must also correctly guess grammatical features. Therefore, significant reordering must take place during translation. Hierarchical phrase-based translation is based on synchronous context-free grammars (SCFG). Like classical phrase-based translation, pairs of corresponding source and target language phrases (sequences of tokens) are learned from training data. The difference is that in hierarchical models, phrases may contain “gaps”, and are represented by non-terminal symbols of the SCFG. If a source phrase contains a non-terminal, then the target phrase will also contain that non-terminal, and the decoder can replace the non-terminal by any source phrase and its translation, respectively.

This follows the observation that hierarchical models have been shown to produce better translation results than classic phrase-based models (D Chiang, 2005).

As far as automatic evaluation is concerned, the best result from these tests is 4.5269 NIST and 0.3708 BLEU using the Joshua-based system trained on the 50K corpus. Moses was not able to outperform these scores, despite its ability to learn factored

models. The best Moses score in these tests was 4.4925 NIST and 0.3496 BLEU. The Moses and Joshua systems were trained under identical conditions, in order to ascertain which decoder yielded better performance in which translation direction: both the translation model and language model are trained on the same monolingual corpus (IRNA) for the English-Persian direction, and news commentaries for the Persian-English direction.

There are some significant and somewhat unusual differences in the output of a conventional phrase-based model and a hierarchical model when working with the English/Persian language pair. We can conclude that adding more training data to the system for both translation directions either helps significantly, or (more often) brings down the BLEU score. Both BLEU and NIST scores improved when we trained with Joshua in the English-Persian direction, whereas Moses performed better in the Persian-English direction.

It is concluded that the hierarchical decoder Joshua surpasses Moses in its ability to capture word order. This is confirmed by Joshua's consistently higher results in the English-to-Persian translation direction.

5.7 Joshua 4.0

Due to release of Joshua 4.0 during this research, we ran new experiments using Joshua 4.0, and compared the output to that of the earlier version.

In the latest version of Joshua (4.0), the main changes include further implementation of Thrax, which enables extended extraction of Hiero grammars, and a modified hypothesis exploration method (Ganitkevitch, Cao *et al.*, 2012).

In Joshua 4.0, an SCFG can be represented as a set of rules given as:

$$C_i \rightarrow < \alpha_i , \gamma_i , \sim_i , \varphi_i > \qquad \textbf{5.4}$$

where C_i is a non-terminal symbol of the grammar, α_i and γ_i are sequences of terminal and non-terminal symbols for the source and target sides, respectively, $\sim_i$ is a correspondence between the non-terminals of α_i and γ_i, and φ_i is a feature vector defining the probability of translation from α_i to γ_i.

These rules are loaded to the memory decoding run time, indexed by the source-side and stored in a trie data structure. In this way, the decoder is able to access the rules for a certain span of input. Joshua 4.0 implements a packed trie representation for the SCFGs, instead of basing the structure on hash maps as is commonly seen in other

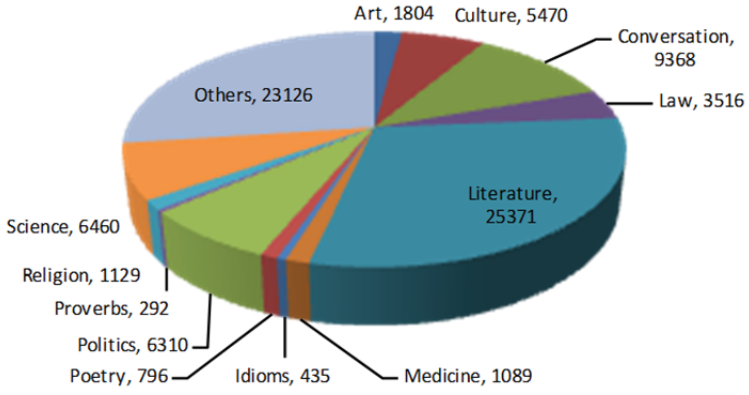
systems. In this way again, the issue of memory space for grammar rules is further minimized. This method is similar to work in phrase table storage by Zens and Ney (2007).

5.7.1 Experiments and Results

5.7.1.1 Baseline SMT

This section covers the tests comparing Joshua versions 4.0 and 1.3. In these tests, the default settings of Joshua 4.0 were used. The parallel corpus used for the training set was based on the NSPEC corpus but, as noted earlier, the subtitle addition to the corpus adversely affected the output result, since it was composed of highly colloquial sentences and phrases which presented significant inconsistencies in the translation output. Because of this, we removed the subtitle addition. The final corpus (named NPEC) consisted of almost 85,000 sentence pairs of 1.4 million words, originating mostly from bilingual news sites.

There are a number of different domains covered in this corpus, but the majority of the text was in literature, science and conversation. Figure 5-5 shows the NPEC corpus divided into separate domains.



Genre	#Sentences
Art	1804
Culture	5470
Conversation	9368
Law	3516
Literature	25371
Medicine	1089
Idioms	435
Poetry	796
Politics	6310
Proverbs	292
Religion	1129
Science	6460
Others	23126
Total	85166

Figure 5-5: NPEC Corpus composition

The language model used in the tests was extracted from IRNA[7] website, covering news stories, and comprised over 66 million words. The details of the components of the baseline system are shown in Table 5-7.

Table 5-7: Baseline System Components

	English		Persian	
Training Set	Sentences	83042	Sentences	82496
	Words	1322470	Words	1399759
Tunings Set	Sentences	1578	Sentences	1578
	Words	40044	Words	41287
Language Model	Sentences	5852532		
	Words	66331086		

5.7.1.2 Test Data Set

We used eight test sets based on text extracted from certain bilingual websites for our experiments. We perform translation in the English-Persian translation direction. The Persian side of the test sets was used as translation reference when using scoring metrics to evaluate output of both the baseline system and the post-APE output.

The size of test data varies from one paragraph of text to a complete page. The number of sentences in both sides is equal. The composition of the test sets, together with their genres, is shown in Table 5-8.

Table 5-8: Statistics of eight test sets used in automatic and manual evaluation

Testing set data	English		Persian		Genre
	Words	Characters	Words	Characters	
set #1	163	878	158	551	Culture
set #2	218	1381	222	955	Art
set #3	371	1941	403	1663	News Stories
set #4	362	1922	337	1230	Religious
set #5	101	589	115	430	Medicine
set #6	354	1887	386	1717	Politics
set #7	555	2902	653	2551	Economic
set #8	259	1325	297	1063	Literature
Total	2383	12825	2571	10160	

[7] http://www.irna.ir/ENIndex.htm

Table 5-9 shows both the BLEU and NIST scores for Joshua 1.3 and Joshua 4.0 over 8 test sets, highlighting the difference in scores between each version of Joshua.

Table 5-9: Difference of BLEU and NIST Score after using Joshua 4.0 on eight test sets

Input	Joshua 1.3		Joshua 4.0		BLEU Difference	NIST Difference
	BLEU	NIST	BLEU	NIST		
#1	0.4804	4.9046	0.6523	6.5740	0.1719	1.6694
#2	0.0113	0.6690	0.0159	1.1239	0.0046	0.4549
#3	0.1741	2.7099	0.5914	6.1083	0.4173	0.4173
#4	0.0103	0.6529	0.0101	0.7962	-0.002	0.1433
#5	0.0600	1.6569	0.7925	5.7332	0.7325	4.0763
#6	0.0223	1.6056	0.0302	1.8922	0.0079	0.2866
#7	0.0126	1.3370	0.0193	2.0457	0.0067	0.7087
#8	0.0218	1.7715	0.0719	2.6759	0.0501	0.9044

In the default configuration file for Joshua 4.0, the set evaluation metric is Multi-BLEU. In the Multi-BLEU scoring metric, the translation output of a system is measured against reference translations. This is done by summing the 4-gram, tri-gram, bi-gram and uni-gram matches, and dividing this number by the number of the same found in the reference translation set. Like BLEU, scoring is between 0 and 1.0, with 1.0 being the highest (best possible) output (K. Papineni, Roukos *et al.*, 2002).

Table 5-10 shows the Multi-BLEU scores over 8 different test sets. At the time, it was understood this was the first time Joshua 4.0 had been used for the Persian/English language pair. The result from test set 8 yielded the highest Multi-BLEU scores achieved in this study. The improved scores were attributed to the new features in Joshua 4.0.

Table 5-10: Multi–BLEU Joshua 4.0 on eight test sets

Test Sets	BLEU	P_1	P_2	P_3	P_4	BP
Test set #1	**0.7761**	0.8629	0.7976	0.7516	0.7013	1.0000
Test set #2	**0.0245**	0.4355	0.0837	0.0130	0.0020	0.7819
Test set #3	**0.6244**	0.7712	0.6512	0.5859	0.5314	0.9929
Test set #4	**0.0091**	0.1532	0.0095	0.0037	0.0014	0.9752
Test set #5	**0.8072**	0.9266	0.8632	0.8519	0.8358	0.9292
Test set #6	**0.0548**	0.3496	0.0902	0.0381	0.0145	0.8489
Test set #7	**0.0207**	0.3123	0.0571	0.0149	0.0007	1.0000
Test set #8	**0.0779**	0.4150	0.1250	0.0489	0.0198	0.9248

5.8 Multiple References

Human translation may differ from system output for any given sentence, as different professional human translators may choose different words or even different phrases to accurately represent a source language sentence in that of the target language. Since evaluation metrics operate by comparing system output with a human-translated reference sentence, in some cases, a decrease in score may incorrectly represent a translation as poor quality. This is avoided when the evaluation metric is run using multiple [different] reference sentence translations, which increases the likelihood of a system output word or phrase that is indeed correct, matching an equivalent in one of the reference translations. Where multiple references are used, it is necessary to store them in distinct separate files, to prevent cases such as a poor system translation being evaluated higher than it actually is due to incorrect combination of words across both or all reference translations.

This arrangement of multiple referencing was used to evaluate test set 5, using both BLEU and NIST metrics on Joshua 1.3 and 4.0 based systems. The data shown in Tables 5-11 to 5-14 shows a significant increase in metric scores over all evaluation metrics when multiple references are used.

Table 5-11: Multiple-reference BLEU/NIST scores for Joshua 1.3-based system output

SMT Output	Reference 1	Reference 2	Reference 1 & 2
BLEU	0.4972	0.3483	**0.6148**
NIST	4.8334	4.6662	**6.7231**

Table 5-12: Multiple-reference Multi-BLEU scores for Joshua 1.3-based system output

SMT Output	Reference 1	Reference 1 & 2
MULTI-BLEU	0.4327	**0.6175**

Table 5-13: Multiple-reference BLEU/NIST scores for Joshua 4.0-based system output

SMT Output	Reference 1	Reference 2	Reference 1 & 2
BLEU	0.5672	0.2842	**0.6523**
NIST	4.9467	4.1775	**6.5740**

Table 5-14: Multiple-reference Multi-BLEU scores for Joshua 4.0-based system output

SMT Output	Reference 1	Reference 1 & 2
MULTI-BLEU	0.7013	**0.7761**

5.9 Summary

In this chapter, the use of the hierarchical phrase-based decoder, Joshua, was investigated and compared with Moses, the standard phrase-based decoder also used in this project. It is shown that a hierarchical phrase-based system is able to outperform a standard phrase-based system in the English-Persian translation direction due to the decoder's ability to perform phrase reordering, by using SCFGs to construct hierarchical phrases from words and sub-phrases, and efficiently capture both word order, and long-distance phenomena. This is important especially for Persian, because of the language's standard SOV (subject-object-verb) sentence structure. Joshua's latest version (4.0) is also compared to Joshua 1.3, with improved results for the most part. Also shown is the increased accuracy of evaluation while using multiple referencing, where translation output is evaluated against two different references.

Chapter 6. APE - Automatic Post-Editing System: Background

"You have to learn the rules of the game. And then you have to play better than anyone else."

~ Albert Einstein

6.1 Introduction

Machine translation output is often seriously grammatically flawed. This is more often the case with SMT than with other approaches due to the absence of linguistic rules for the language pair on which it is being applied, and is most prevalent in translation of long-distance phenomena. Grammatical error not only weakens the fluency of translated language, but, in certain cases, can completely change the meaning of a sentence. In morphologically-rich languages, grammatical accuracy is very important, as the interpretation of syntactic relations depends heavily on morphological agreement of sentences. Since our main system's approach is SMT, and deals with Persian, a morphologically-rich language, post-editing translation output is an important step in maintaining the fluency of the translation. Since most mistakes associated with machine translation are repetitive, the task of post-editing can be made automatic (Allen & Hogan, 2000). When repetitive errors occur, the system can be ruled to correct known mistakes. Furthermore, the process of Automatic Post-Editing (APE) is very similar to a machine translation process (Simard, Goutte *et al.*, 2007b), and because of this, certain MT systems can be used to model the APE process.

6.2 Motivation for an APE Approach

After the statistical machine translation system was designed, various tests were run to determine parameters affecting the system's output, such as the effect of corpus size, the impact and importance of corpus domain, and different alignment techniques were experimented with. These tests showed that certain combinations yielded significant improvement in translation output. After determining the best system arrangement, initial hybrid techniques were examined, details of which will be discussed later in this chapter.

It was necessary to experiment with several different techniques to deal with word order differences encountered when using an SMT approach, and overcome alignment

errors because of the differences between English and Persian. Two alignment methods were tested, the results of which were reported in the previous chapter.

Although significant improvement was made compared to the initial tests, it was observed that the evaluation metric scores were still unsatisfactory, and the output of the system still suffered from OOV words (out-of vocabulary), which remained untranslated. Also, because of the difficulties faced while dealing with the requirements of linguistic knowledge, such as morphology, syntactic functions and word order, all of which led to a loss of accuracy, it was decided it was necessary to investigate implementation of an APE module based on a Hybrid Machine Translation (HMT) approach.

In many cases, two approaches may be combined in various ways to create a hybrid system where one approach's strengths complement the other's. Such a hybrid system may be constructed with a number of different orientations and architecture, and is generally referred to as a Hybrid Machine Translation approach.

A Hybrid Machine Translation approach integrates the core of the engines of existing approaches such as Rule-Based Machine Translation (RBMT), Statistical Machine Translation (SMT), and Example-Based Machine Translation (EBMT). An HMT system combining the advantageous characteristics of each individual approach is defined as a multiple-engine HMT system.

The advantages and disadvantages of RBMT and SMT approaches may be summarised as follows: RBMT is strong in syntax, morphology, structural semantics, and lexical reliability, but demonstrates weakness in the areas of lexical semantics and lexical adaptivity. SMT, while being weak in areas of syntax, morphology, and structural semantics, is superior to RBMT in areas of lexical semantics and adaptability, although the advantage of adaptability to other language pairs is only valuable when the system is to be used with a wider range of languages. However, in the case of focussed development in one language pair, lexical reliability is more important, and, in this area, RBMT is superior.

RBMT translation approaches are categorized into three types: Direct Systems (such as Dictionary-based Machine Translation) that map input to output with basic rules; Transfer RBMT Systems (Transfer-based Machine Translation) that employ morphological and syntactical analysis; and Interlingua RBMT Systems (Interlingua) that use an abstract meaning. The APE system's main algorithm follows a Transfer-based approach. Transfer-based MT is among the most commonly-used approaches

for MT. This method involves capturing the meaning of a source sentence using intermediate representations and, from it, generating a target output. One advantage of the transfer-based approach is the analysis step, since the analysis representation step tends to become more abstract the deeper the linguistic analysis goes (Mohamed, 2000). Generally, RBMT approaches translate very strictly, representing each individual segment of the input; however, their effectiveness is somewhat limited in transfer by lexical selection. SMT and EBMT are two empirical methods. SMT proves to be a more robust approach, and because of its use of language and translation models, together with better lexical selection, is capable of providing fluent translation. However, both approaches lack linguistic knowledge, which is somewhat indispensable when it comes to grammar correction.

6.3 Related Work

Simard, Goutte *et al.* (2007b) published a paper in which they describe their utilisation as a phrase-based SMT system for use in post-editing. APE, which is suggested in Simard, Goutte *et al.* (2007b), is built on phrase-based statistical machine translation, with a parallel bilingual corpus. The corpus they used in this research was based on raw output from RBMT together with accurate human translation of the same text. Their work is based on the fact that most mistakes associated with machine translation are of a repetitive nature and, therefore, the task of post-editing can be made automatic. They showed, as far as repetitive errors occur, the system can learn post-editing rules from a tri-parallel corpus of source data, raw machine translation output, and post-edited text (Allen & Hogan, 2000). They show that the output of their automatic post-editing system (APE) is better than that of RBMT and SMT alone.

Their research was mainly motivated by certain legislation from the Canadian government's Department of Human Resources and Social Development (HRSDC). HRSDC operates a website called Job Bank[8], which lists advertisements for over a million jobs. By legislation, all these advertisements must be in both French and English. To address the task of providing advertisements in both languages, HRSDC make use of machine translation in addition to 20 full-time human post-editors, most of whom are junior translators. The authors' goal was to decrease the task of post-

[8] http://www.jobbank.gc.ca/intro-eng.aspx

editing compared to the TER score. They show that the process of APE is very similar in nature to a machine translation process, therefore, they used Portage, developed at the National Research Council of Canada (NRC) (Sadat, Johnson *et al.*, 2005) as the statistical machine translator to perform APE's task in this research. The system translates text in three main phases: pre-process and tokenise the data; decoding the data in order to produce one or more translation hypothesis; finally, error-driven re-scoring to choose the best final hypothesis. Portage's probability model is a log-linear combination consisting of four main components. One or more n-gram target language models, one or more phrase-based translation models, a distortion (word-reordering) model, and a sentence-length feature. This phrase-based translation model is similar to that of Koehn, Och *et al.* (2003a), with the exception that Portage's phrase probability estimates are smoothed using the Good-Turing technique (Foster, Kuhn *et al.*, 2006), and that a final cost is added to account for sentence endings.
The authors conclude by showing that using a phrase-based machine translation system to automatically post-edit the output of another machine translation system not only greatly improves the translation output quality, but outperforms a stand-alone phrase-based translation system.

Lagarda, Alabau *et al.* (2009) present an automatic post-editing (APE) system that can be added to a commercial rule-based machine translation (RBMT) system in order to produce a better quality translation output. For APE, they used a statistical machine translation (SMT) system to enhance the output from the RBMT system. They also propose a new human evaluation measure, to estimate productivity increase. A desired feature of this measure was to impose less effort on the human evaluator. This was accomplished by determination of translation suitability, that is, whether or not a translation is suitable for post-editing with little effort at that stage, or whether the whole translation needed to be discarded and started over. They demonstrate their APE system which is tested with two different corpora of different complexity.
One corpus used in this study was the Parliament corpus, which is a collection of documents compiled from the proceedings of parliamentary sessions, collected by a client of the translation agency involved in this work. The Protocols corpus is a collection of medical protocols which is another corpus used in the study. They show that this corpus was more difficult to compile and work with, given such factors as the different companies involved with the training and test sets, and some out-of-domain test data.

In their results they show that with the Parliament corpus, the use of APE can improve the RBMT system output by an average of 59.5%, based on better performance in real translation scenarios. When used with the Protocols corpus, the improvement was not as significant, but still at an average of 6.5% improvement. The use of the suitability evaluation also proved to be promising, with 94% of Parliament-based and 67% of Protocols-based translations being evaluated as suitable.

The APE in this research is a process carried out between the stages of output of RBMT and end result – translated text in the target language. The rule-based machine translation system in this research is based on a commercial RBMT, and the APE is an SMT system based on the open-source toolkit Moses. The SMT system is trained by receiving the output of RBMT as input (source) and producing translation in the target language.

Isabelle, Goutte *et al.* (2007) reported the results of a set of experiments about the use of a statistical machine translation system as an APE module for addition to RBMT. They showed that the performance of RBMT systems can significantly improve by adapting the system to a specific domain.

They showed that system adaptation can enable machine learning from post-editing by human translation. For this study, they use Portage's system for the APE component.

Portage's probability model is a log-linear combination consisting of four main components: one or more language models in the target language, one or more translation models for source language, one distortion model and, finally, a sentence length feature. In this research also, HRSDC supplied Portage with data from *Job Bank* for their research. This data is in corpus format, and consists of blocks of data. Each block has four parts: first, source language, secondly, a translation (T_1) from source language (produced from a commercial RBMT system), thirdly, a second translation (T_2) from source produced from a customised dictionary from the same RBMT (using domain-specific dictionary, manually developed), and, finally, a reference translation (T_R) which is manually (humanly) post-edited.

In this paper they document the results of a set of experiments covering the use of phrase-based SMT technology used in the building of an APE module for RBMT systems.

The *Job Bank* data that was used included source language texts together with two different RBMT translations, as well as a manually post-edited reference translation.

One of the two RBMT outputs was a result of the vanilla version of the system, and the other output used manually-developed domain-specific dictionaries.

At this stage, Portage's SMT system was used in the post-editing of the outputs into the reference translation language. They also use Portage's SMT system directly in translation from source to target language (without APE module). TER metric was then used to evaluate translations of these different system configurations. In their results, they show that TER of the adapted version of the RBMT system was 10% lower (yet still above 50%), the French-English Portage translation results were significantly better than those of the adapted system (in spite of a small training set), but TER scores showed that the combination of the RBMT system with the Portage SMT system as the APE module yielded the best result.

This research shows that using a bigger corpus can make a significant difference between a raw translation and an edited translation. The APE layer is able to automatically extract all useful information from an existing dictionary. Training this APE layer has a fixed cost regardless of what the main translation system is. In this paper, Isabelle, Goutte *et al.* (2007) also argue that human post-editors can sometimes examine the source language to obtain a better translation. They argue that this same approach could be utilised by an APE system.

de Ilarraza, Labaka *et al.* (2008) investigate an SPE system involving the use of an SMT system for post-editing an RBMT system's output. They employ this system for the low-resource Basque-English language pair. Since corpus-based machine translation requires large bilingual corpora, and the accuracy of the translation output relies significantly on the size of the corpus, the authors started to build a parallel corpus covering different domains. They succeeded in constructing this corpus to a size of almost 3 million words. They show that the domain adaptation technique for Basque-English machine translation and the combination of RBMT and SPE, will still yield promising results when used in a restricted domain, despite a training corpus size of only half a million words. The goal of this research was to improve the accuracy of this machine translation system while restricted to a specific domain. Their results confirm the improvements proposed when using the specific domain, but not so when applied to other domains, or as a generic cross-domain system. Their experiments are, by nature, different when compared to other work in this area. This is due to the use of the morphological component in RBMT and SMT systems, and also due to the scarcity of available corpora for this language pair. Their results show

phenomenal improvement in BLEU scores when using RBMT + SPE in the specific domain. The same method results in a smaller improvement when using a general corpora, as is also presented by Terumasa (2007) and Simard, Goutte *et al.* (2007b). They could not present the result with manually post-edited corpora as Isabelle, Goutte *et al.* (2007) and Simard, Goutte *et al.* (2007a) had, since there was no corpus of large enough size for that language pair at the time.

A. H. Pilevar (2011) uses an existing RBMT system and SMT for post-editing for the English-Persian language pair in order to improve the translation of subtitles for movies. They build an SPE module to edit the output of RBMT in order to adapt it to a new domain. The RBMT system used in this research translates from English to Persian, and the SMT system was trained on a bilingual corpus. The goal of this research was to change the domain of the output of the RBMT system in order to produce film subtitles in Persian. When given the same input, MT will produce the same output and, therefore, will encounter the same mistakes. The English-Persian RBMT system used in this work employs synchronous tree adjoining-grammars (S-TAG) in order to improve the connection of languages with greatly differing characteristics. It also uses a classical transfer system consisting of three main components: first, analysis and ordering of the source language into a tree structure, secondly, transfer from source language to target language structure, finally, producing the output in the target language. The SMT system and SPE module both use the Moses toolkit (P Koehn, H Hoang *et al.*, 2007) which automatically translates the output of RBMT system into the language of the reference translation. The parallel corpus used in this project is extracted from movie subtitles, consisting of 150,000 sentences and over 4 million tokens in the language pair. In order to undertake post-editing, a language model is required. A language model for SPE requires a monolingual corpus, as opposed to the parallel corpus required by a translation model. The monolingual data used for construction of the language model for both SMT and SPE are the same. The monolingual corpus built consists of over 10 million tokens from movie subtitles. The SMT system is trained with a parallel corpus of movie subtitles. The English side of the same parallel corpus was given to the RBMT system as input, in order to produce a translation. The SPE system used this Persian output as the input to produce a better translation.

The SPE system is trained based on the Persian output of the RBMT system (as the source language), and the Persian side of the training data as the target language. To

evaluate the performance of SPE, the data were first used in a translation using the RBMT system, and then passed to the post-editing module. The author then used two evaluation metrics – BLEU and TER – for a comparison of the results in the different systems. The results show that the SPE module can improve the performance of the RBMT system's output when used in a new domain. However, the use of the SMT system alone yields a better result compared to the combination of RBMT and SPE. The author notes that this result is the opposite of what is encountered in MT systems with different language pairs (the RBMT + SPE module system configuration usually outperforms SMT alone). They propose that this result is due to the significant differences between spoken and formal Persian. These differences encompass a number of linguistic areas, including sentence structure, syntax and morphology.

Béchara, Ma *et al.* (2011) study the impact of SPE on a phrase-based statistical machine translation system (PB-SMT). They use PB-SMT for the initial translation component, and also in the post-editing module. The authors claim that theirs was the first attempt to fully design and analyse a full SPE pipeline using SMT approaches for both initial MT and also for post-editing. Simard, Goutte *et al.* (2007b) briefly present this approach, proposing, however, that such an arrangement would not yield useful results. Oflazer and El-Kahlout (2007) document their attempts with such an arrangement, but did not observe significant improvement in the output.

The authors of the above work demonstrate that although a straightforward approach to using SPE for the task of post-editing in a pipeline system may lead to only minor improvement in output accuracy, if source-context modelling and thresholding are used together, this combination can yield a significant improvement in comparison with a baseline system. This approach improved the BLEU metric scores by two points.

In their experiments, they use the original SPE design as used by Simard, Goutte *et al.* (2007a), where the output of the MT system is used to train a monolingual system to be used in the APE stage (i.e., used to correct or improve the output of the initial MT). However, unlike Simard, Goutte *et al.* (2007a), instead of using RBMT for the initial MT system, followed by an SPE module, they use a PB-SMT approach for both stages (MT and APE). This is the same approach attempted by Oflazer and El-Kahlout (2007).

The goal of this research is to investigate to what extent PB-SMT technology can be used in an APE module to improve initial translation using a PB-SMT system. The

methodology in this paper is focussed on the English-French language pair, which was originally studied by Simard, Goutte *et al.* (2007a). They use the standard Moses PB-SMT toolkit, with a 5-gram language model and Kneser-Ney smoothing trained with the SRLIM toolkit (A Stolcke, 2002), the Giza++ implementation of IBM word alignment model 4 (Och & Ney, 2003) and refinement and phrase extracting heuristics described by Koehn, Och *et al.* (2003b).

The data used in these experiments originated from the IT Company *Symantec* as an English-French Translation Memory. The data domain is technical software user help information. The total training data which was extracted for training purposes was about 52,000 English-French segment pairs. The average segment length in a training set is 13 words for English, and 15 words for French. The training set has a vocabulary size of more than 9,000 words for the English side of the data, and 12,000 words on the French side. The SPE architecture follows the original post-editing design in Simard, Goutte *et al.* (2007a) and, like Oflazer and El-Kahlout (2007), used PB-SMT for both stages in the post-editing pipeline. In PB-SMT, in the first stage, the machine was trained on a parallel corpus of English and French, and produces an intermediate output (F'). This output (F') is then used in the second stage of the system. In order to avoid translating "used" data in the building of the monolingual training section (F'), the source training data (F') for the second-stage monolingual PB-SMT system (the APE module) was obtained by training another English-to-French PB-SMT system using a 10-fold cross-validation approach. The output of this system (F') is then used in the training of the PB-SMT system in the SPE module. Here, the authors note that this SPE module is somewhat disconnected from the source text, and observe the advantage of being able to ascertain whether the output of the 10-fold cross-validation PB-SMT system is an acceptable translation, or whether it is necessary to be passed through to the SPE module. For some of their experiments, the authors essentially "create" an intermediate language by concatenating the word or phrase output of the initial system (F') with its source counterpart in the source language (coded as E). This is done in an attempt to preserve the context of the text. Concatenation is accurately performed using GIZA++ word alignments. This intermediate language text is then passed as source text to the second-stage context-aware SPE system.

Despite this approach preserving the text context information, the training set vocabulary size increases from about 9,000 to 70,000 (on the English side), and the

word alignment data used are not always reliable. The authors note that the vocabulary set size increase results in sparsely-spread data, and, due to this, they risk a reduced translation output quality. In an attempt to counter this, they experiment with thresholding context information, as shown in some of their experiments.

Ahsan, Kolachina *et al.* (2010), document work using a modular RBMT system which is able to combine the two translation approaches at different stages in the RBMT system's pipeline. In this way, exploration of rules for both local and long-distance reordering was able to be performed independently, and such reordering leading to improved translation output could be identified and utilized. They show an increase in the output score for each stage of combination.

Marecek, Rosa *et al.* (2011) report experimental work in correcting the output of an English-Czech MT system by performing several rule-based grammatical corrections on sentences parsed to dependency trees. Their post-processing system, DEPFIX (Rosa, Marecek *et al.*, 2012), is somewhat different from common approaches; with a statistical system being used to post-process rule-based translation output. Their baseline SMT system relies on Moses, a phrase-based translation tool. The two-step translation is a set-up in which, first, the English source is translated to simplify Czech, and secondly, the simplified Czech is monotonically translated to fully-inflected Czech. Both steps are based on simple phrase-based models. To implement the post-processing component DEPFIX, MT outputs were POS-tagged and then parsed with MST Parser (McDonald, Pereira *et al.*, 2005) after they were trained on the Prague Dependency Treebank (Hajic, 2005), to reflect correct Czech sentences. Along with the dependency trees of MT output sentences, they have used the dependency tree of the source sentences. Hence, they also consider the dependency relations and morphological categories of their English counterparts in the input sentence in their rules. The input of a rule is the dependency tree of the MT output together with its source sentence (i.e., MT input along with the nodes aligned, where possible.)

They test the proposed system on two sets of data, WMT'10 and WMT'11. The improvement was quite different in both test sets, with WMT'10 gaining a 0.21 improvement in BLEU score, and WMT'11 gaining an improvement of only 0.05. The authors propose that this variation is due to the data difference. They also run a manual evaluation in which two annotators evaluate the output of DEPFIX and determine whether pre-DEPFIX or post-DEPFIX MT outputs are more accurate.

Their result shows that approximately 60% of the sentences run through DEPFIX were improved, and only about 20% were worse. They argue that although DEPFIX is unable to correct a number of serious MT errors, such as incorrect lexical choices, it can improve the grammar of the output in a way that the language model is often incapable of doing, leading to output that is considered better by humans.

Rosa, Marecek *et al.* (2012) enrich the rule set of DEPFIX and use a modified version of MST Parser. Their results show both modifications led to better performance of DEPFIX 2012, however, they mention that since the effect DEPFIX has on the output BLEU score is not significant, the results are not as reliable as results obtained through manual evaluation.

E. Wehrli, Nerima *et al.* (2009) present a proposal for the development of an MT model that is capable of multilingual translation, specifically between English, German, French and Italian. This work closely follows previous work by the author, with the use of the Fips Parser (Eric Wehrli, 2007). For software realization, the author proposes the use of an object-oriented design, and an abstract level of syntactic representation for the transfer level. Object-oriented design is utilized, which enables language-pair specific properties to be catered for with extension of type and redefinition of methods.

The standard transfer system pattern is used to model this system's main algorithm. First, the Fips Parser is used to parse input sentences, the output of which is a phrase-structure representation that is information-rich and linked with predicate-argument representations. Next, the source-language representation is mapped to a target-language representation by the transfer module. Mapping is accomplished by passing over the phrase structure (head, RH sub-constituent, LH sub-constituent) in the source language. Lexical terms in the source language are mapped to terms in the target language at the head-transfer level. This process gives equivalent terms in the target language which are frequently in the same category as the source-language. Target-language sentence structure is determined by the lexical head in the source language.

Alegria, De Ilarraza *et al.* (2005) present an MT system using a transfer-based approach. The transfer module used in this transfer-based system uses words (nodes), phrases, and sentences. Lexical transfer is executed with the use of a parallel corpus compiled to form a finite-state transducer. Information is then transferred between phrases during the structural transfer at sentence level, with verb chains proving to be a more complex transfer process than noun chains.

Grammatical relations may be represented by dependency structures. Compared to syntactic trees, dependency structures are more specific in regard to semantics rather than their strict word order. In a sentence dependency tree, words and relations are graphed, with each word either modifying or being modified by another word, and the root in each tree is the only word which does not modify any other word. The 'parent-child' relationship is often used to describe modifier and modified words in the tree (Ambati, 2008).

Shirko, Omar *et al.* (2000) present work on a transfer-based MT approach, this approach is built on the rule-based approach to MT, and is a very popular choice of method for many different language pairs. This method involves capturing the meaning of a source sentence using intermediate representations, and from it generating a target output. One advantage of the transfer-based approach is the analysis step, since this tends to become more abstract the deeper linguistic analysis goes.

6.4 Other Hybrid Approaches

According to Xuan, Li *et al.* (2011) the most popular combinations of MT systems and APE modules are RBMT + PB-SMT, and RBMT + EBMT. Figures 6.1 – 6.4 show various different post-editing architectures. One hybrid system proposed by Eisele, Federmann *et al.* (2008) is a phrase-based SMT + RBMT configuration. In this system, the exact translation from the RBMT output is used to build phrase tables, which are then combined. An SMT decoder is then used in generating the final output translation. Several problems or difficulties arise with the basic configuration of this system, which could be improved with modification. In order to maximise performance, multiple RBMT components are necessary, and yet, often, useful structures generated by the RBMT system are not utilised fully by the SMT component. Because of this, it was decided to use a combination of SMT as the main system, and RBMT as the APE module.

Figure 6-1: Syntactic Selection

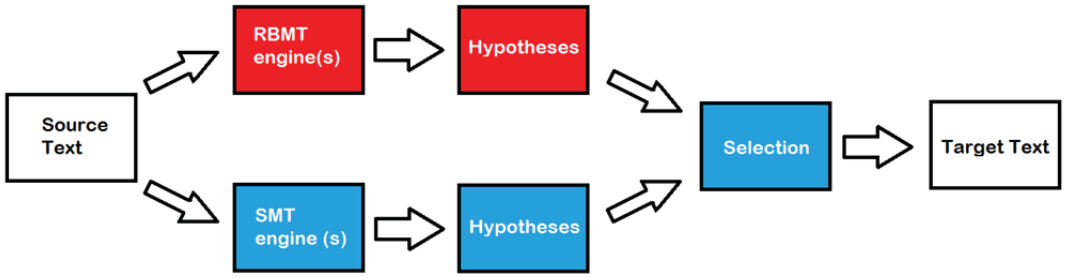


Figure 6-2: Stochastic Selection

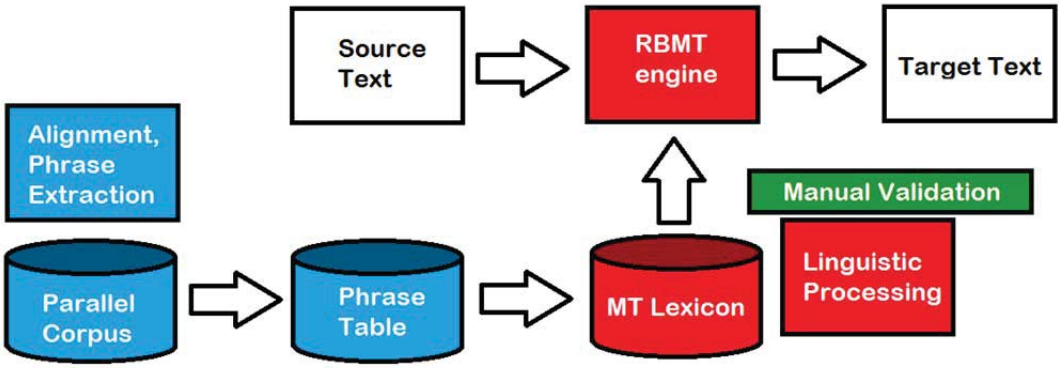


Figure 6-3: SMT-fed RBMT

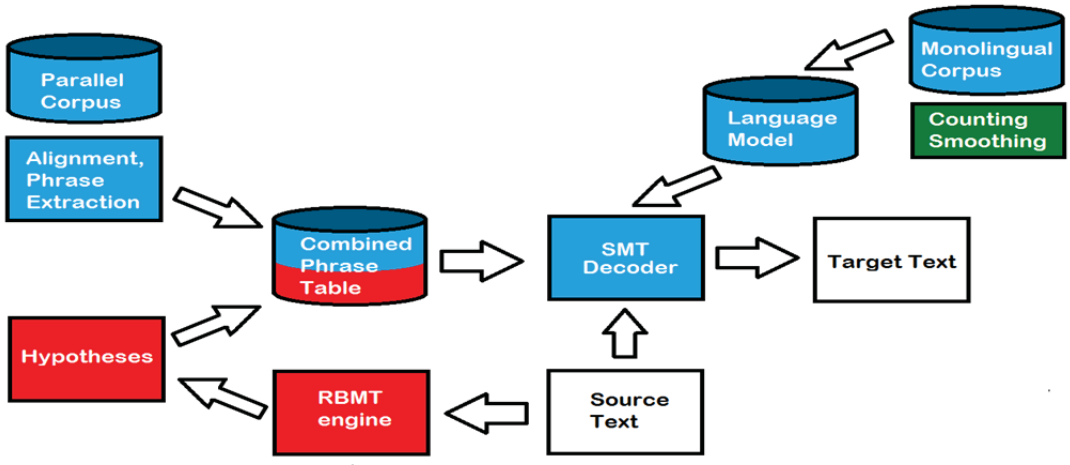


Figure 6-4: Hybrid Architecture

6.5 Proposed APE Approach

Our Transfer-based APE method consists of three levels of transformation: lexical transformer, shallow transformers and deep transformers. As shown in Figure 6.5,

first OOVRemover and Transliterator are run using a bilingual dictionary, after which some shallow transformers are run based on POS-tag patterns. Deep transformation at the third level is applied in which the rules exploit the tree dependency structure of sentences. In each step, information is transferred from some chunks to others, and, in some cases, certain chunks may disappear or even have information added to them.
The high-level diagram of this rule-based APE approach is demonstrated here.

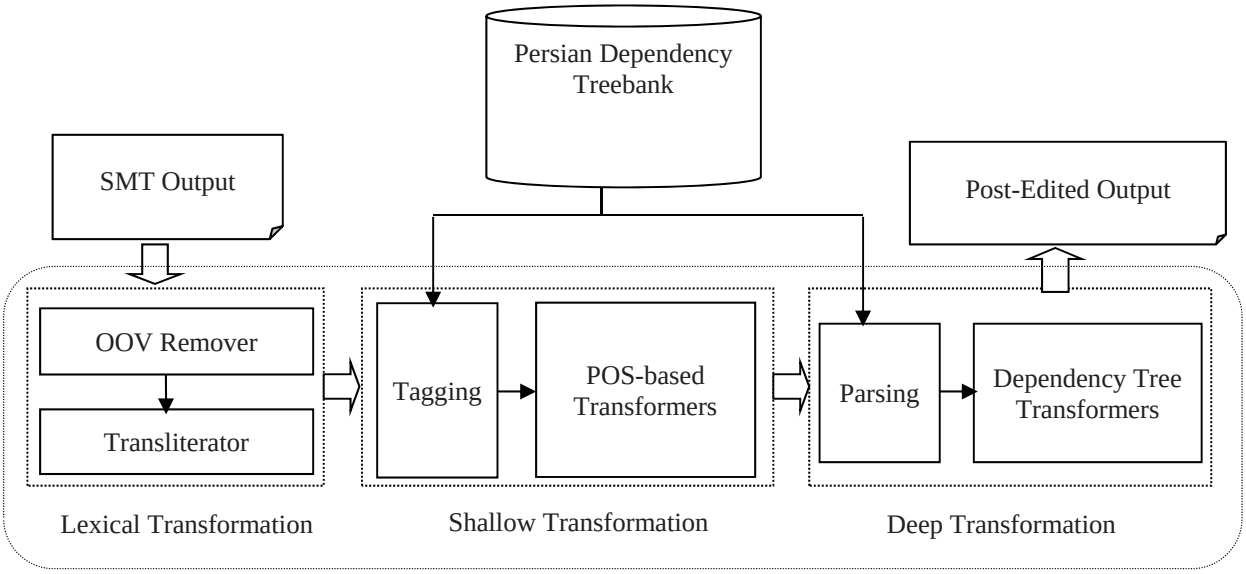


Figure 6-5: High-level Diagram of the Rule-based APE

The first stage of the method involves POS-tagging the input text. The MLE POS-tagger is used in the next stage and trained with the Persian Dependency Treebank. Once the text is tagged, some preparation is performed to parse the input. The Persian Dependency Treebank is also used in the training process of parser. MSTParser was used to parse input. This parser is an implementation of Dependence Parsing using the Maximum Spanning Tree algorithm in which the maximum spanning tree should be found in order to find the best parse tree (Kübler, McDonald *et al.*, 2009).

Lexical Transformation: The first level benefits from the outcome of two components. OOV[9]Remover is a simple substitute rule to replace an English word with the correct translation in Persian → E_1, E_2…E_n where E is an English word and F_i = F_1, F_2…F_n are different translations of that word in Persian. Since no WSD component is present, it is assumed that the first meaning found for that English word in the dictionary used is the most frequent translation of that word, so it is used as the replacement for the English word. Since the OOV words do not exist in the training

[9] Out Of Vocabulary

data for POS-tagging, and the program relies only on Persian linguistic knowledge, putting this component after the POS-tagger to use syntactic information and retrieve more suitable substitution is not possible. However, there are instances such as named entities where OOV Remover could not find equivalent Persian translations for English words appearing as OOV in the output. Transliterator is used to replace English words by their equivalents in Persian scripts. Transliterator works based on training an amount of prepared data to produce the most likely Persian word for the English word remaining in the sentence. The result is an English word appearing composed with Persian character scripts. In order to implement the transliterator component, several libraries from Virastyar[10] software were used.

Shallow Transformation: The second stage of the method involves a shallow transfer module. The transformers are developed based on some POS patterns identified as being wrong. The MLE POS-tagger is used in this stage and trained with the Persian Dependency Treebank data. Once the text is tagged, some preparation is performed to prepare input for the next level of transformation.

Deep Transformation: In the third level, the input is parsed by the dependency parser, MSTParser. Once parsed, the sentence structure is checked against the dependency tree to determine whether it demonstrates correct sentence structure.

6.6 Description of our APE Approach

The output of the SMT system is passed to the APE as input. The output of the system is in the same language as the input; the task of the APE is to fine-tune it in an effort to achieve a more accurate translation.

The first stage of our approach involves POS-tagging the input text. The MLE POS-tagger is used in the next stage and trained with the Persian Dependency Treebank (covered in the next paragraph). Once the text is tagged, it must then be prepared for parsing. This preparation is based on the MSTParser (McDonald, Pereira *et al.*, 2005) input format. In doing so, a new structure for parsing was created. In this structure, each sentence in the input text is represented by 3 or 4 lines, and sentences are space separated.

[10] http://sourceforge.net/projects/virastyar/

The general format is:

$$W_1 \quad W_2 \quad ... \quad W_n$$
$$P_1 \quad P_2 \quad ... \quad P_n$$
$$L_1 \quad L_2 \quad ... \quad L_n$$
$$D_1 \quad D_2 \quad ... \quad D_n$$

W_1 ... W_n are the n words of the sentence (tab-removed)

P_1 ... P_n are the POS tags for each word

L_1 ... L_n are the labels of the incoming edge to each word

D_1 ... D_n are integers representing the position of each word's parent

The MSTParser was used on both the reference and output text. The result from the parser is shown in Figures 6-6 and 6-7.

هنر	است	بوده	توجه	مورد	قبل	مدتها	از	که	است	پدیده ای	زیبایی	.	
N	N	V	SUBR	P	N	N	N	N	AJ	V	N	PUNC	
SBJ	NVE	ROOT	VCL	ADV	POSDEP	MOZ	MOZ	PREDEP	NPOSTMOD		PRD	VCL	PUNC
3	11	0	3	11	5	6	7	11	9	4	11	3	

principle_OOV	است	ذاتی	پدیده	در	زیبایی	که	دهد	نشان	.			
N	N	V	SUBR	N	P	N	AJ	V	PUNC			
SBJ	NVE	MOZ	VCL	PRD	NPP	POSDEP	NPOSTMOD		ROOT	PUNC		
9	3	1	3	4	5	6	7	0	3			

but_OOV	می کند	زیبایی	حس	دیگر	اي	مردم	نگرش	نوع	از	،	هر	در	.
N	P	QUA	PUNC	P	N	N	N	OH	AJ	N	N	V	PUNC
SBJ	ADV	POSDEP	POSDEP	ADV	POSDEP	MOZ	MOZ	NCL	MOZ	NEZ	NVE	ROOT	PUNC
13	13	2	2	13	5	6	7	8	6	10	13	0	13

Figure 6-6: Output text parsed with MSTParser

است	بوده	توجه	مورد	قبل	مدتها	از	که	است	ای	پدیده	زیبایی	.			
N	N	OH	V	SUBR	P	N	N	N	N	AJ	V	PUNC			
SBJ	MOZ	MOS	ROOT	VCL	ADV	POSDEP	MOZ	MOZ	NVE	NPOSTMOD		PRD	PUNC		.
12	1	4	0	4	12	6	7	8	12	10	5	4			

هر	در	است.اما	ذاتی	پدیده	یک	زیبایی	که	دهد	می	نشان	اصلی	هنر	.			
PUNC	N	N	N	PUNC	N	AJ	P	N	P	N	N	V	PUNC			
PRD	POSDEP	PUNC	SBJ	MOZ	MOZ	PUNC	SBJ	NPOSTMOD		NPP	POSDEP	NPP	POSDEP	NVE	ROOT	PUNC
14	27	16	17	18	27	20	21	22	23	24	27	0	5			

دهد	می	نشان	را	زیبایی	از	متفاوتی	درک	،	انسانها	نگرش	نوع	،	زمان
N	AJ	N	N	V	SUBR	N	N	N	AJ	N	P	QUA	N
SBJ	NPOSTMOD		MOZ	NVE	ROOT	AJUCL	SBJ	MOZ	MOZ	NPOSTMOD		MOZ	ADV
27	1	1	5	0	27	27	7	8	9	9	27	6	12

Figure 6-7: Reference text parsed with MSTParser

The Persian Dependency Treebank is also used in the training process of parser. The training set was prepared based on four fields of this Treebank to produce the training data format. It should be noted here that the format for training and input data are the same in MSTParser. MSTParser is an implementation of Dependence Parsing using the Maximum Spanning Tree algorithm in which the maximum spanning tree should be found in order to find the best parse tree (Kübler, McDonald *et al.*, 2009).

A number of different methods of syntactic annotation have been proposed over the history of NLP and information retrieval. Approaches based on phrase structure, dependency structure, and specific linguistic theories have been developed or proposed, while others use a theory-neutral approach (Nivre & McDonald, 2008). Recently, syntactic parsing based on dependency structure has become more attractive in NLP, specifically for languages having flexible word order. The most important and desired result from this technique is automatic learning.

6.7 Persian Dependency Treebank

Persian Dependency Treebank is the first Persian Treebank available free of charge (for non-commercial use). It is developed by Dadegan research group and a pre-version is available on their website[11]. The data format is based on CoNLL Shared Task on Dependency Parsing (Buchholz & Marsi, 2006).The sentences are manually annotated in the corpus, which contains about 12,500 sentences and 189,000 tokens.

6.8 Corpus Study for POS-Tagging Experiments

6.8.1 Related Work

F Oroumchian, S Tasharofi *et al.* (2006) document the development of a corpus suitable for Persian POS-tagging. This corpus is based on part of the Bijankhan tagged Persian corpus (Bijankhan, 2004), with over 2 million words in the training part and about 400,000 in test data. The content for this corpus is gathered from daily news and common text. Each document is assigned a subject (political, etc.) and there are 4300 subjects. The way the subjects are categorised provides a great experimental environment for clustering, filtering and categorisation research. Originally, the corpus had 550 tags. For the purpose of their research, F Oroumchian, S Tasharofi *et*

[11] http://dadegan.ir/en

al. (2006) ignored the subject categories of the documents, and only concentrated on the POS-tags. Since the large number of tags caused great difficulty with automatic machine learning, they reduced the number of tags to 40. In order to do this, the frequency of appearance of each tag was collected, and many similar tags were grouped together under one tag. They were then ordered in a hierarchical structure.

For the purpose of this thesis, it was not considered necessary to go through the process of reducing the number of tags. More details about this procedure can be found in (F Oroumchian, S Tasharofi *et al.*, 2006).

Table 6-1 below shows the tag distribution. Note in the table the most and least frequent tags. "N_SING" (Noun-Singular) is the most common tag, occurring 826,571 times, while "NN" (Number) is the least common, appearing only twice in the training set, and never in the test set.

There are numbers of different tagging models differing from the amount of processing information and amount of training necessary, right through to differences in the actual internal model. Most available taggers are designed to work with English text, and higher-resource languages. Low-resource languages and those with language characteristics differing greatly from English (such as Persian) are not so common.

The Maximum Likelihood Estimation (MLE) approach was chosen as the post-tagger component for the APE system, due to its ability to be implemented easily. Another factor influencing this choice was the success documented by F Oroumchian, S Tasharofi *et al.* (2006). Tagging tests showed the best accuracy achieved to be 95.43%. These sorts of success figures are more often seen in tagging for other high-resource languages such as English, German and Spanish.

Table 6-1: Tag Names

Tag Name	Frequency in Training Set	Percentage in Training Set	Frequency in Test Set	Percentage in Test Set
ADJ	21	0.001	1	0.000
ADJ_CMPR	5968	0.270	1475	0.377
ADJ_INO	22503	1.020	4693	1.199
ADJ_ORD	5743	0.260	849	0.217
ADJ_SIM	192171	8.709	38980	9.961
ADJ_SUP	6342	0.287	1001	0.256
ADV	1291	0.059	224	0.057
ADV_EXM	2398	0.109	793	0.203
ADV_I	1917	0.087	177	0.045

ADV_NEGG	1495	0.068	173	0.044
ADV_NI	18635	0.845	3265	0.834
ADV_TIME	7564	0.343	863	0.221
AR	2318	0.105	1175	0.300
CON	177769	8.056	32523	8.311
DEFAULT	48	0.002	32	0.008
DELM	217533	9.858	39062	9.982
DET	39783	1.803	6115	1.563
IF	2575	0.117	547	0.140
INT	111	0.005	2	0.001
MORP	2823	0.128	204	0.052
MQUA	250	0.011	111	0.028
MS	8	0.000	253	0.065
N_PL	135474	6.139	24945	6.375
N_SING	826571	37.459	140975	36.026
NN	2	0.000	0	0.000
NP	42	0.002	10	0.003
OH	271	0.012	12	0.003
OHH	15	0.001	5	0.001
P	270894	12.276	48964	12.513
PP	755	0.034	125	0.032
PRO	53792	2.438	8067	2.062
PS	296	0.013	37	0.009
QUA	12745	0.578	2673	0.683
SPEC	3281	0.149	528	0.135
V_AUX	13484	0.611	2386	0.610
V_IMP	1044	0.047	113	0.029
V_PA	73591	3.335	7003	1.790
V_PRE	35286	1.599	7209	1.842
V_PRS	41226	1.868	10512	2.686
V_SUB	28592	1.296	5228	1.336

6.8.2 POS-tagging for Persian language – difficulties

The Persian language possesses certain characteristics that can present difficulties when involved in POS-tagging. Basic Persian sentence structure follows a *subject–object–verb* pattern, differing from English's *subject–verb–object* structure. In Persian, objects are usually identified by the suffix "*-ra*". There are fewer verb tenses in Persian than in English, and the tense of a sentence is usually determined by the context. Persian also demonstrates inflectional and derivational morphology. Persian grammar is unaffected by gender. Word stems are either prefixed or suffixed in order to obtain derivational words (as in English).

A number of different problems are encountered by a POS-tagger in Persian. For example, if a word is in plural form, indefinite, and is a possessive pronoun, all these affixes attempt to connect to each other. In Persian, the person state of a verb can define the inflection of the verb for a number of different verb categories. As a result, "new" words can be formed. Blank spaces in Persian text can also present difficulties, such as the suffix "*-ha*", indicating plurality, can appear immediately suffixed to the host word, or with a space, as a separate entity. These factors can all result in both complicated and/or unknown outcomes in the tagging output (Mohseni, Motalebi *et al.*, 2008).

6.8.2.1 POS-Tagging – General overview and Persian-specific

POS (Part-Of-Speech) tagging, which can be defined as lexically tagging the words and symbols which make up a sentence or phrase, is indispensable to a number of areas of NLP. A number of different approaches to this task have been developed (such as rule-based and statistical-based approaches). Although a number of methods have been used extensively for a variety of different languages, this is not the case with Persian. There are several instances of work or research in tagging for Persian, for example, a system by Assi and Abdolhosseini (2000), based on (Schütze, 1995) proposed method and used in the annotation of FLDB corpus, and research by Megerdoomian (2004) documenting challenges and difficulties accompanying the development of a Persian-specific POS-tagger. However, there is still a significant lack of work in this area.

6.9 Parsing Approaches

There are several parsers available for Persian. One such open-source parser is the Persian LG Syntax Parser, by Dehdari and Lonsdale (2008), which is based on the Link Grammar (LG) system.

6.9.1 Link Grammar Parser

The Link Grammar parser system is a syntactic parser for the English language, operating with link grammar, which is an original theory of syntax for English. It is written in C code, and has an API, making embedding to other applications possible.
The system structures an input sentence syntactically, connecting pairs of words with labelled links, and also outputs the sentence showing parts of speech. Having a

dictionary with 60,000 word forms enables the parser to cover a large number of different syntactic arrangements. When it comes to dealing with unknown words or parts of a sentence, the parser performs exceptionally well, being able to skip over unknown sentence phrases while structuring the rest of the sentence. Unknown words are usually guessed based on sentence context, word placement and word ending.

Initially, John Dehdari was contacted in order to find out about using the LG Syntax Parser for the APE. He advised that there is no current development or maintenance for this parser, and suggested that a data-driven dependency parser be used instead. Based on his advice, MSTParser was used, as well as the Persian Dependency Treebank from Dadegan research group[12] (email\personal communication, 2012).

6.9.2 Data-Driven Dependency Parsing

Recently, there has been increased interest in dependency parsing for a number of different applications including machine translation (Ding & Palmer, 2005). Probably the biggest factor leading to its popularity is the efficiency with which dependency parsers can implement machine learning, yet still be able to encode a significant amount of predicate-argument data which is required by many applications.

In dependency parsing, words are linked to their arguments by dependency representations (Hudson, 1984). These representations have been in use for many years. An example is shown in Figure 6-8. The sentence, shown in sentence tree form, is a dependency tree. Each word depends on a "parent" word or a root symbol. This dependency tree is projective, which means that each parent word and its preceding words form a contiguous substring of the sentence in the tree.There are two data-driven dependency parsers that can be used for the Persian language: MaltParser, and MSTParser.

Label	PUNC	ROOT	OBJ	PREDEP	NPREMOD	SBJ
Token	.	می خوانم	را	نامه	یک	من
Pronunciation	.	/mi:khãnam/	/rã/	/nãmæ/	/yek/	/man/
POS	PUNC	V	POSTP	N	PRENUM	PR
English Equivalent	.	read		letter	a	I

Figure 6-8: Dependency parsing example

[12] http://dadegan.ir/en

6.9.3 MaltParser

MaltParser, developed in Sweden at Växjö and Uppsala Universities by Johan Hall, Jens Nilsson and Joakim Nivre, is a data-driven dependency parser-generator that can both parse new data (from an induced model), and induce a parsing model itself, with treebank data as input[13]. MaltParser's operation is different from a traditional parser-generator, which forms parsers from grammar. It employs inductive dependency parsing, where a dependency structure is derived from the sentence, and machine learning aids in parsing from non-deterministic points (Nivre, Hall *et al.*, 2006) .

6.10 Initial Steps for an RBMT-APE Approach

The purpose of this approach is to perform the process of post-editing for the output of the SMT machine. There are several components in this approach.

6.10.1 MLETagger

This component is an implementation of maximum likelihood estimation (MLE). Evaluation showed that the use of this approach for tagging the Persian language yielded promising results (Raja, Amiri *et al.*, 2007). There are several classes included in this component. The function of each class is explained in detail below.

6.10.2 Tagger class

This class, which is called Tagger, is used to tag the input text. *Tag()* is the only method for this class, and has three main parameters essential to the running of MLETagger:

Train set: this parameter is used to define the name and address of the training file for the tagger. The format for the training file is such that each line contains one word and its tag, separated by a tab.

Test set: this parameter is used to define the name and address for the input file containing tokens which require tagging.

Result: this parameter is used to save the name and address for the tagged file.

[13] http://www.maltparser.org/intro.html

While running this method, first the tagger will be trained based on the data in the train set file. Next, the test data will begin to be tagged.

6.10.3 MLETagger class

There are two parts in this class: training tagger and tagging process.

6.10.3.1 Training tagger:

Method *learning()* is used to train the tagger. In this method, the training data file is loaded and read line by line. In each line tokens and their parts of speech are detected, and, based on that information, a key content of a combination of that token and its part of speech can be made, together with the number of times that token is repeated with the particular part of speech in the training file. This information will be kept in a new collection with the name of *htNewStat*. In the next stage, another process will modify *htNewStat* in order to find the variety of parts of speech which that token is linked to, and the number of times each part of speech occurs in the training set. The new information is maintained in a collection called *ht*. For example, if the word "باز" had the part-of-speech classification of *noun,* occurring 20 times, but also had part-of-speech classification of *adjective,* which occurred 15 times, *ht* would have an entry as shown below:

باز N^20 AJ^15

Then, from this collection, the part-of-speech with the highest repetition would be chosen, and would be considered to have the greatest maximum likelihood probability for that token, to be used in the tagging process. In the previous example, part-of-speech *noun* which had a repetition of 20 (compared to *adjective* which had 15) would be chosen, and added to the *MLiklihood* set.

6.10.3.2 Tagging process:

The tagging process will run with the method *tagging()* in MLETagger class. In this method the input file (normal file, including words,

punctuation, numbers etc.) is loaded, and then tokenised. After this, each token is examined for its existence in *MLiklihood* collection. If it is available, the tag equal to that word will be linked to the word. If not, the part-of-speech *noun* will be considered as the default part of speech for that token.

In order to account for the difference in Unicode for some characters in Persian and Arabic (such as "ی" and "ک"), words with these characters will be considered with both Persian and Arabic Unicode.

Next, the probability of the tag for that word will be evaluated in the training set. This difference is due to the fact that the source of training data for tagger and source of input data have been generated in different machines.

6.10.4 CoNLL class

In order to generate the training data set for tagger from the Persian Dependency Treebank, the code in this file is used, and the method *PrepareTrainData*. In the Persian Dependency Treebank in CoNLL-2005, in each line there are several fields for each token, two of which are *token*, and *token part of speech*. The reason Treebank was used as a training set for the tagger was because of the compatibility the generated tagSet has with the tagger training set.

6.11 POS-Tagger

In order for our RBMT-based APE algorithm to work with our SMT system, it was necessary to parse both the output and also the reference text in order to extract rules to map the reference and output together, and improve the quality of the output by performing revision tasks, such as replacing OOV words with their equivalents in the target language, correcting grammar, and modifying the word order.

To accomplish this, the Persian output must be parsed. The first stage of parsing is POS-tagging, or annotating each word for its part of speech (grammatical type) in a given sentence. Examples of POS-tagging Persian output are shown in Table 6-2:

Table 6-2: Examples of pos-tagging Persian output

زیبایی	N_SING
پدیده	N_SING
ای	OH
است	V_PRE
که	CON
از	P
مدتها	N_PL
قبل	N_SING
مورد	N_SING
توجه	N_SING
بوده	ADJ_INO
است	V_PRE
.	DELM

Generally, POS-tagging helps with parsing, and resolves pronunciation and semantic ambiguities.

POS-tagging is a useful task for many applications such as word sense disambiguation**,** parsing, and language modelling. Tagging techniques can also be used for a variety of tasks such as semantic tagging, dialogue tagging and information retrieval.

Not all pos-taggers follow the same standard for tagging. Some use coarse classes, such as N, V, A, Aux… (Amiri, Raja *et al.*, 2007).

Some other taggers, such as Penn Treebank, prefer finer distinctions:

- PRP: personal pronouns (you, me, she, he, them, him, …)
- PRPS: possessive pronouns (my, our, her, his, …)
- NN: singular common nouns (leg, plate, calculator, …)
- NNS: plural common nouns (legs, plates, calculators, …)
- NNP: singular proper names (Microsoft, Europe, London, …)
- NNPS: plural proper names (Americas, Carolinas, …)

Data is tagged for POS in the same way that humans tag a corpus. A POS-tagger attempts to model human performance by matching their performance. To build the model, corpora are hand-tagged for POS by more than one annotator before being checked for reliability. The corpus used for the tagger in this research is the Bijankhan corpus.

The Bijankhan corpus is a collection of articles from daily news and common texts. The articles and documents are categorized, divided into different domains and subjects (literature, politics, culture, science etc.), that is, about 4300 separate subjects in total. The corpus itself is a tagged corpus, containing about 2.6 million manually-tagged words. They are tagged with a tag set containing 40 Persian POS tags. It is used by researchers in natural language processing, and is distributed by a database research group at the University of Tehran[14].

As shown in Figure 6-9, there is a number of different approaches to POS tagging:

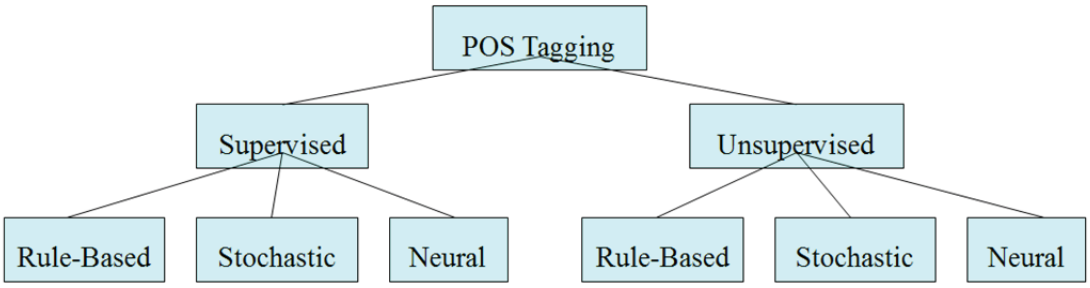


Figure 6-9: POS-Tagging Approaches

For this APE, MLE parser was used, which is stochastic. Automatic training is made possible with the use of a probabilistic POS-tagger. In this way, rule revision, which is tedious and takes time, can be avoided. Automatic training also makes adaptation to new text domains possible.

The chosen approach to stochastic parsing was Maximum Likelihood Estimation (MLE). MLE calculates the maximum likelihood probability for each tag assigned to the words in the training set. In the second stage, for each word, the tag with the greatest maximum likelihood probability will be set specifically for that word alone. In the evaluation stage, the test set words are analysed, and those tags that were set specifically are assigned to those same words in the test set.

MLE parser can provide accurate parsing when it is trained on a large corpus. Unigram statistics (the most common part of speech for each word) can achieve up to 90% accuracy. Further accuracy is achievable with more information on adjacent words.

[14] http://www.ut.ac.ir/en

In a statistical model, the probability can be extracted from the tagger corpus which the MLE tagger has trained on it. Also, a corpus embedded too deeply in a particular domain may not be transferrable or usable by other domains, yet, if it is too generic, it may be unable to benefit from domain-specific probabilities.

A tagging model can be tested, typically, by splitting the corpus into the training set and the test set. The test set should be held out from the training set. The tagger can learn the tag sequences that can maximize the probability for that model. Finally, the tagger can be tested on the test set. Although the tagger should not be trained on the test data (as an unreliable result would be generated), it is possible to have test data very similar to training data.

The MLE tagger is run on both output and reference texts from the SMT system. Details of this can be found in Appendix III, section 4. The results are as follows:

Output text:

زیبایی	N_SING
پدیده ای	N_SING
است	V_PRE
که	CON
از	P
مدتها	N_PL
قبل	N_SING
مورد	N_SING
توجه	N_SING
بوده	ADJ_INO
است	V_PRE
هنر	N_SING
.principle_OOV	N_SING
نشان	N_SING
دهد	V_SUB
که	CON
زیبایی	N_SING
در	P
پدیده	N_SING
ذاتی	ADJ_SIM
است	V_PRE

Reference text:

زیبایی	N_SING
پدیده	N_SING
ای	OH
است	V_PRE
که	CON
از	P
مدتها	N_PL
قبل	N_SING
مورد	N_SING
توجه	N_SING
بوده	ADJ_INO

```
است             V_PRE
.               DELM
هنر             N_SING
اصلی            ADJ_SIM
نشان            N_SING
می              N_SING
دهد             V_SUB
که              CON
زیبایی          N_SING
یک              N_SING
پدیده           N_SING
ذاتی            ADJ_SIM
است.            N_SING
```

6.12 Summary

In summary, this chapter shows the motivation behind the development of an automatic post-editing approach, and gives an overview of related work in automatic post-editing approaches, showing the different architecture of various hybrid systems. In particular it is shown that the method of a Rule-based automatic post-editing approach has not been explored extensively, specifically with respect to correction of an SMT system's output. The chapter also shows the preparation necessary for a Rule-based APE approach, such as POS-tagging and parsing, and shows the particular POS-tagging and parsing approaches used for this system.

Chapter 7. APE Method Development, Experiments and Results

"Those who know nothing of foreign languages know nothing of their own."~ Johann Wolfgang von Goethe

7.1 Introduction

This chapter firstly explains the implementation of our proposed approach, giving details of the development of our three-level automatic post-editing method. We show application examples of each transformer, and how they improve sentence structure with examples before and after transformation. The second part of the chapter shows the output of our SMT system before and after APE, and is evaluated using both automatic and manual methods.

7.2 Program class

This class is the starting point for our APE approach. In the *main()* method of this class, there are several parameters which are adjustable as inputs for the program. The main inputs are, first, the name for input file (inputFile), the name for the tagger train set (taggerTrainFile), and the name for the parser train set (parserTrainFile). The assumption for this program's operation is based on all the inputs and outputs being in the same directory. This path should be introduced through the DataFolderPath parameter. For running or executing this program, first, one instance of this parser will be created, and then all the necessary settings such as defining the paths for data files and the name for the tagger train file should take place. The stage for training the parser begins by calling the method *train( ).* When the program runs for the first time, the training step is necessary. After training, a file with the name of *model.dep* will be created in the same directory as the other files. Since the training process is extremely time-consuming, for the next runs of the program the information of *model.dep* can be used, and there is no need for retraining the parser. In order to prevent the parser from retraining the next runs, it is enough to change the value of switch t=0 in the command line of the program. The default value for this switch is 1. The next stage involves parsing the input file. The output of the parsing process will be saved with the same filename as the input, but with suffix *-parsed*, again in the same directory as the other data.

7.2.1 ParserDataLine class

The input formats for both the training and the parsing are the same in each case. In this format, every sentence is located in one line, and every token of that sentence is tab-separated. Three subsequent lines are assigned to each sentence. The first line shows the part of speech for each token (tab-separated), the second line shows the label of incoming edge for each node in the tree. The third line contains integers representing the position of each word's parent, again tab-separated. In order to save the data in each line, the ParserDataLine structure is created, which models these four lines with four lists, including tokens, POSes, labels and parent nodes. Information for each line is loaded to these lists.

7.2.2 DataPreparation class

In this class, two methods are implemented in order to prepare data for both training and running the parser. The first method *prepareFromTreebank( )* is used for preparation of training data. Training data is extracted from the Persian Dependency Treebank. In order to accomplish this, a list of ParserDataLine objects will be created, and then all the sentences in the Treebank with all the required annotation information such as tokens, POSes, labels and parent nodes are fitted into this structure. An example from a sentence extracted from the parser structure of Treebank is shown below in Table 7-1. Note that the integer value assigned to the position of the parent node for the verb of the sentence is zero.

Table 7-1: DataPreparation class

به	همین	دلیل	با	"	ریسمان	باز	"	تفاوت	دارد	.
PREP	PREM	N	PREP	PUNC	N	ADJ	PUNC	N	V	PUNC
ADV	NPREMOD	POSDEP	NPP	PUNC	POSDEP	NPOSTMOD	PUNC	OBJ	ROOT	PUNC
10	3	1	9	6	4	6	6	10	0	10

The second method *prepareFromTagger( )* converts the output of the tagger to the appropriate format for the parser. In this method, again, a list of ParserDataLine structure will be created, and each tagged sentence will be fitted in this structure. At the time the output is being produced based on the parser format from the input data, there are only two lines present: tokens and POSes. Since the other two lines (labels

and parent nodes) have no data, these lines will be filled with artificial data. For example, for all tokens in line 3, LAB (label) is entered. In line 4, for all parent nodes, zeros are entered. The resulting output file has the same name as input file but with the suffix *_prepared.*

7.3 MSTParser Details

7.3.1 Class Parser

This class trains the parser with the train set data and parses input data. In order to train the parser and run the parsing procedure, the converted code of MSTParser was used in C#, called MSTParserCSharp. In this class, after training the parser, the model file *model.dep* will be created, which includes the trained model. The input will be parsed based on this model.

7.3.2 Training Parser

Method *Train()* in this class will receive the name of the training file through the trainFile parameter, and call the method *Train()* of MSTParser class. The inputs of this method are as follows:

Function: public Static Void Train(string trainFile, string modelName, intnumOfTrainingIterations, bool isProjective, inttrainingK, bool createForest, int order).

This function will train a parser with all the default properties. Additional properties can be described with the following flags:

Train – if present, parser will train a new model

modelName – stores trained model in a file named *model.dep*

numOfTrainingIterations – runs training algorithm for numIters epochs (default is 10)

isProjective – type is either "proj"=true or "non-proj"=false (e.g., decode-type:proj). The default is "proj". "proj" uses the projective parsing algorithm during training (i.e., the Eisner algorithm), while “non-proj” uses the non-projective parsing algorithm during training (i.e., the Chu-Liu-Edmonds algorithm).

training – specifies the k-best parse set size to create constraints during training (default is 1). For non-projective parsing algorithm, k-best decoding is approximate.

createForest – cf. is either “true” or “false” (default is “true”). If *createForest* is “false” it will not create the training parse forest, instead it assumes it has already been created. This flag is useful if you are training many models on the same data and features but using different parameters (e.g., training iters, decoding type).

order – word is either 1 or 2, with the default set as 1. This flag specifies the order/scope of features. 1 only has features over single edges, while 2 has features over pairs of adjacent edges in the tree.

The following flags are set:

MSTParser.Train(trainFilePath, modelPath, 20, false, 1, true, 2);

7.3.3 Parsing inputs

Method *parse()* is used to parse the input file. This method tags the input, after receiving the input file path by running the MLEtagger component. In order to train the data for tagging, first the path for the training file should be defined for MLETagger. The output of this step will be saved in the same directory as other files with same name as input, only suffixed with *-tagged*. In the next step, the tagged input should be converted to a compatible format for MSTparser. To achieve this, the method *PrepareFromTagger()* from class DataPreparation must be applied to the tagged input.

The next step is the parsing, which uses method *Test()* from MSTParser class. In order to run this method, the path of the new formatted data (in compatible format for parser) and the name and path for the output file should be defined. After parsing, the output file will be in the same directory as the input file, and will be saved in the same name as input, suffixed with *“_parsed”*.

Table 7-2 below gives an example of parsed data from MSTParser for an input sentence:

Table 7-2: Parsing Inputs

زیبایی	پدیده	ای	است	که	از	مدتها	قبل	مورد	توجه	بوده	است	.
N	N	ADR	V	SUBR	PREP	N	PREP	N	N	V	V	PUNC
SBJ	MOZ	MOS	ROOT	AJUCL	ADV	POSDEP	NPP	POSDEP	MOZ	PRD	PRD	PUNC
4	1	4	0	4	12	6	7	8	9	5	5	4

Tagger will define POS “N” - NOUN for those words in the input that have no POS in the training data set. For those input words which the SMT system has not been able to translate to Farsi because of being out of vocabulary - shown as _OOV in the input - the parser will use the same POS “N”. Table 7-3 below shows a list of tags used.

Table 7-3: POS-Tagger: Parts of speech categorised

ACL	Complement Clause of Adjective
ADV	Adverb
ADVC	Adverbial Complement of Verb
AJCONJ	Conjunction of Adjective
AJPP	Prepositional Complement of Adjective
AJUCL	Adjunct Clause
APOSTMOD	Adjective Post-Modifier
APP	Apposition
APREMOD	Adjective Pre-Modifier
AVCONJ	Conjunction of Adverb
COMPPP	Comparative Preposition
ENC	Enclitic Non-Verbal Element
LVP	Light Verb Particle
MESU	Measure
MOS	Mosnad
MOZ	Ezafe Dependent
NADV	Adverb of Noun
NCL	Clause of Noun
NCONJ	Conjunction of Noun
NE	Non-Verbal Element of Infinitive
NEZ	Ezafe Complement of Adjective
NPOSTMOD	Post-Modifier of Noun
NPP	Preposition of Noun
NPREMOD	Pre-Modifier of Noun
NPRT	Particle of Infinitive
NVE	Non-Verbal Element
ODJ	Object
ODJ2	Second Object
PARCL	Participle Clause
PART	Interrogative Particle
PCONJ	Conjunction of Preposition
POSDEP	Post-Dependent
PRO	Predicate
PREDEP	Pre-Dependent
PROG	Progressive Auxiliary

PUNC	Punctuation Mark
ROOT	Root
SBJ	Subject
TAM	Tamiz
VCL	Complement Clause of Verb
VCONJ	Conjunction of Verb
VPP	Prepositional Complement of Verb
VPRT	Verb Particle

7.4 Transformers

In order to improve the accuracy of the SMT system, the transformer component in our APE approach performs necessary changes to SMT output. By investigation of incorrect and incomplete translation outputs and considering the dependency parser output for these sentences, a number of incorrect sequences were identified in the POS sequence and dependency parse tree of these sentences. The incorrect patterns are compared against the Persian Dependency Treebank to ensure there is no such pattern used in normal sentences. If it appears that the sequence or pattern is unknown, it is deemed incorrect, and rules are defined (modified or corrected) and implemented as transformer classes. The transformers are run on each sentence of the input for the APE system, and correction is made where incorrect POS sequence or tree structure pattern is detected.

All these transformers will be run on each sentence and in the case of detection of an incorrect pattern they will be transferred to the correct equal sentence.

Transformers are divided into two main groups, depending on their run times: those which are run on input sentences prior to the parsing process, and those which run on parsed inputs, after the parsing process.

In the first group of transformers, incorrect or incomplete POS sequence patterns will be controlled for each sentence, and appropriate rules executed to revise them. In the second group of transformers, the tree structure of each parsed sentence will be analysed, and those sentences with incorrect parsed tree structure will be evaluated to determine whether correction is necessary.

In the following section, different classes in this component, together with different transformers and their function, will be explained in more detail, and examples of operation will be shown.

7.4.1 OOV Remover class

The first transformer implemented in our APE approach removes the English words in the SMT output which remain untranslated ("out of vocabulary", or OOV), the SMT system being unable to find the equivalent Persian translation for them. This transformer is implemented on text before the parsing process.

This class uses a simple substitute rule to replace an English word with the correct translation in Persian. $E \rightarrow F_1, F_2 \ldots F_n$ where E is an English word and $F_i = F_1, F2 \ldots F_n$ are different translations of that word in Persian. Since no WSD component is present, it is assumed that the first meaning found for the English word in the dictionary used is the most frequent translation of that word, so it is used as a replacement for the English word.

A condition created for this substitute rule is a limitation on the number of characters comprising the words that a replacement definition can have. For instance, some dictionary translations of single words in English can result in multiple-word phrases in Persian. If the first occurring translation for any English word exceeds the set character number threshold, the next translation of that word from the dictionary is used.

If no Persian translation for an English word is found, the word will remain in the text unchanged as English.

The output of this transformer is the same as the input text, but (where successful) with OOV words replaced with their Persian equivalents. The most common examples of English words remaining unchanged were proper nouns, including people and place names.

7.4.2 Dictionary class

The previous class (OOVRemover) used the dictionary class in order to find the correct translation for English words in the text. In this class, method *LoadDictionary()*, loads the data files of the English-Persian dictionary. The

dictionary used in this transformer includes 65,000 entries for English words and their corresponding meanings. Method *GetMeaning (string word)()*gets back each word's meaning in Persian. For plural words, the singular meaning of that word is searched, and postfix "ها" is added, (which is the most common plural postfix in Persian), in order to construct the plural Persian word of the particular plural English word.

7.4.3 TransferEngineclass

This class is developed to execute all transformers or grammar rules to every sentence in the input. There are two methods in this class, *TransformBeforeParse(string file)()* and *TransformAfterParse(string file)()* which consecutively manage execution of the transformers. In each of these methods, all relative transformers are run on each sentence of the input text, and, where necessary, applicable rules in the transformers able to revise whole or part of the sentence will be implemented in order to accomplish accurate revision.

7.4.4 NumberPreserverclass

In order to process numbers in the input text, the parser replaces each number with the label <num>. Unlike other transformers, this transformer applies to texts both before and after the parsing process. Method *Preserver(ParserDataLine [])( )* preserves the numbers and their place in each sentence before and after parsing. Replacing label <num> with the correct number and in the correct place in the sentence is accomplished using the *ReplacePreservations(ParserDataLine[])* method.

7.4.5 IncompleteDependentTransformerclass

A relative clause, or adjectival clause, modifies a noun or noun phrase, and is introduced by a relative pronoun (*which, that, who, whom, whose*) or a relative adverb (*where, when, why).* In Persian, as in English, they are usually connected by relative pronouns such as "که". For example,

Persian:
« به این دلیل است که به او اعتماد کردم. »

English:
"This is the car that was used in the race".

In this case, the first part of the sentence - *"This is the car that"* – is not complete without the second part. The word *"that"* suggests continuation of the phrase. Both parts need to have at least one verb to complete the sentence. Therefore, while checking the POS sequence for each sentence, if it is observed that the sentence finishes immediately after the relative pronoun, and the POS sequence suggests that the sentence would not make sense, the case is revised.

The revision procedure that was developed operates in this way: In such a tagged sentence, there should be at least one verb tag V in between the subject tag SUBR and the period punctuation tag PUNC. If any instance is observed that does not follow this order, it follows that a verb should be added to form a correct sentence. Currently, in most instances, the verb «است» (*"is"* in English) is suggested.

In the notation below, * represents any number of POS, and ^ means 'except'. The following notation shows how an incorrect sentence (on the left) is corrected, and changed to the correct form (on the right).

[* SUBR *^V PUNC] → [* SUBR V PUNC]

The incorrect pattern is shown in the example below:

Table 7-4: IncompleteDependentTransformerclass – Before

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
تقاضا	می	افزودن	,	که	بیش	از	یک	میلیارد	انسان	در	سنین	بین	15	تا	24	سال‌ها	.
N	N	N	N	SUBR	ADJ	PREP	PRENUM	N	N	PREP	N	PREP	PRENUM	SUBR	PRENUM	N	PUNC

Table 7-5: IncompleteDependentTransformerclass – After

The modified sentence after running the transformer is shown below:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
تقاضا	می	افزودن	,	که	بیش	از	یک	میلیارد	انسان	در	سنین	بین	15	تا	24	سال‌ها	است	.
N	N	N	N	SUBR	ADJ	PREP	PRENUM	N	N	PREP	N	PREP	PRENUM	SUBR	PRENUM	N	V	PUNC

7.4.6 IncompleteEndedPREMTransformerclass

In some SMT outputs, the POS sequence shown below was observed:

[N PREP PREM PUNC]

where PREM is a pre-modifier. Pre-modifiers are a class of noun modifiers that precede nouns and are in complementary distribution with other members of the class. According to the definition for PREM, modifiers should precede nouns, so it can be seen that the POS sequence shown above is incorrect, having no example of it in the Persian Dependency Treebank. This sequence must be revised, to bring the preposition before the pre-modifier.

Table 7-6 below gives an example of an incorrect sequence:

Table 7-6: IncompleteEndedPREMTransformerclass- Before

1	2	3	4	5	6	7	8	9	10	11	12
اصل	هنر	را	نشان	می	دهد	که	زیبایی	یک	پدیده	ذاتی	است
N	N	POSTP	N	N	V	SUBR	N	PRENUM	N	ADJ	V

13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
پاگیر	در	هر	؛	به	نوع	نگرش	انسانها	درک	متفاوتی	از	زیبایی	را	نشان	می	دهد	.
N	PREP	PREM	PUNC	PREP	N	N	N	N	ADJ	PREP	N	POSTP	N	N	V	PUNC

Since there is no such translation for the given input, the following sequences were removed from the sentence.

[$*_a$ N PREP PREM PUNC $*_b$] → [$*_a$ $*_b$]

The modified sentence after running the transformer is shown below in Table 7-7:

Table 7-7: IncompleteEndedPREMTransformerclass- After

1	2	3	4	5	6	7	8	9	10	11	12	13
اصل	هنر	را	نشان	می	دهد	که	زیبایی	یک	پدیده	ذاتی	است	.
N	N	POSTP	N	N	V	SUBR	N	PRENUM	N	ADJ	V	PUNC

14	15	16	17	18	19	20	21	22	23	24	25	26
به	نوع	نگرش	انسانها	درک	متفاوتی	از	زیبایی	را	نشان	می	دهد	.
PREP	N	N	N	N	ADJ	PREP	N	POSTP	N	N	V	PUNC

7.4.7 AdjectiveArrangementTransformerclass

In the Persian language, adjectives usually come after the nouns they describe. For instance “heavy bag” «کیف سنگین», or “beautiful flower” «گل زیبا» are literally “bag heavy”, and “flower beautiful”. The only exceptions in this group are superlative adjectives (for example, “high*est* mountain” «بلندترین کوه»). In this case, the adjective comes before the noun to define it.

It can be concluded then that the appearance of non-superlative adjectives before their described nouns indicates incorrect composition, which needs revision if present in the output of the SMT system. The transformer described in this section is designed to check whether this is the case - if non-superlative adjectives appear before nouns, word reordering must be initiated.

This class detects all adjectives in the POS sequence of each sentence, and identifies adjectives not ending with «ترین» as non-superlative adjectives. Each sentence having non-superlative adjectives will have its parse tree checked.

If any NPREMOD tag is noticed (where NPREMOD is the tag for a defining word where it precedes the word that is defined), and the following word is POS-tagged as a noun, reordering moves the adjective to appear after the noun. The following example in Table 7-8 shows an instance of a sentence with incorrect combination (see columns 4 and 5).

Table 7-8: AdjectiveArrangementTransformerclass- Before

1	2	3	**4**	**5**	6	7	8	9	10	11
متاسفانه	امروزه	،	**گران**	**تزئینی**	مد	لباس	و	جراحی	زیبایی	مختلف
N	ADV	PUNC	**ADJ**	**N**	N	N	CONJ	N	N	ADJ
SBJ	ADV	PUNC	**NPREMOD**	**OBJ**	MOZ	MOZ	NCONJ	POSDEP	OBJ	NPOSTMOD
20	20	2	**5**	**20**	5	6	7	8	20	10

Table 7-9 shows the corrected version of the above sentence after running the transformer.

Table 7-9: AdjectiveArrangementTransformerclass- After

1	2	3	**4**	**5**	6	7	8	9	10	11
متاسفانه	امروزه	،	**تزئینی**	**گران**	مد	لباس	و	جراحی	زیبایی	مختلف
N	ADV	PUNC	**N**	**ADJ**	N	N	CONJ	N	N	ADJ
SBJ	ADV	PUNC	**OBJ**	**NPREMOD**	MOZ	MOZ	NCONJ	POSDEP	OBJ	NPOSTMOD
20	20	2	**20**	**5**	5	6	7	8	20	10

7.4.8 NoSubjectSentenceTransformerclass

SMT output occasionally contains instances of sentences with a third person verb, no definite subject and an object tagged as OBJ in the parse tree and tagged as POSTP in POS sequence. When tested with known reference sentences, it was seen that what was parsed as the object in the sentence was actually the subject. The transformer is designed to revise the sentence by removing the preposition «را» which is the indicator of an object in the sentence. Removal of this preposition changes the sentence to one without an object.

Table 7-10 below is an example of a sentence with this problem:

Table 7-10: NoSubjectSentenceTransformer class - Before

1	2	**3**	4	5	6
اصل	هنر	**را**	نشان	می	دهد
N	N	**POSTP**	N	N	V
PREDEP	MOZ	**OBJ**	NVE	MOZ	ROOT
3	1	**6**	6	4	0

The revised sentence is shown below in Table 7-11:

Table 7-11: NoSubjectSentenceTransformer class - After

1	2	3	4	5
اصل	هنر	نشان	می	دهد
N	N	N	N	V
PREDEP	MOZ	NVE	MOZ	ROOT
3	1	5	4	0

7.4.9 PluralNounsTransformer class

Another incorrect syntax pattern observed in SMT output is instances where plural nouns are located after a number (< PRENUM> POS).

Unlike English, in the Persian language the word coming after a number is always singular. (The only instance in English when the word following the number is singular is when the number defining the word is 1). For example, in the English phrase "two ways", "*ways*" is in plural form. However, in Persian, the form is "«دوراه»". PluralNounsTransformer checks the SMT output for such a pattern, correcting it if it is found. The following example in Table 7-12 shows this pattern, and the modified version in Table 7-13:

Table 7-12: PluralNounsTransformer class - Before

1	2	3	4	5	6	7	8	**9**	**10**	11
رابطه	هنر	و	پول	فهمیده	شود	در	حداقل	**دو**	**راه‌ها**	.
N	N	CONJ	N	N	V	PREP	ADV	**PRENUM**	**N**	PUNC
SBJ	MOZ	NCONJ	POSDEP	NVE	ROOT	ADV	APREMOD	**NPREMOD**	**POSDEP**	PUNC
6	1	2	3	6	0	6	9	**10**	**7**	6

Table 7-13: PluralNounsTransformer class - After

1	2	3	4	5	6	7	8	**9**	**10**	11
رابطه	هنر	و	پول	فهمیده	شود	در	حداقل	**دو**	**راه**	.
N	N	CONJ	N	N	V	PREP	ADV	**PRENUM**	**N**	PUNC
SBJ	MOZ	NCONJ	POSDEP	NVE	ROOT	ADV	APREMOD	**NPREMOD**	**POSDEP**	PUNC
6	1	2	3	6	0	6	9	**10**	**7**	6

7.4.10 VerbArrangementTransformerclass

Persian as a natural language has a word order preference, with SOV (subject-object-verb) being the most common type, followed by SVO. These two types make up more than 75% of natural languages which have a preferred order (Crystal, 2004). Although reordering of sentence components does not necessarily lead to a significant change in the meaning of the sentence, there are many cases where these changes may disturb the fluency and accuracy of the sentence. One example is compound verbs: in linguistics, a compound verb or complex predicate is a compound consisting of two or more words acting as a single verb. These words should appear at the end of the

sentence and be kept in the correct order to maintain fluency and, sometimes, the correct meaning of the sentence. For example, in the case of a compound with noun+verb, the noun is converted into a verbal structure; the arguments and semantics are determined by the noun, and the tense markers or inflections are carried by the verb and should both be located at the end of the sentence. By examining edge labels in the parse tree for the sentence, a compound verb with an NVE label can be seen in the parsed tree structure. This transformer is applied to sentences which have one main verb labelled *Root*, which does not occur immediately before the period punctuation. The matching procedure is as follows: for the verb of the sentence which is labelled as Root, the dependants are found. Then for those dependants whose edge label is NVE, reordering is performed by moving Root verb and its NVE dependants to the end of the sentence, just before the period punctuation. The verb dependants could be identified by their parent node index in the last line of ParserDataLine of the sentence. The following example in Table 7-14 shows the compound verb (N+V) appeared in the middle of the sentence. This is considered to be an incorrect pattern, and is modified by VerbArrangementTransformer.

Table 7-14: VerbArrangementTransformer class -Before

1	2	3	4	**5**	**6**	7	8	9	10	11
رابطه	هنر	و	پول	**فهمیده**	**شود**	در	حداقل	دو	راه	.
N	N	CONJ	N	**N**	**V**	PREP	ADV	PRENUM	N	PUNC
SBJ	MOZ	NCONJ	POSDEP	**NVE**	**ROOT**	ADV	APREMOD	NPREMOD	POSDEP	PUNC
6	1	2	3	**6**	**0**	6	9	10	7	6

The modified version is shown below in Table 7-15:

Table 7-15: VerbArrangementTransformer class -After

1	2	3	4	5	6	7	8	**9**	**10**	11
رابطه	هنر	و	پول	در	حداقل	دو	راه	**فهمیده**	**شود**	.
N	N	CONJ	N	PREP	ADV	PRENUM	N	**N**	**V**	PUNC
SBJ	MOZ	NCONJ	POSDEP	ADV	APREMOD	NPREMOD	POSDEP	**NVE**	**ROOT**	PUNC
6	1	2	3	6	9	10	7	**6**	**0**	6

7.4.11 Transliteratorclass

This class is used to replace English words by their equivalent in Persian script. This class is useful for instances where OOVRemover could not find equivalent Persian translations for English words appearing as OOV in the output. For example, some proper names in English have no equivalent in the Persian dictionary, and will stay untranslated in the output. The object is to reduce the number of source words appearing in the target translation. Having fewer words in the source language has a large effect in the metric evaluation results and can decrease the scores to some extent.

For such words, equal transliteration is proposed. The Transliterator transformer works based on training an amount of prepared data to produce the most likely Persian word for the English word remaining in the sentence. The result is the English word appearing composed in Persian character script.

In order to implement this class, the following two components were used: PinglishConverter and Transliteration library from Virastyar[15] Software.

There are two important methods which are defined and used in this class. The first is *Train*, which is used for training the transliterator. For this method, two input files are necessary: one with the name *TrainingFilePath*, which defines the path and address for the training data; the other, *PreprocessFilePath*, which defines the path for the pre-processing data files. Pre-processing data files contain almost 4000 of some of the most frequent Persian words written in English script, together with the correct writing of each of them in Persian script. The second method is *Transliterate(*stringword*)*, which suggests a word in Persian script for any English input word.

Some of the words which transfer correctly with this class to their equivalents in Persian are shown in Table 7-16 below:

[15] http://sourceforge.net/projects/virastyar/

Table 7-16: En-Fa Transliteration (1)

English	Farsi Transliteration
Mehdi	مهدی
Sepideh	سپیده
Amir	امیر
Kabir	کبیر
Khahar	خواهر
Alireza	علیرضا

In some of the other instances, the suggested word from the Transliterator is not a correct word. Table 7-17 below shows some of the words from this group:

Table 7-17: En-Fa Transliteration (2)

English	Transliteration Suggestion	Correct Word
Morteza	مرتزا	مرتضي
Esfahan	اصفهن	اصفهان

7.4.12 ConjunctedTokenTransfer class

Words are able to be connected to each other using co-ordinating conjunctions. Conjunctions are joining words used to connect words to words, phrases to phrases, or clauses to clauses. The connected words have the same POS. An example of this is shown in the phrase *"They used **time** and **energy**"*, where the conjunction is the underlined word *and*, and the words joined are *time* and *energy*, which are POS-tagged as nouns (N). Because of this, if any words connected with a conjunction do not have the same POS-tag, this is an indication of an error, and ConjunctedTokenTransfer class must correct it. The transformer works by identifying the POS-tags of the joined words. Where they differ, the transformer will preserve the POS of the first word, and change the second word so that its POS matches the POS of the first word. Table 7-18 below gives an example of this in which the noun form of the word "زیبا" (that is identified as an adjective) is built:

Table 7-18: ConjunctedTokenTransfer class - Before

1	2	3	4	5	6	7	8
منزلت	خاطر	حفظ	حجاب	**اسلامی**	**و**	**زیبا**	است
N	N	N	N	**N**	**CONJ**	**ADJ**	V

The sentence after operation of the transformer is shown in , the form of the sentence will be as below:

Table 7-19: ConjunctedTokenTransfer class - After

1	2	3	4	5	6	7	8
منزلت	خاطر	حفظ	حجاب	**اسلامی**	**و**	**زیبایی**	است
N	N	N	N	**N**	**CONJ**	**N**	V

7.4.13 Syntactic Valency Lexicon

The Persian Syntactic Valency Lexicon[16] contains 5600 verbs, including all compulsory and alternative non-verbal elements of these verbs. The information stored for each verb consists of the following: past tense stem, present stem, prefix, non-verbal element of the verb, verbal preposition and syntactic verb structure. In construction of the Persian Valency Lexicon, Persian compound verbs had to be manually identified, as there is no complete existing list of compound verbs based on linguistic characteristics. This proved to be a difficult task, since numerous sequences of words had to be evaluated to determine whether or not they were valid compound verbs. Eventually, a large Persian corpus was examined, and verbs were extracted and processed, and their valencies annotated (Rasooli, Moloodi *et al.*, 2011).

The idea of valency springs from dependency grammar (DG), a theory in which the relation between a head and its dependants determines syntactic structure. Valency lexicons contain information concerning obligatory and optional word complements. This information covers most verbs, and certain nouns and adjectives. The term, valency, is chemistry-inspired, based on an element's ability to share valence

[16] http://dadegan.ir/en/valency-lexicon

electrons of other elements in order to form a compound, or molecule. In the same way, these parts of speech are able to adopt both obligatory and optional dependants, each POS with its dependants being referred to as valency (Rasooli, Moloodi *et al.*, 2011).

In work by Mohadjer-Ghomi (1978), ten compound verb complement types are listed. This is the first work to use the lexical valency theory to identify and address Persian verb valencies. The ten complement types include:

- nominative object
- accusative object
- genitive object
- dative object
- prepositional object
- adverb of quantity
- adverb of direction
- number
- comparison
- verbal complement

A more recent work documenting research in syntactic verb complements in the Persian language is by Ahadi (2002). In Ahadi's work, the author shows eleven verb complements, as follows:

- subject
- direct
- pre-ezafe
- ezafe3
- ezafe complement followed by the morpheme "را"
- enclitic
- place
- quantity
- nominal
- adjectival
- verbal

Information such as the necessity of having a subject or object is included in the valency structure and defined for the verb. Table 7-20 below shows an example of two verbs in the valency lexicon:

Table 7-20: Syntactic Valency Lexicon

Past-stem	Present-Stem	Pre-fix	Nonverbal-element	Verbal-Preposition	Valency-structure
کرد	کن	-	تضمین	-	<فا،مف[را+/-]>
افروخت	افروز	بر	-	-	<فا،(مف)[را+/-]>

7.4.14 VerbValency class

The VerbValency class is used to function with the valency lexicon based on the following methods: LoadValencyLexicon(*string verbLexiconFile*) – this method loads all the data from the valency lexicon based on the processes explained previously.
FindNonVerbalElement(*string nonVerbalElement*) – this method is used for finding verbs with special complements which can be defined with non-verbal element parameters.
IsPastStem(*string stem*) – used to check whether or not the input parameter is the past tense stem.
IsPresentStem(*string stem*) – used to check whether or not the input parameter is the present tense stem.

7.4.15 MissingVerb Transformer class

The verb is the “action denoter” of a sentence, and is probably the main and the most important component of any sentence. The verb carries the weight of the whole sentence and dictates the base structure of the sentence. Without it, the sentence is unfinished and unclear.
In certain cases, sentences from SMT output can occur with missing verbs. The MissingVerb transformer can be used to suggest a correct verb for the sentence. Determining the correct verb for the sentence does not follow any specific pattern, therefore, this transformer was developed for compound verbs. In this group, the non-verbal element of the verb can be used to determine the main verb of the sentence. When the MT system processes a sentence containing a compound verb, it must determine which other words in the sentence are dependants of the main verb (non-

verbal element). To aid in this task, it is possible to compile a list of compound verbs for the machine to use as a reference. Use of this list as a reference can reduce the error rate caused by compound verb misidentification.

The Persian [verb] Valency Lexicon (Rasooli, Moloodi *et al.*, 2011) helps determine the proper verb for a non-verbal element in the sentence. All obligatory and optional non-verbal elements (main-verb dependants) are listed in this lexicon. For example, the word "صرف کردن" is composed of two verbs. Searching for "صرف" in this lexicon will return "کردن" as the main verb in that compound.

Persian generally differentiates between past and present roots for verbs. Because of this, the correct root should be found in order to properly represent the correct tense. Examination of the other verbs in a sentence can show the correct tense intended for that sentence. In some cases, however, there are no other verbs in the sentence to indicate tense. When this is the case, the present tense is chosen by default.

In order to find the missing verb in a sentence, the parse tree of the sentence is examined. As has been mentioned before, general Persian sentence structure is SOV (subject–object–verb). The system's task is to identify any subject with apparent node referring to a verb preceding the subject in the sentence. If such a case occurs, it is determined that this subject is not correctly linked to any verb, since the sentence does not follow the standard SOV structure. The MissingVerb transformer must then find the correct verb.

In this case, it can be assumed that the last word in the sentence can act as a candidate in order to find the non-verbal element in the verb valency lexicon. If this verb with the non-verbal element is found, that verb will be suggested to fill the space of the missing verb. The tense of the verb can then be modified to match that of the subject of the sentence.

Table 7-21 shows an example of this case:

Table 7-21: MissingVerb Transformer class- Before

1	2	3	4	5	6	7	8	9	10	11	12
اما	بیشتر	اوقات	،	زیبایی	آنها	راضی	نمی	با	ظاهر	هستند	و
CONJ	ADJ	N	PUNC	ADJ	PR	ADJ	N	PREP	N	V	CONJ
ADV	PREDEP	SBJ	PUNC	MOS	MOZ	MOS	NEZ	NPP	POSDEP	ROOT	PUNC
11	3	11	3	11	5	11	7	8	9	0	11

13	14	15	16	17	18	19	20	21
آنها	پول	،	زمان	و	انرژی	بیشتری	صرف	.
PR	N	PUNC	N	CONJ	N	ADJ	N	PUNC
SBJ	POSDEP	PUNC	NCONJ	NCONJ	POSDEP	NPOSTMOD	NEZ	PUNC
11	13	14	14	16	17	14	19	11

In this example, the subject "آنها" does not have a linked verb in standard SOV structure, instead it is linked to the verb of the previous sentence (هستند) incorrectly, therefore, the last word of the sentence, " صرف ", should be used as a base to search in the verb valency lexicon to determine the correct main verb. In this case, " کردن" was found. The tense is examined and matched with the subject tense by changing the verb form to "کنند ". The resulting modified sentence is shown in Table 7-22:

Table 7-22: MissingVerb Transformer class - After

1	2	3	4	5	6	7	8	9	10	11	12
اما	بیشتر	اوقات	،	زیبایی	آنها	راضی	نمی	با	ظاهر	هستند	و
CONJ	ADJ	N	PUNC	ADJ	PR	ADJ	N	PREP	N	V	CONJ
ADV	PREDEP	SBJ	PUNC	MOS	MOZ	MOS	NEZ	NPP	POSDEP	ROOT	PUNC
11	3	11	3	11	5	11	7	8	9	0	11

13	14	15	16	17	18	19	20	21	22
آنها	پول	،	زمان	و	انرژی	بیشتری	صرف	می کنند	.
PR	N	PUNC	N	CONJ	N	ADJ	N	V	PUNC
SBJ	POSDEP	PUNC	NCONJ	NCONJ	POSDEP	NPOSTMOD	NVE	PRD	PUNC
11	13	14	14	16	17	14	21	12	11

7.4.16 MozafOfAlefEndedTokenTransformer class

In Persian, there are certain nouns or pronouns following a head noun which signify relationships with the head noun, such as possession or name relation. Such nouns/pronouns are known as Ezafe-dependent. Indication of such in the language is given as a vowel sound /e/, coming immediately after pronunciation of the head noun.

Relation between an ezafe-dependent noun or pronoun and the head noun it is related to, is shown by tag MOZ[17] (Dadegan Research Group 2012).
For example:

ketab -e Ali
book [ezafe] Ali
MOZ (ketab, Ali)
Translation: Ali's book

This transformer searches for words in the dependency parse tree tagged as MOZ. If the head word ends in " ا " /a/, then the character "ی" must be added to the end of that word. Because MOZ comes after one noun, if the POS for the head word is ADJ, then in order to convert this adjective into a noun, "يي" must be added to the end of the word. Table 7-23 and 7-24 below give a sentence and parse tree information showing the transformer at work:

Table 7-23: MozafOfAlefEndedTokenTransformer class - Before

1	2	3	4	5	6	7	8	9	10	11
اما	بیشتر	اوقات	،	زیبا	آنها	راضی	نمی	با	ظاهر	هستند
CONJ	ADJ	N	PUNC	ADJ	PR	ADJ	N	PREP	N	V
ADV	PREDEP	SBJ	PUNC	MOS	MOZ	MOS	NEZ	NPP	POSDEP	ROOT
11	3	11	3	11	5	11	7	8	9	0

Table 7-24: MozafOfAlefEndedTokenTransformer class - After

1	2	3	4	5	6	7	8	9	10	11
اما	بیشتر	اوقات	،	زیبایی	آنها	راضی	نمی	با	ظاهر	هستند
CONJ	ADJ	N	PUNC	N	PR	ADJ	N	PREP	N	V
ADV	PREDEP	SBJ	PUNC	MOS	MOZ	MOS	NEZ	NPP	POSDEP	ROOT
11	3	11	3	11	5	11	7	8	9	0

A flowchart of the whole process of the APE component, with details of the tree levels of transformation, together with pseudo code can be found in Appendix III, sections 2 and 3.

[17] http://dadegan.ir/en/.

7.5 Experiments and Results

7.5.1 Baseline SMT

The SMT system used is run with Joshua 4.0 configured the same as shown in section 5.8, tested using the same training and language models for each test set (Figure 5-5, Tables 5-7, 5-8).

7.5.2 Automatic Evaluation

The translation output before and after our APE approach is applied is scored with both BLEU and NIST, the results of which are shown in Table 7-25.

Table 7-25: Scores of APE based on SMT Joshua version 4.0

Input	Before APE		After APE		BLEU Difference	NIST Difference
	BLEU	NIST	BLEU	NIST		
#1	0.6523	6.5740	0.6770	6.7349	0.0247	0.1609
#2	0.2232	1.0870	0.2187	1.0935	-0.0045	0.0065
#3	0.5914	6.1083	0.7388	6.1089	0.1474	0.0006
#4	0.1365	0.7962	0.1214	0.7064	-0.0151	-0.0898
#5	0.7925	5.7332	0.8716	5.4624	0.0791	-0.2708
#6	0.2738	1.8922	0.2779	1.9196	0.0041	0.0274
#7	0.2945	2.0457	0.2951	2.0333	0.0006	-0.0124
#8	0.4048	2.3940	0.4089	2.4052	0.0041	0.0112

The results generally show increases in both metrics, as also shown in Figures 7-1 and 7-2. The greatest increase in the BLEU score due to the APE was achieved in test set #3, with an increase of about 0.15 BLEU, while the greatest NIST score increase was in test set #1, with a 0.16 increase. However, in certain test sets, the scoring metrics report a decrease in output quality, the worst BLEU score being at a difference of -0.0151, and NIST at -0.27.

The output of the APE method in the experiments is also evaluated with manual (human) evaluation. In order to carry out the manual evaluation, MT output of test sets were assessed by two native Persian speakers.

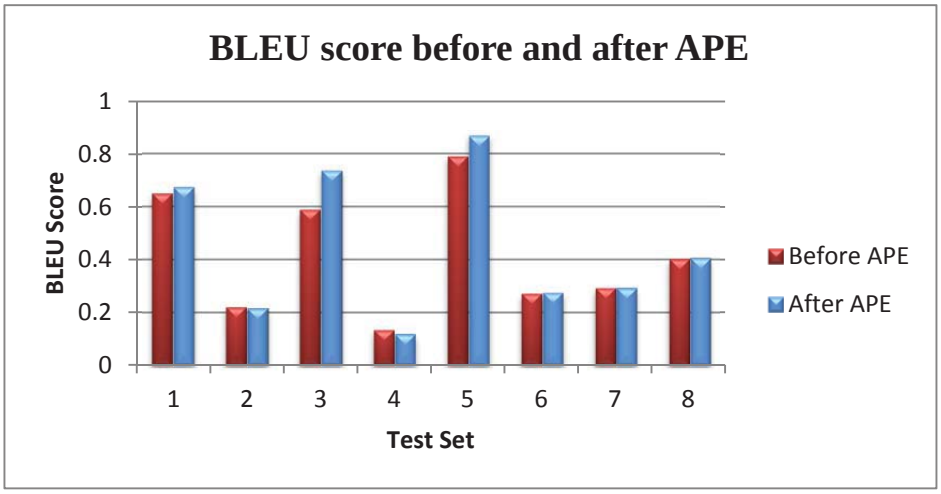


Figure 7-1: BLEU score before and after APE

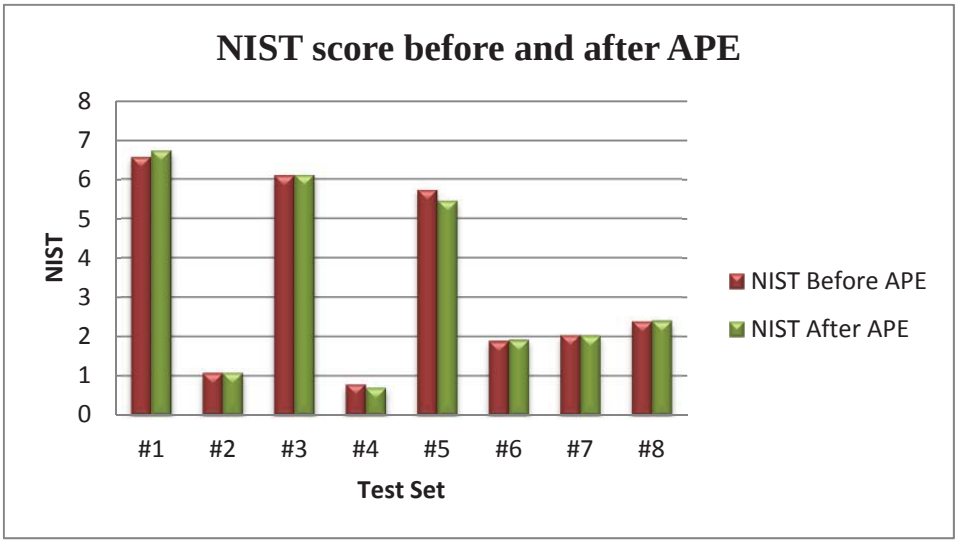


Figure 7-2: NIST score before and after APE

It is proposed that the results showing a decrease in accuracy are mainly due to the lack of training data for the Transliterator module in which some proper names and terms are scripted incorrectly in Persian. Note particularly the decrease in both BLEU and NIST scored output quality for test set #4 in the religious genre: since the parallel corpus has much less data in the religious genre, the quality of SMT is weak for this test set. Furthermore, where there remained some English words in the SMT output that OOVRemover was unable to correct, the Transliterator transformer generated a Persian script which completely changed the meaning of the original sentence.

7.5.3 Manual Evaluation

Based on the suggestion of Marecek, Rosa *et al.* (2011), which states that grammatical correctness of sentences cannot be measured appropriately using BLEU metrics, the proposed model was evaluated using a manual evaluation activity. The same test sets as automatic evaluation were used, containing 153 sentences. The sentences were then translated using SMT and post-edited by our proposed APE approach. The APE output was assigned to two separate annotators, and these were instructed to rank the APE output based on the following criteria:

No Change: No difference between APE output and SMT output

Improved: Certain changes improving fluency

Worse: Certain changes decreasing fluency

The results of the manual evaluation are shown below in Table 7-26.

Table 7-26: Scores of two human evaluators for 153 test sentences

Annotator/Rank	Improved	No Change	Worse
Annotator 1	47	95	11
Annotator 2	43	99	11

Both annotators, who completed the evaluation without discussion with each other, have a very similar judgment of the APE system's output. The results show that the quality of the baseline SMT system output has improved by 29.4%. The rules developed in the APE system are not applicable to more than half (63.4%) of the SMT output. On the other hand, human evaluation also shows that, in some cases, the output is worsened after applying APE.

To reach a more accurate evaluation, both annotator agreements were identified in ranked sentences. Based on their agreement, the quality improvement from the APE system is 25%. In contrast, APE has worsened the machine translation quality by 3%. The comparison of mutual scoring of two annotators is illustrated in Table 7-27.

Table 7-27: Mutual score for both human evaluator I and evaluator II

I / II	Improved	No Change	Worse
Improved	**39**	5	5
No Change	3	**90**	2
Worse	3	4	**4**

The comparison of average percent of manual scores and the agreement of both annotators is shown in Figure 7-3.

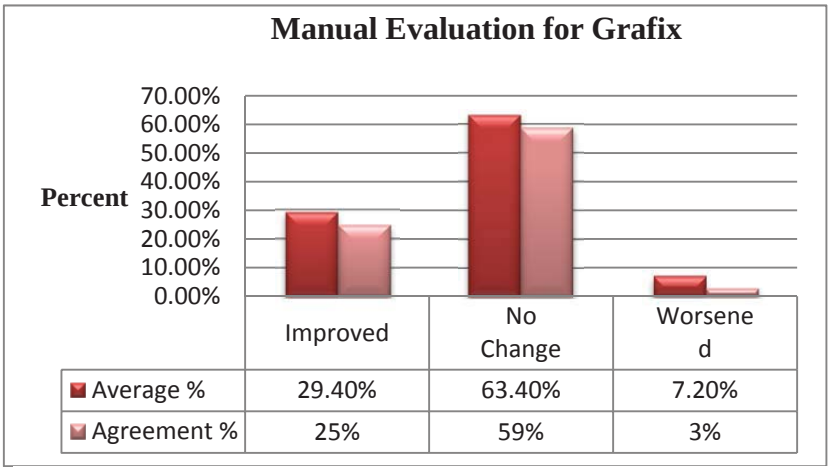


Figure 7-3: Manual evaluation comparison

7.6 Summary

This chapter gives comprehensive details behind our RBMT automatic post-editing approach. It shows the operation steps in each of the three levels, and gives detailed information about each transformer in use, how and where they operates, and clear examples of their use. The results of extensive tests carried out confirm the effectiveness of the novel method, with evaluation performed both automatically and manually by different sources. In the 3% of cases where the translation output has been shown to decrease in accuracy after the APE, explanation is given and these decreased results are justified.

Chapter 8. Discussion and Conclusions

"Patience, persistence and perspiration make an unbeatable combination for success."

~ Napoleon Hill

We were successful in developing methods and techniques to implement an SMT system for the first time on the English/Persian language pair. The main goal was to determine the parameters and characteristics of the system and system components which aided or hindered the output fluency. The initial tests determined what sort of output could be achieved by running the statistical decoder, Moses, with a small bilingual corpus. The output was evaluated using metrics such as BLEU and NIST. After an output was produced and scored, it was necessary to determine which conditions of the system gave rise to lower scores, and which caused an increase in accuracy, to show us which were the most critical areas requiring further attention.

The second part of the work was to repeat each process using different sized parallel corpora, compare results and find a relationship between the size of the parallel corpus and the quality of the output. Although the size of the language model and the training model both affect the translation output, the size of the training model was seen to be more influential. At this stage, there were several issues surrounding sentence alignment which needed to be investigated further. It was believed that accuracy could be increased by categorising the corpus into different subject domains. At the time, the corpus consisted of a mix of genres, such as news stories, poetry, scientific documents and other literature. After tests and test analyses, it was proposed that incorporating linguistic inputs, such as POS (part-of-speech) tagging, parsing, morphological analysis, semantic modelling and a dictionary specific to the domain, would make such a system more robust in terms of accuracy, and they were suggested as an area for development. However, the biggest requirement, and, incidentally, what proved to be an ongoing challenge, was developing and concatenating a parallel corpus large enough for feasible use in an effective SMT system. Other language pairs in successful SMT systems use parallel corpora up to billions of words in size.

Upon obtaining several different sized bilingual corpora, we concatenated and developed them into a single corpus of approximately 3.1 million sentences – now currently the largest English/Persian corpus available. We developed and experimented with a manual alignment of the parallel corpus, using a hybrid sentence

alignment method incorporating both sentence length-based and word correspondence-based models. The results showed this method to be invaluable in obtaining more accurate results from the system. We show that increasing the size of the corpus alone does not necessarily lead to better results. Instead, more attention must be given to the domain of the corpus. There is no doubt that the parallel corpora used in the system are small when compared to other corpora used in training SMT systems for other language pairs, such as German and Chinese, or on Google, which has access to extensive resources. However, the results from our system compare quite favourably, despite the shortage of data.

We have shown that for the English-Persian translation direction, Joshua 4.0, an open-source hierarchical decoder based on SCFGs, is able to achieve a better translation output than that of Moses, as evidenced by its ability to capture long-distance phenomena and model phrasal gaps with non-terminal symbols of SCFGs – cases which are common in the Persian language. Again, it was shown that observation of the corpus domain for both monolingual and bilingual corpora is a critical task of system architecture arrangement.

In our development of an APE approach, we present a novel automatic post-editing model for English-Persian SMT, modeled on a rule-based approach in different levels of transformation. The system performs a range of corrections on sentences, from lexical transformation to complex syntactical rearrangement. It analyses the target sentence (the SMT output in the Persian language) and performs corrections by applying a number of rules which enforce consistency with Persian grammar. The automatic evaluation results, in terms of BLEU metrics, indicate that 75% of test sets show an improvement in the quality of translation after post-editing by the system. Although the improvement in some test sets is small, the SMT output is still improved by up to 0.15 BLEU. Where we faced decreases in the quality of translation by applying the system, it was found that those results originated from the lexical transformer level, and were due to certain OOV words remaining in the original script. The application of OOVRemover and Transliterator only produced a new unknown (incorrect) word as the original word equivalent. However, this occurrence only decreased the BLEU score in one case by up to -0.015 BLEU. The output scores in terms of NIST show that this measurement, just like BLEU, does not reveal the quality of grammatical changes with enough accuracy. On the other hand, manual evaluation scores show that the use of this rules-based approach for an APE system

can yield even better results, with improvement of at least 25% in the translation output.

8.1 Research Contributions

The developments in communication technology, together with the increasing ability for people worldwide to interact, have been two of the main driving forces behind research work in the field of machine translation. As technological developments advance communication opportunities, the task of removing the language barriers becomes more and more critical in order to further the impact of communication solutions. There has been very little work in statistical machine translation between English and Persian, a situation which is mainly put down to the absence of the Persian data necessary for taking the first steps in an SMT system. Several research works from various authors documenting work on this language pair have been published since the commencement of this thesis, however, the translation output shown in these research efforts still falls far short of what is reasonably acceptable as usable language translation. Even output from leading online machine translation services, such as Google Translate and Bing Translator, is considerably unreliable.

In this thesis, we have given details on the construction of what is currently the largest English/Persian bilingual corpus. Our aligning methods show that focus on alignment and corpus domain are more important than corpus size, and, as our results show, high quality translation can be achieved even with the comparatively low data resources available. We plan to make our corpora publicly available after the completion of this thesis in order to aid future research opportunities in this field.

A significant problem with most machine translation systems is the structure of the output. Incorrect grammar, sentence structure and syntax are a continual problem with MT output, and its impact on translation meaning and fluency is often underestimated. Moreover, despite the success well-designed SMT systems have had in general, SMT is more susceptible to these kinds of errors. Currently, research work in APE systems for all language pairs is focussed mostly on the use of statistical methods, with some exploratory work in hybrid approaches. In this thesis we have shown that our hybrid RBMT Transfer-based approach is a superior method, because of its ability to capture grammatical features as a result of linguistic knowledge, and perform transformations

based on linguistic rules specific to the target language. In addition, many of the transformer features in our APE approach, as well as certain techniques for data gathering, can be used for languages very similar to Persian, such as Dari and Tajik. It is also feasible to use our approach in research work for other low-resource languages, such as Maori.

The main motivation for this thesis was to investigate the application of SMT on the English/Persian language pair, and to pioneer research in providing a means of fluent English language translation for over 130 million Persian speakers worldwide. Our research work documented in this thesis has shown success in this undertaking, with our new methods and techniques yielding translation output accuracy with score levels never before seen with any other English-Persian language translation system.

8.2 Directions for Future Work

The progress thus far in the solution of language translation between English and Persian opens more doors and presents more possibilities for work in this area. The most obvious area of potential improvement is the continual accumulation of English/Persian parallel text. The great success the existing SMT systems have had with other high-resource language pairs is largely due to that – the accumulation of extremely large amounts of data. What could be developed is a program to automatically extract English and Persian data as it is published, and add it to an existing open-ended corpus. However, as discussed in this thesis, the sheer quantity of data alone is insufficient if the domain of that data is ignored, and sufficient attention is not given to accurate and stringent alignment techniques.

An area of possible future development is to research a technique to detect the domain of the input, and select the appropriate training model to use based on a domain match between input and corpus. This would allow for greater accuracy of word and phrase probability calculation, and, in effect, provide a better-refined system capable of delivering domain-specific translation.

In the area of the English-Persian APE, in order to increase the improvement and decrease the loss of accuracy, research could be carried out on enriching the bilingual dictionary used in OOVRemover, as well as in the training data for Transliterator. Extending the rules in both shallow and deep levels is another task that could be focused on.

Perhaps one of the most significant areas with great opportunity for further development is in a Persian-English focused APE approach a hybrid Persian-English machine with more focus on improvement in this direction of translation. This would involve extensive research in English grammar, and construction of transformers based on English language syntax.

It has been discussed here in New Zealand that it may be possible to use many of our algorithms and low-resource language-specific characteristics of the system on the Maori language, in order to further develop means to accurately and quickly translate between this low-resource language pair. This has already attracted significant interest, and we have currently been offered work in Maori language translation using various components of our system.

The system that has been developed thus far can be implemented in a speech-to-speech translation system relatively easily. Components of such a system like speech recognition, speech-to-text translation, and corresponding text-to-speech translation, are all easily extended with this SMT and APE system. Although the introduction of the parameter of spoken language introduces more complications, such as the increased likelihood of incorrect grammar and poorly spoken sentences, the potential for the great benefit of developing this area provides significant motivation.

References

Agarwal, A., & Lavie, A. (2008). Meteor, m-bleu and m-ter: Evaluation metrics for high-correlation with human rankings of machine translation output. *In Proceedings of the Third Workshop on Statistical Machine Translation* (pp. 115-118). Columbus, Ohio, USA: Association for Computational Linguistics.

Ahadi, S. (2002). *New Persian language and linguistics: A selected bibliography up to 2001* (Vol. 17): Otto Harrassowitz Verlag.

Ahsan, A., Kolachina, P., et al. (2010). Coupling statistical machine translation with rule-based transfer and generation. *In Proceedings of the 9th Conference of the Association for Machine Translation in the Americas*. Denver, Colorado.

Aiken, M., & Balan, S. (2011). An analysis of Google Translate accuracy. *Translation Journal, 16*(2).

Al-Onaizan, Y., Curin, J., et al. (1999). Statistical machine translation *In Final Report, JHU Summer Workshop* (Vol. 30).

AleAhmad, A., Amiri, H., et al. (2009). Hamshahri: A standard Persian text collection. *Knowledge-Based Systems, 22*(5), 382-387.

Aleahmad, A., Hakimian, P., et al. (2007). N-gram and local context analysis for Persian text retrieval. *In Signal Processing and Its Applications,ISSPA 2007* (pp. 1-4). Sharjah, UAE: IEEE.

Alegria, I., De Ilarraza, A. D., et al. (2005). An open architecture for transfer-based machine translation between Spanish and Basque *Proceedings of the MT Summit X Workshop. Workshop on Open-Source Machine Traslation* (pp. 7-14).

Allen, J., & Hogan, C. (2000). Toward the Development of a Post-editing Module for Raw Machine Translation Output: A Controlled Language Perspective. *Third International Controlled Language Applications Workshop (CLAW-00)* (pp. 62-71).

Ambati, V. (2008). Dependency Structure Trees in Syntax Based Machine Translation. In Advanced MT Seminar Course Report.

Amiri, H., Raja, F., et al. (2007). A survey of part of speech tagging in Persian. *Data base Research Group*.

Amtrup, J. W., Rad, H. M., et al. (2000). Persian-English machine translation: An overview of the Shiraz project *Memoranda in Computer and Cognitive Science MCCS-00-319, NMSU, CRL*.

Arnold, D., Balkan, L., et al. (1994). *Machine translation*: NCC Blackwell Manchester, England.

Assi, S. (1997). Farsi linguistic database (FLDB). *International Journal of Lexicography, 10*(3), 5.

Assi, S., & Abdolhosseini, M. (2000). Grammatical tagging of a Persian corpus. *International Journal of Corpus Linguistics, 5*(1), 69-82.

Bach, N., Eck, M., et al. (2007). The CMU TransTac 2007 eyes-free and hands-free two-way speech-to-speech translation system *In Proceedings of the IWSLT* (Vol. 7).

Bahl, L., Baker, J., et al. (1978). Recognition of continuously read natural corpus. *In Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'78.* (Vol. 3, pp. 422-424): IEEE.

Bakhshaei, S., Khadivi, S., et al. (2010). Farsi - German statistical machine translation through bridge language. *In Proceedings of Telecommunications (IST)* (pp. 557-561). Tehran,Iran: IEEE.

Bakhshaei, S., Khadivi, S., et al. (2010). A study to find influential parameters on a Farsi-English statistical machine translation system. *In Proceedings of Telecommunications (IST)* (pp. 985-991). Tehran,Iran: IEEE.

Banerjee, S., & Lavie, A. (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. *Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 65.

Baradaran Hashemi, H., Shakery, A., et al. (2010). Creating a Persian-English comparable corpus. *Multilingual and Multimodal Information Access Evaluation*, 27-39.

Béchara, H., Ma, Y., et al. (2011). Statistical Post-Editing for a Statistical MT System. *In MT Summit XIII* (pp. 308-315). Xiamen, China.

Belvin, R., May, W., et al. (2004). Creation of a doctor-patient dialogue corpus using standardized patients. *In Proceedings of the Language Resources and Evaluation Conference (LREC)*. Lisbon, Portugal.

Berger, A., Brown, P., et al. (1994). The Candide system for machine translation. *In Proceedings of the ARPA workshop on Human Language Technology* (pp. 157-162). Plainsboro, New Jersey.

Berger, A. L., Brown, P. F., et al. (1996). Language translation apparatus and method using context-based translation models *U.S. Patent and Trademark Office*. Washington, DC.

Bijankhan, M. (2004). The role of the corpus in writing a grammar: An introduction to a software. *Iranian Journal of Linguistics, 19*(2).

Brown, P., Cocke, J., et al. (1988). A statistical approach to French/English translation. *In Proceedings of RIA088 Conference on User-Orinted Content based Text and Image Handling*. Cambridge,Masschusetts: RIAO.

Brown, P., Della Pietra, S., et al. (1990). A statistical approach to machine translation. *Computational Linguistics, 16*(2), 79-85.

Brown, P., Della Pietra, V., et al. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics, 19*(2), 263-311.

Buchholz, S., & Marsi, E. (2006). CoNLL-X shared task on multilingual dependency parsing. *In Proceedings of the Tenth Conference on Computational Natural Language Learning* (pp. 149-164). New York City, USA: Association for Computational Linguistics.

Carbonell, J. G., Cullinford, R. E., et al. (1978). Knowledge-Based Machine Translation: DTIC Document.

Chiang, D. (2005). A hierarchical phrase-based model for statistical machine translation *In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics* (pp. 263-270). University of Michigan, USA.: Association for Computational Linguistics.

Chiang, D. (2007). Hierarchical phrase-based translation. *Computational Linguistics, 33*(2), 201-228.

Chiang, D., Marton, Y., et al. (2008). Online large-margin training of syntactic and structural translation features. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 224-233). Honolulu, Hawaii: Association for Computational Linguistics.

Creutz, M., & Lagus, K. (2005a). Inducing the morphological lexicon of a natural language from unannotated text *International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning*. Espoo, Finland.

Creutz, M., & Lagus, K. (2005b). Inducing the morphological lexicon of a natural language from unannotated text *International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning*. Espoo, Finland.

Creutz, M., & Lagus, K. (2007). Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing, 4*(1), 1-34.

Crystal, D. (2004). The Cambridge Encyclopedia of English Language: Ernst Klett Sprachen.

Dadegan Research Group (2012). Persian Dependency Treebank Version 0.1, from *http://dadegan.ir/en/*.

Davis, C. I. (2012). Tajik-Farsi Persian Transliteration Using Statistical Machine Translation. *In Proceedings of Language Resources and Evaluation Conference (LREC)* Istanbul, Turkey.

de Ilarraza, A. D., Labaka, G., et al. (2008). Statistical postediting: A valuable method in domain adaptation of RBMT systems for less-resourced languages *MATMT 2008: Mixing Approaches to Machine Translation* (pp. 27-34). Donostia-San Sebastian,Spain.

Dehdari, J., & Lonsdale, D. (2008). *A Link Grammar Parser for Persian* (Vol. 1): Cambridge Scholars Press.

Dempster, A. P., Laird, N. M., et al. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1-38.

Deng, Y., & Byrne, W. (2006). MTTK: An alignment toolkit for statistical machine translation. *In Proceedings of the 2006 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume: demonstrations* (pp. 265-268). New York City,USA: Association for Computational Linguistics.

Ding, Y., & Palmer, M. (2005). Machine translation using probabilistic synchronous dependency insertion grammars *In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics* (pp. 541-548). Ann Arbor, USA: Association for Computational Linguistics.

Duda, R. O., Hart, P. E., et al. (1976). Subjective Bayesian methods for rule-based inference systems: DTIC Document.

Dyer, C., Weese, J., et al. (2010). cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. *In Proceedings of the ACL 2010 System Demonstrations* (pp. 7-12). Uppsala,Sweden: Association for Computational Linguistics.

Ehsan, N., & Faili, H. (2010). Towards grammar checker development for Persian language. *In Proceedings The 6th IEEE International Conference on Natural Language Processing and Knowledge Engineering (IEEE NLPK-10)* (pp. 150-157). Beijing, China.

Eisele, A., Federmann, C., et al. (2008). Using Moses to integrate multiple rule-based machine translation engines into a hybrid system. *In Proceedings of the Third Workshop on Statistical Machine Translation* (pp. 179-182). Columbus, Ohio, USA: Association for Computational Linguistics.

Ettelaie, E., Gandhe, S., et al. (2005) Transonics: A practical speech-to-speech translator for English-Farsi medical dialogues. *In Proceedings of 43rd Annual Meeting of the Association for Computational Linguistics: ACL-05 Interactive Poster and Demonstration Sessions* (pp. 89–92).

Farajian, M. A. (2011). PEN: Parallel English-Persian News Corpus. *Proceedings of the 2011th World Congress in Computer Science*. Nevada, USA.: Computer Engineering and Applied Computing.

Foster, G., Kuhn, R., et al. (2006). Phrasetable smoothing for statistical machine translation. *In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Sydney, Australia.

Fry, J., & Center, A. I. (2005). Assembling a parallel corpus from RSS news feeds. *In Proceedings of EBMT Workshop ofMT Summit X* (pp. 59-62). Phuket, Thailand.

Ganitkevitch, J., Cao, Y., et al. (2012). Joshua 4.0: Packing, PRO, and paraphrases. *In Proceedings of the Seventh Workshop on Statistical Machine Translation* (pp. 283-291). Montr´eal, Canada: Association for Computational Linguistics.

Ganjavi, S., Georgiou, P. G., et al. (2003). ASCII based transcription systems for languages with the Arabic script: The case of Persian. *In Proceedings of Automatic Speech Recognition and Understanding, 2003* (pp. 595-600): IEEE.

Georgiou, P., Sethy, P., et al. (2006). An English-Persian Automatic Speech Translator: Recent Developments in Domain Portability and User Modeling. *In Proceedings of ISYC2006, Ayia Napa, Cyprus, July 2006*.

Germann, U., Jahr, M., et al. (2001). Fast decoding and optimal decoding for machine translation. *Annual Meeting-Association for Computational Linguistics* (Vol. 39, pp. 228-235).

Gollins, T., & Sanderson, M. (2001). Improving cross language retrieval with triangulated translation *In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 90-95): ACM.

Gross, A. (1992). Limitations of computers as translation tools *Computers in Translation: A Practical Appraisal* (pp. 96–130).

Hajic, J. (2005). Complex Corpus Annotation: The Prague Dependency Treebank *Insight into the Slovak and Czech Corpus Linguistics* (pp. 54).

Haspelmath, M., & Bibiko, H. J. (2005). *The world atlas of language structures* (Vol. 1): Oxford University Press, USA.

Henderson, F. (2010). Giving a voice to more languages on Google Translate, the official Google translate blog.
Available: "http://googletranslate.blogspot.com/2010/05/giving-voice-to-more-languages-on.html"

Huang, D., Zhao, L., et al. (2010). Mining large-scale comparable corpora from Chinese-English news collections. *Proceedings of the 23rd International Conference on Computational Linguistics* (pp. 472-480). Beijing, China: Association for Computational Linguistics.

Hudson, R. A. (1984). Word grammar: Blackwell Oxford.

Hutchins, W., & Somers, H. (1992). An introduction to machine translation: Academic Press New York.

Isabelle, P., Goutte, C., et al. (2007). Domain adaptation of MT systems through automatic post-editing *In Proceedings of Machine Translation Summit XI* (pp. 255-261). Phuket, Thailand.

Jurafsky, D., Martin, J., et al. (2000). Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition (Vol. 163): MIT Press.

Kathol, A., & Zheng, J. (2008). Strategies for building a Farsi-English SMT system from limited resources. *In Proceedings of INTERSPEECH 2008, Ninth Annual*

Conference of the International Speech Communication Association. Brisbane, Australia.

Kiani, S., Akhavan, T., et al. (2009). Developing a Persian chunker using a hybrid approach. *In Procceedings of Computer Science and Information Technology, 2009. IMCSIT'09. International Multiconference on* (pp. 227-234). Mrągowo, Poland: IEEE.

Kies, D. (2008). Evaluating grammar checkers: A comparative ten-year study. *In Proceedings of 6th International Conference on Education and Information Systems, Technologies and Applications: EISTA*. Orlando, Florida, USA.

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. *In Proceedings of EMNLP* (Vol. 4, pp. 388-395).

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation *In MT Summit* (Vol. 5): Citeseer.

Koehn, P., Hoang, H., et al. (2007a). Moses: Open source toolkit for statistical machine translation *In Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions* (pp. 177-180). Prague, Czech Republic.: Association for Computational Linguistics.

Koehn, P., Hoang, H., et al. (2007). *Moses: Open source toolkit for statistical machine translation*.

Koehn, P., Hoang, H., et al. (2007b). Moses: Open source toolkit for statistical machine translation *In Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions* (pp. 177-180). Prague, Czech Republic.: Association for Computational Linguistics.

Koehn, P., Och, F., et al. (2003a). Statistical phrase-based translation (pp. 48-54). In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1: Association for Computational Linguistics Morristown, NJ, USA.

Koehn, P., Och, F., et al. (2003b). Statistical phrase-based translation (pp. 48-54). In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1: Association for Computational Linguistics Morristown, NJ, USA.

Koehn, P., & Senellart, J. (2010). Convergence of translation memory and statistical machine translation. *In Proceedings of AMTA Workshop on MT Research and the Translation Industry* (pp. 21-31). Denver,Colorado.

Kübler, S., McDonald, R., et al. (2009). Dependency parsing *Synthesis Lectures on Human Language Technologies* (Vol. 1, pp. 1-127).

Kukich, K. (1992). Techniques for automatically correcting words in text *ACM Computing Surveys (CSUR)* (Vol. 24, pp. 377-439).

Kumar, S., Och, F., et al. (2007). Improving word alignment with bridge languages. *In Proceedings of EMNLP-CoNLL* (Vol. 7).

Lagarda, A. L., Alabau, V., et al. (2009). Statistical post-editing of a rule-based machine translation system. *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers* (pp. 217-220). Boulder, Colorado: Association for Computational Linguistics.

Leacock, C., Chodorow, M., et al. (2010). Automated grammatical error detection for language learners. *Synthesis Lectures on Human Language Technologies, 3*(1), 1-134.

Leavitt, J. R., Lonsdale, D. W., et al. (1994). *A reasoned interlingua for knowledge-based machine translation.* Paper presented in the *Proceedings of the Biennial Conference - Canadian Society for Computational Studies of Intelligence*.

Levin, L., Gates, D., et al. (1998). *An interlingua based on domain actions for machine translation of task-oriented dialogues.* Paper presented at the Proceedings of the International Conference on Spoken Language Processing (ICSLP'98).

Li, Z., Callison-Burch, C., et al. (2009). Joshua: An open source toolkit for parsing-based machine translation. *In Proceedings of the Fourth Workshop on Statistical Machine Translation* (pp. 135-139). Athens, Greece: Association for Computational Linguistics.

Li, Z., Callison-Burch, C., et al. (2009). Decoding in Joshua. *Prague Bulletin of Mathematical Linguistics, 91*, 47-56.

Liang, P., Taskar, B., et al. (2006). *Alignment by agreement.* Paper presented at the Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics.

Lopez, A. (2008). Statistical machine translation. *ACM Computing Surveys, 40*(3), 1-49.

Ma, Y., & Way, A. (2009). Bilingually Motivated Domain-Adapted Word Segmentation for Statistical Machine Translation. *In Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)* (pp. 549-557). Athens, Greece: Association for Computational Linguistics.

Mansouri, A., & Faili, H. (2012). State-of-the-art English to Persian Statistical Machine Translation System. *In Artificial Intelligence and Signal Processing (AISP)* (pp. 174-179). Shiraz,Iran: IEEE.

Marcu, D., & Wong, W. (2002). A phrase-based, joint probability model for statistical machine translation. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Vol. 2, pp. 133-139). Philadelphia, USA: Association for Computational Linguistics.

Marecek, D., Rosa, R., et al. (2011). Two-step translation with grammatical post-processing. *In Proceedings of the Sixth Workshop on Statistical Machine Translation* (pp. 426-432). Stroudsburg, PA,USA: Association for Computational Linguistics.

McDonald, R., Pereira, F., et al. (2005). Non-projective dependency parsing using spanning tree algorithms. *In Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing* (pp. 523-530). Vancouver, B.C., Canada: Association for Computational Linguistics.

Megerdoomian, K. (2000). *Persian Computational Morphology: A unification-based approach*: Computing Research Laboratory, New Mexico State University.

Megerdoomian, K. (2000). Unification-based Persian morphology. *In Proceedings of CICLing* (pp. 311-318). Centro de Investigación en Computación-IPN, Mexico.: Citeseer.

Megerdoomian, K. (2004). Developing a Persian part of speech tagger. *In Proceedings of the 1st Workshop on Persian Language and Computer* (pp. 99-105). Tehran,Iran.

Mohadjer-Ghomi, S. (1978). *Eine kontrastive Untersuchung der Satzbaupläne im Deutschen und Persischen*: Burg-Verlag.

Mohaghegh, M., Sarrafzadeh, A., et al. (2010). Improved Language Modeling for English-Persian Statistical Machine Translation. *In Proceedings of SSST-4, Fourth Workshop on Syntax and Structure in Statistical Translation,Coling 2010* (pp. 75-82). Beijing.

Mohaghegh , M., & Sarrafzadeh, H. (2009). *An analysis of the effect of training data variation in English-Persian statistical Machine Translation.* Paper presented at the 6th International conference on Innovations in information Technology United Arab Emirates.

Mohamed, A. A. E. M. (2000). Machine Translation of Noun Phrases from English to Arabic. *Faculty of Engineering, Cairo University, Giza*.

Mohseni, M., Motalebi, H., et al. (2008). A Farsi part-of-speech tagger based on Markov model. *In Proceedings of the 2008 ACM symposium on Applied computing* (pp. 1588-1589): ACM.

Moore, R. (2002). Fast and accurate sentence alignment of bilingual corpora *Lecture notes in computer science* (pp. 135-144).

Mustafa, S. (2005). Character contiguity in N-gram-based word matching: the case for Arabic text searching *Information Processing and Management* (Vol. 41, pp. 819-827).

Nadeau, D., & Foster, G. (2004). Real-time identification of parallel texts from bilingual newsfeed *CLINE 2004, Computational Linguistics in the North East*.

Nagao, M. (1994). Machine translation. *In Proceedings of the Second International Symposium on the Frontiers of Science and Technology* (pp. 273). Kyoto,Japan: United Nations University Press.

Ney, H., Nießen, S., et al. (2000). Algorithms for statistical translation of spoken language *IEEE Transactions on Speech and Audio Processing* (Vol. 8, pp. 24-36).

Nivre, J., Hall, J., et al. (2006). Labeled pseudo-projective dependency parsing with support vector machines. *In Proceedings of the Tenth Conference on Computational Natural Language Learning* (pp. 221-225). New York City, USA: Association for Computational Linguistics.

Nivre, J., & McDonald, R. (2008). Integrating graph-based and transition-based dependency parsers. *In Proceedings of ACL-08: HLT* (pp. 950-958).

Och, F. (2002). *Statistical Machine Translation from Single Word Models to Alignment Templates*: Aachen.

Och, F. (2003). Minimum error rate training in statistical machine translation. *In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics* (Vol. 1, pp. 160-167): Association for Computational Linguistics Morristown, NJ, USA.

Och, F., & Ney, H. (2000). Improved statistical alignment models. *In Proceedings of the 38th Annual Meeting on Association for Computational Linguistics* (pp. 440-447). Hong Kong: Association for Computational Linguistics.

Och, F., & Ney, H. (2002). Discriminative training and maximum entropy models for statistical machine translation. *In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics* (pp. 295-302): Association for Computational Linguistics Morristown, NJ, USA.

Och, F., & Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics, 29*(1), 19-51.

Och, F., & Ney, H. (2004). The alignment template approach to statistical machine translation. *Computational Linguistics, 30*(4), 417-449.

Och, F., Tillmann, C., et al. (1999). Improved alignment models for statistical machine translation. *In Proceedings of the Joint SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora* (pp. 20–28). University of Maryland, College Park, MD, USA.

Oflazer, K., & El-Kahlout, I. D. (2007). Exploring different representational units in English-to-Turkish statistical machine translation. *In Proceedings of the Second Workshop on Statistical Machine Translation* (pp. 25-32). Prague, Czech Republic: Association for Computational Linguistics.

Olteanu, M., Davis, C., et al. (2006). Phramer: an open source statistical phrase-based translator. *In Proceedings of the NAACL 2006 Workshop on Statistical Machine Translation* (pp. 146-149). New York, NY,USA: Association for Computational Linguistics.

Oroumchian, F., Tasharofi, S., et al. (2006). Creating a feasible corpus for Persian POS tagging. *Department of Electrical and Computer Engineering, University of Tehran*.

Oroumchian, F., Tasharofi, S., et al. (2006). Creating a Feasible Corpus for Persian POS Tagging: Technical Report, no. TR3/06, University of Wollongong (Dubai Campus).

Papineni, K., Roukos, S., et al. (2002). BLEU: a method for automatic evaluation of machine translation. *In Proceedings of the 40th annual meeting on association for computational linguistics* (pp. 311-318). Philadelphia, Pennsylvania: Association for Computational Linguistics.

Papineni, K. A., Roukos, S., et al. (1998). Maximum likelihood and discriminative training of direct translation models *In Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing* (Vol. 1, pp. 189-192 vol. 181): IEEE.

Pilevar, A. H. (2011). Using statistical post-editing to improve the output of rule-based machine translation system *Training* (Vol. 330).

Pilevar, M., Faili, H., et al. (2011). TEP: Tehran English-Persian Parallel Corpus. *Computational Linguistics and Intelligent Text Processing*, 68-79.

Pilevar, M. T., & Feili, H. (2010). Persiansmt: A first attempt to english-persian statistical machine translation. *In Proceedings of 15 th International Conference of Statistical Analysis of Textual Data (JADT)* (pp. 1101-1112).

Qasemizadeh, B., Rahimi, S., et al. (2007). The First Parallel Multilingual Corpus of Persian: Toward a Persian BLARK. *In Proceedings of The second workshop on Computational Approaches to Arabic Script-based Languages (CAASL-2)*. California, USA.

Raja, F., Amiri, H., et al. (2007). Evaluation of part of speech tagging on Persian text. *University of Wollongong in Dubai-Papers*, 8.

Ramanathan, A. (2008). Statistical Machine Translation: Ph. D. Seminar Report. IIT-Bombay, India.

Rasooli, M. S., Moloodi, A., et al. (2011). A syntactic valency lexicon for Persian verbs: The first steps towards Persian dependency treebank. *In Proceedings of 5th Language & Technology Conference (LTC): Human Language Technologies as a Challenge for Computer Science and Linguistics* (pp. 227-231).

Rosa, R., Marecek, D., et al. (2012). DEPFIX: A System for Automatic Correction of Czech MT Outputs. *In Proceedings of the Seventh Workshop on Statistical Machine Translation,* . Montreal, Canada: Association for Computational Linguistics.

Sadat, F., Johnson, J. H., et al. (2005). PORTAGE: A phrase-based machine translation system. *In Proceedings of the ACL Workshop on Building and Using Parallel Texts* (pp. 129-132). Ann Arbor, USA.

Schütze, H. (1995). *Distributional part-of-speech tagging.* Paper presented at the Proceedings of the seventh conference on European chapter of the Association for Computational Linguistics.

Senellart, J., Dienes, P., et al. (2001). *New generation systran translation system.* Paper presented at the In Proceedings of MT Summit IIX Senellart J., Yang J., Rebollo A. 2003. SYSTRAN Intuitive Coding Technology. In Proceedings of MT Summit IX.

Shirko, O., Omar, N., et al. (2000). Machine translation of noun phrases from Arabic to English using transfer-based approach. *Journal of Computer Science, 6*(3), 350-356.

Short, D. M. (2008). Indo-European Languages. *Available: http://www.danshort.com/ie/iesatem.htm*

Simard, M., Goutte, C., et al. (2007a). Statistical phrase-based post-editing.

Simard, M., Goutte, C., et al. (2007b). Statistical phrase-based post-editing *In Human LanguageTechnologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference* (pp. 508-515). Rochester, USA.

Soricut, R., Bach, N., et al. (2012). *The SDL language weaver systems in the WMT12 quality estimation shared task.* Paper presented at the Proceedings of the Seventh Workshop on Statistical Machine Translation.

Stolcke, A. (2002). *SRILM-an extensible language modeling toolkit*.

Stolcke, A. (2002). SRILM-an extensible language modeling toolkit. *In Proceedings of the international conference on spoken language processing* (Vol. 2, pp. 901-904). Denver, Colorado.

Taghiyareh, F., Darrudi, E., et al. (2003). Compression of Persian text for web-based applications, without explicit decompression. *WSEAS Transactions on Computers, 2*(4), 961-966.

Terumasa, E. (2007). Rule based machine translation combined with statistical post editor for Japanese to English patent translation. *In Proceedings of MT Summit XI Workshop on patent translation* (Vol. 11, pp. 13-18).

Tillmann, C., & Ney, H. (2003). Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Computational Linguistics, 29*(1), 97-133.

Tillmann, C., Vogel, S., et al. (1997). A DP based search using monotone alignments in statistical translation. (pp. 289-296). In Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics: Association for Computational Linguistics Morristown, NJ, USA.

Tom'as, J., & Casacuberta, F. (2001). Monotone statistical translation using word groups. *In Proceedings of the Machine Translation Summit VIII* (pp. 357-361).

Vasconcellos, M., & Bostad, D. (1992). Machine translation in a high-volume translation environment. *Computers in Translation: A Practical Appraisal*, 58.

Vilar, D., Stein, D., et al. (2010). Jane: Open source hierarchical translation, extended with reordering and lexicon models. *In Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR* (pp. 262-270). Uppsala,Sweden: Association for Computational Linguistics.

Vogel, S., Och, F. J., et al. (2000). *Statistical methods for machine translation*. Berlin, Heidelberg, New York: Springer Verlag.

Wang, H., Wu, H., et al. (2006). Word alignment for languages with scarce resources using bilingual corpora of other language pairs. *In Proceedings of the COLING/ACL in Main Conference Poster Sessions* (pp. 874-881). Sydney, Australia: Association for Computational Linguistics.

Wang, Y. Y., & Waibel, A. (1997). Decoding algorithm in statistical machine translation. *In Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics* (pp. 366-372). Madrid, Spain: Association for Computational Linguistics.

Weaver, W. (1949a). Translation *Mimeographed* (pp. 15-23). New York: Technology Press of the Massachusetts Institute of Technology,Cambridge, MA., and John Wiley & Sons.

Weaver, W. (1949b). Translation *Mimeographed* (pp. 15-23). New York: Technology Press of the Massachusetts Institute of Technology,Cambridge, MA., and John Wiley & Sons.

Wehrli, E. (2007). *Fips, a deep linguistic multilingual parser*. Paper presented at the Proceedings of the Workshop on Deep Linguistic Processing.

Wehrli, E., Nerima, L., et al. (2009). Deep linguistic multilingual translation and bilingual dictionaries. *In Proceedings of the Fourth Workshop on Statistical Machine Translation* (pp. 90-94). Athens, Greece: Association for Computational Linguistics.

Windfuhr, G. (2009). *The Iranian languages*: Routledge New York.

Xuan, H., Li, W., et al. (2011). An Advanced Review of Hybrid Machine Translation (HMT). *Procedia Engineering, 29*, 3017-3022.

Zajac, R., Helmreich, S., et al. (2000). Black-Box/Glass-Box Evaluation in Shiraz. *Workshop on Machine Translation Evaluation at LREC-2000*. Athens, Greece: Citeseer.

Zens, R., & Ney, H. (2007). Efficient phrase-table representation for machine translation with applications to online MT and speech translation. *In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)* (pp. 492-499).

Zhang, Y., Vogel, S., et al. (2004). Interpreting BLEU/NIST scores: How much improvement do we need to have a better system? *In Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC)* (pp. 2051–2054).

Zollmann, A., & Venugopal, A. (2006). Syntax augmented machine translation via chart parsing. *In Proceedings of the Workshop on Statistical Machine Translation* (pp. 138-141). New York City, USA: Association for Computational Linguistics.

Appendix I:

Persian Alphabet

Final	Medial	Initial	Freestanding	Character Name Transcribed
آ	آ / ـآ	آ	آ = الف + ~	*aa*
ـا	ـا	ا	ا	*alef*
ـب	ـبـ	بـ	ب	*be*
ـپ	ـپـ	پـ	پ	*pe*
ـت	ـتـ	تـ	ت	*te*
ـث	ـثـ	ثـ	ث	*se*
ـج	ـجـ	جـ	ج	*jim*
ـچ	ـچـ	چـ	چ	*che*
ـح	ـحـ	حـ	ح	*he*
ـخ	ـخـ	خـ	خ	*kh*
ـد	ـد	د	د	*daal*
ـذ	ـذ	ذ	ذ	*zaal*
ـر	ـر	ر	ر	*re*
ـز	ـز	ز	ز	*ze*
ـژ	ـژ	ژ	ژ	*zhe*
ـس	ـسـ	سـ	س	*sin*
ـش	ـشـ	شـ	ش	*shin*
ـص	ـصـ	صـ	ص	*saad*
ـض	ـضـ	ضـ	ض	*zaad*
ـط	ـطـ	طـ	ط	*taa*
ـظ	ـظـ	ظـ	ظ	*zaa*
ـع	ـعـ	عـ	ع	*eyn*
ـغ	ـغـ	غـ	غ	*gheyn*
ـف	ـفـ	فـ	ف	*fe*
ـق	ـقـ	قـ	ق	*qaaf*
ـک	ـکـ	کـ	ک	*kaaf*
ـگ	ـگـ	گـ	گ	*gaaf*

ل	ـلـ	ـل	ل	*laam*
م	ـمـ	ـم	م	*mim*
ن	ـنـ	ـن	ن	*nun*
و	ـو	و	و	*vaav*
ه	ـهـ	ـه	ه	*he*
ی	ـیـ	ـی	ی	*ye*

Persian Numerals

English	Persian	Pronunciation
1	۱	*yek*
2	۲	*do*
3	۳	*she*
4	۴	*chœha:r*
5	۵	*pœnj*
6	۶	*shish*
7	۷	*hœft*
8	۸	*hœsht*
9	۹	*noh*
0	۰	*sefr*

Appendix II:

Language Model Example

-5.665747	تعلل	-0.099412
-5.665747	گرانوديوريت	-0.099412
-5.665747	رومي	-0.099412
-5.665747	لابلايگردشگران	-0.099412
-5.665747	پشيموني	-0.099412
-4.109445	شعر	-0.230691
-5.063687	كتش	-0.701472
-5.665747	خيمه	-0.099412
-4.966777	ژانگ	-0.196322
-5.665747	نابودكننده	-0.099412
-5.665747	آشغالداني	-0.099412
-4.887596	احساسي	-0.24554
-5.063687	كمسيون	-0.196322
-5.665747	دستآوردهايي	-0.099412
-5.188626	خروشان	-0.099412
-5.364717	متفاوته	-0.400442
-5.665747	ببوسم	-0.099412
-4.410474	شركا	-0.208557

\2-grams:

-4.075135	ما	تبليغ	-0.046887
-1.792416	مسايل	حقوق	-0.046887
-3.227643	قابل	ترجيحي	-0.046887
-4.439754	با	صورتي	-0.046887
-1.909634	از	تقريبا	-0.046887
-0.966792	عجولانه	مي	-0.046887
-1.76754	درواقع	عكسهاي	-0.046887
-2.365191	هماهنگ	انجام	-0.046887
-2.105011	سيستم	آموزشي	-0.046887
-2.484296	ديگري	دوباره	-0.046887

\3-grams:

-2.775724	با	يك	شرط	-0.020669
-0.987722	محدوديت	ديگري	ندارند	-0.020669
-0.756922	فروشگاه	مبل	«	-0.020669
-0.977831	و	پيشنهاد	بحث	-0.020669
-0.507439	دخترک	هديه	اي	-0.020669
-0.384797	قرمز	كرد	.	-0.020669
-0.979199	دنيايي	كه	فكر	-0.020669
-0.686566	بيشتري	هست	كه	-0.020669
-0.689844	او	ماهک	را	-0.020669
-0.896646	هم	گذاشت	تا	-0.020669

\ 4-grams:

-3.380008	سازمان	ملل	متحد	كماكان	-0.359669
-0.982518	رهايي	از	بدهي	عمده	-0.359669

-0.359669	به	مهم	تبليغاتي	سلاح	-0.801374
-0.359669	مردم	بين	هايي	پل	-0.809261
-0.359669	سهيم	براي	را	شركا	-1.32598
-0.23473	را	او	توانست	نمي	-0.022643
-0.058639	روي	را	دستهايش	و	-0.565422
-0.058639	گزارش	در	كه	گونه	-1.699714
-0.359669	بياوريم	ياد	به	بايد	-1.300665
-0.359669	برنده	اونها	،	باشه	-1.20919

\ 5-grams:

معاش	امرار	براي	درآمد	كسب	-0.012321
اين	به	انگيز	شگفت	ويژگي	-0.228962
نتيجه	در	و	كرده	تخريب	-0.243074
مسلح	مردان	،	كرد	مي	-0.228347
حدود	محلي	بيمارستان	يک	به	-0.202324
شوراي	راءي	،	گرفت	آمريكا	-0.153268
كمونيسم	با	كاتوليكها	اختلاف	و	-0.203361
طرف	بر	مراكش	ساحل	در	-0.203208
كنم	مي	تحسين	،	رسانند	-0.088262
شد	نخواهد	ظاهر	آينده	در	-0.17305
به	احتياجي	جانور	اين	كه	-0.084072
مقابله	و	پذيري	آسيب	كاهش	-0.20333
متحد	ملل	سازمان	جرم	و	-0.09878
سالمندي	براي	ماديريد	المللي	بين	-0.124507
ديگر	گروههاي	و	افراد	سرانه	-0.200805

\data\

ngram 1=53860

ngram 2=463177

ngram 3=845769

ngram 4=82971

ngram 5=69835

Test Set, Output, Reference, Score Example:

Test set:

Beauty is the phenomenon that has been noticeable for long.

Principle art shows that beauty is innate in human beings.

But at every period, the type of people's attitude makes a different sense of beauty.

For example, in the past, beauty was in one's personality, behaviour, thoughts and way of talking.

Unfortunately, nowadays expensive cosmetics, fashion clothes, and various surgical beauty operations are the chosen ways by people for being more beautiful.

However most of the time they are not satisfied with their appearance, and they spend more money, time and energy seeking improvement.

The reason is that they have really forgotten the true meaning of beauty, and to them, the beauty which lies within a person’s character is just a nice thought.

They, especially women, have forgotten that Iranian women have been popular and beautiful through their maintenance of Islamic hijab and dignity.

Output:

زیبایی پدیده ای است که از مدتها قبل مورد توجه بوده است.

هنر اصلی نشان می دهد که زیبایی یک پدیده ذاتی است در انسان چهارشنبه در هر زمان ، نوع نگرش به افرادی را می درک متفاوتی از زیبایی را نشان می دهد.

است برای مثال در زمان قدیم ، زیبایی در شخصیت ، رفتار ، افکار و روش صحبت کردن بود.

متاسفانه امروزه ، لوازم آرایشی گرانقیمت ، لباسهای مد روز ، و جراحی های مختلف زیبایی عملیات زیبایی ، روشهای مردم برای زیباتر شدن انتخاب.

اما اکثر اوقات آنها را از ظاهرشان راضی نمی باشند ، و با دقت پول ، زمان و انرژی بیشتری صرف آن می کنند.

دلیلش این است که آنها را به طور قطع معنی واقعی زیبایی را گم کرده اند ، درحالیکه ، این فکر یا عقیده خوب است که فرد را زیبا و دوست داشتنی می کند.

او بخصوص زنان ، فراموش کردند که بانوان ایرانی بوده اند و به خاطر حفظ حجاب و متانتشان زیبا و مشهورند.

Reference:

زیبایی پدیده ای است که از مدتها قبل مورد توجه بوده است.

هنر اصلی نشان می دهد که زیبایی یک پدیده ذاتی است.اما در هر زمان ،نوع نگرش انسانها ،درک متفاوتی از زیبایی را نشان می دهد.

برای مثال،در زمان قدیم، زیبایی در شخصیت،رفتار،افکار و روش صحبت کردن بود.

متاسفانه امروزه ،لوازم آرایشی گرانقیمت،لباسهای مد روز،و جراحی های مختلف زیبایی، روشهای مردم برای زیباتر شدن است.

اما اکثر اوقات آنها را از ظاهرشان راضی نمی کند و درنتیجه پول،زمان و انرژی بیشتری صرف آن می کنند.

دلیلش این است که آنها به طور قطع معنی واقعی زیبایی را گم کرده اند ،درحالیکه ، این فکر یا عقیده خوب است که فرد را زیبا و دوست داشتنی میکند.

این افراد و بخصوص زنان ، فراموش کردند که بانوان ایرانی به خاطر حفظ حجاب و متانتشان زیبا و مشهورند.

Score:

Processing 7 sentences...

Evaluating candidate translations in plain file test/1/test.output.1best...

BLEU_precision(1) = 151 / 175 = 0.8629

BLEU_precision(2) = 134 / 168 = 0.7976

BLEU_precision(3) = 121 / 161 = 0.7516

BLEU_precision(4) = 108 / 154 = 0.7013

BLEU_precision = 0.7761

Length of candidate corpus = 175

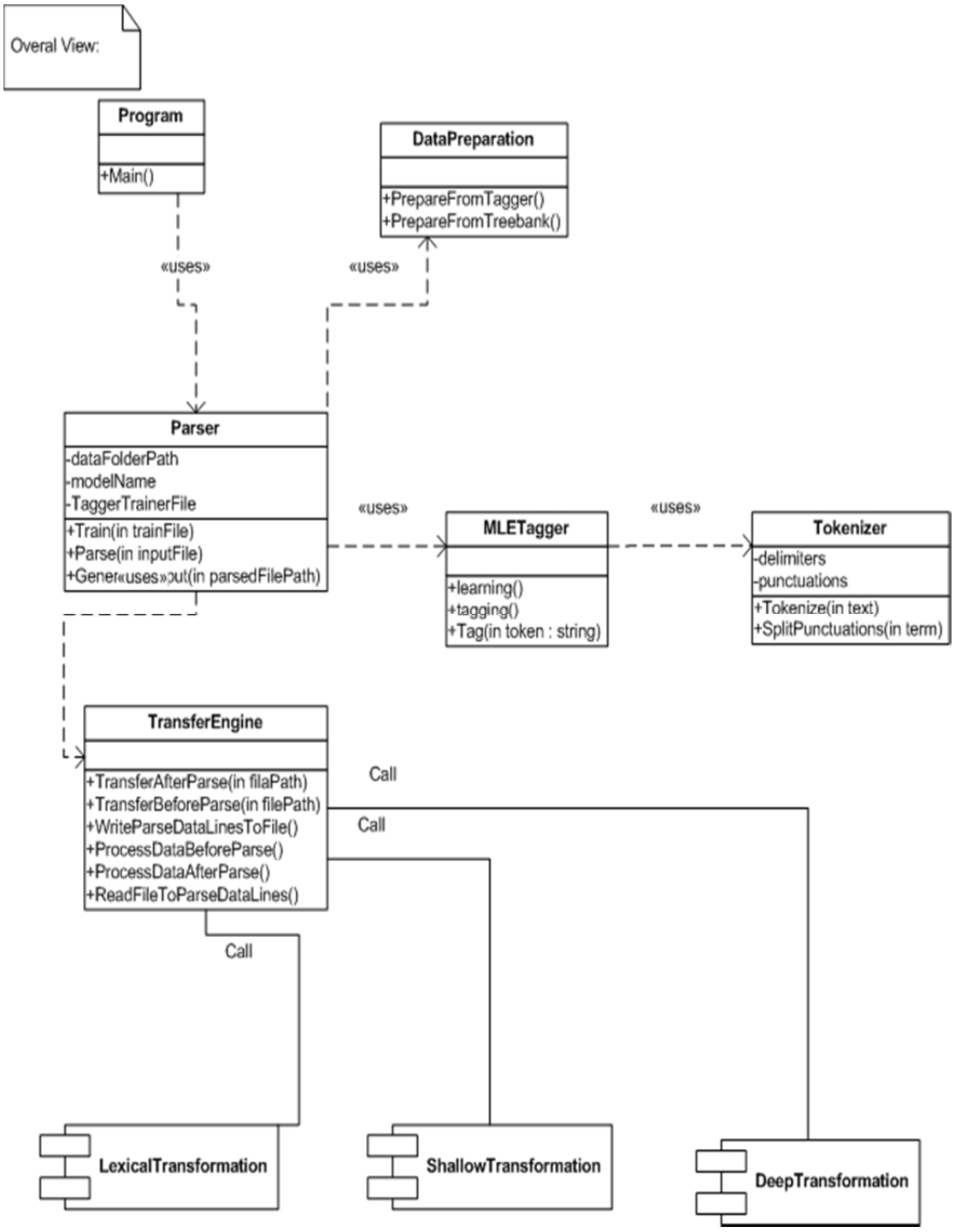
Effective length of reference corpus = 164

BLEU_BP = 1.0000

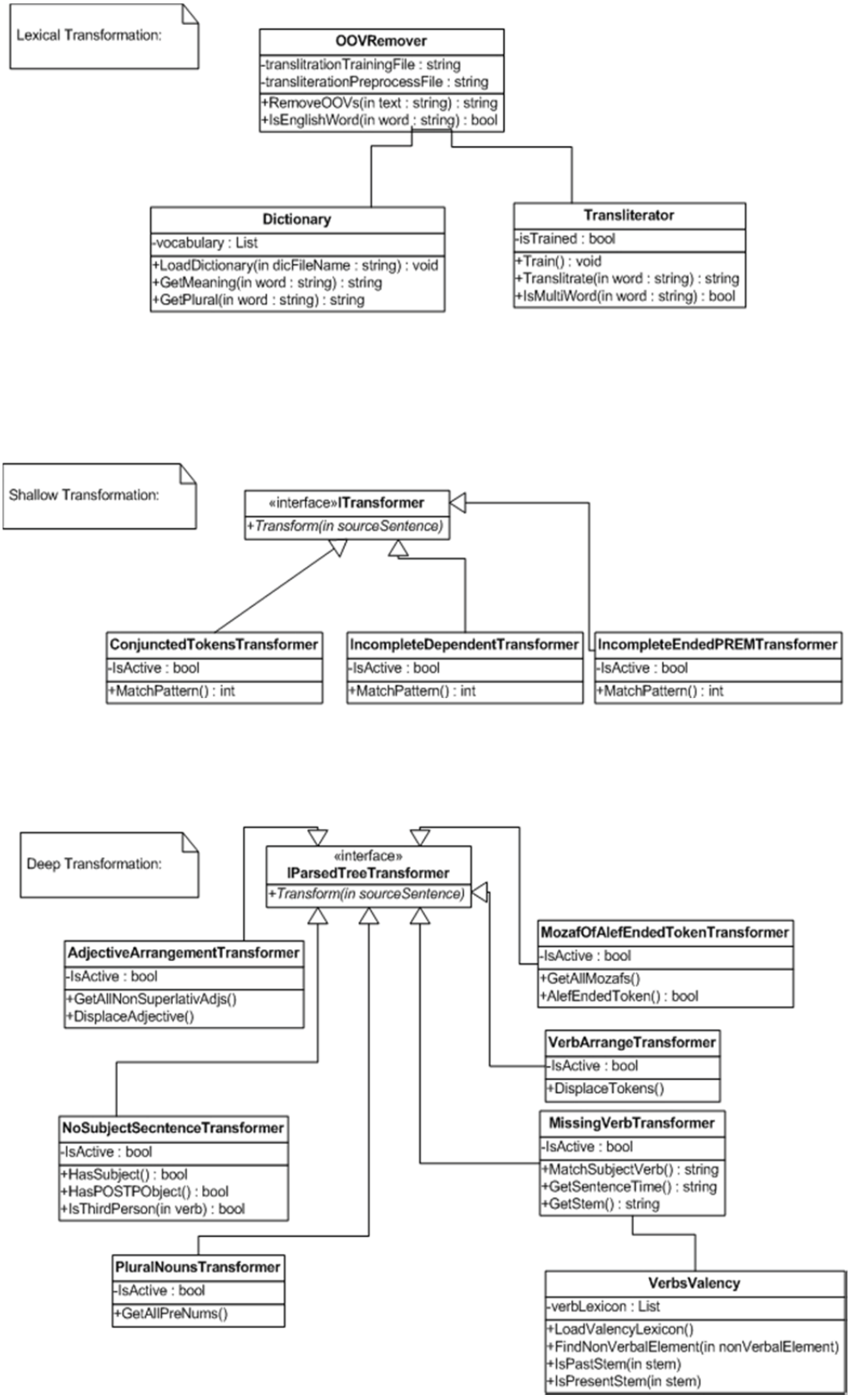
=> BLEU = 0.7761

Appendix III:

APE Diagram 1



APE Diagram 2



Example of MLETagger on Output and Reference Set:

Output text:

زیبایی	N_SING
پدیده ای	N_SING
است	V_PRE
که	CON
از	P
مدتها	N_PL
قبل	N_SING
مورد	N_SING
توجه	N_SING
بوده	ADJ_INO
است	V_PRE
هنر	N_SING
.principle_OOV	N_SING
نشان	N_SING
دهد	V_SUB
که	CON
زیبایی	N_SING
در	P
پدیده	N_SING
ذاتی	ADJ_SIM
است	V_PRE
.but_OOV	N_SING
در	P
هر	QUA
؛	DELM
از	P
نوع	N_SING
نگرش	N_SING
مردم	N_SING
اي	OH
دیگر	ADJ_SIM
حس	N_SING
زیبایی	N_SING
می کند	V_PRS
.	DELM
برای	P
مثال	N_SING
در	P
گذشته	ADJ_SIM
؛	DELM
زیبایی	N_SING
در	P
شخصیت	N_SING
،	DELM
رفتار	N_SING
بود	V_PA
،	DELM
افکار	N_PL
و	CON
روش	N_SING
صحبت	N_SING
کرد	V_PA
.	DELM

متاسفانه	ADV_NI
امروزه	ADV_TIME
،	DELM
لوازم	N_PL
آرایشی	ADJ_SIM
گرانقیمت	ADJ_SIM
،	DELM
لباسهای	N_PL
مد	N_SING
روز	N_SING
،	DELM
لباس	N_SING
و	CON
جراحی	N_SING
های	MORP
مختلف	ADJ_SIM
زیبایی	N_SING
،	DELM
مردم	N_SING
برای	P
زیباتر	ADJ_CMPR
شدن	N_SING
هستند	V_PRS
.	DELM
اما	CON
خیلي	ADV_NI
وقت	N_SING
،	DELM
ظاهرشان	N_SING
راضی	ADJ_SIM
نمی	N_SING
با	P
ظاهر	N_SING
و	CON
آنها	PRO
پول	N_SING
،	DELM
زمان	N_SING
و	CON
انرژی	N_SING
بیشتری	ADJ_CMPR
صرف	N_SING
.	DELM
دلیلش	N_SING
این	DET
است	V_PRE
که	CON
آنها	PRO
معنی	N_SING
واقعی	ADJ_SIM
زیبایی	N_SING
را	P
گم	ADJ_SIM
کرده	ADJ_INO
اند	DELM
،	DELM
ضمن	P
این	DET
خوب	ADJ_SIM
است	V_PRE
اگرچه	CON

```
این                DET
فرد                N_SING
را                 P
زیبا               ADJ_SIM
و                  CON
دوست               N_SING
داشتنی             N_SING
کند                V_SUB
.                  DELM
بخصوص              CON
زنان               N_PL
،                  DELM
فراموش             ADJ_SIM
کردند              V_PA
که                 CON
بانوان             N_PL
ایرانی             ADJ_SIM
بوده               ADJ_INO
اند                DELM
و                  CON
زیبا               ADJ_SIM
عمدة               N_SING
خاطر               N_SING
حفظ                N_SING
حجاب               N_SING
و                  CON
منزلت              N_SING
اسلامی             ADJ_SIM
است                V_PRE
.                  DELM


Started at 14:46 | Finished at 14:46
------------------------------------------------------------------------
-------------

STATISTICS

   TOTAL No. of test words: 144
   ---------------------------
   No. of seen words: 97
       No. of correct seen words: 29
   ---------------------------
   No. of Unseen words: 47
       No. of correct Unseen words (heuristic rules + default tag:
'N_SING'): 39
       No. of correct Unseen words (heuristic rules + default tag:
'DEFAULT'): 0
       No. of correct Unseen words (default tag: 'N_SING'): 47
       No. of correct Unseen words (default tag: 'DEFAULT'): 0
   ---------------------------


PERFORMANCE

   Seen Words Accuracy:  29.9
   ---------------------------
   UNSeen Words Accuracy:
       with Heuristic Rules
             default tag: 'DEFAULT': 0
             default tag: 'N_SING':    82.98
```

```
    without Heuristic Rules
          default tag: 'DEFAULT': 0
          default tag: 'N_SING':    100
--------------------------
Overall
    with Heuristic Rules
          default tag: 'DEFAULT': 20.14
          default tag: 'N_SING':    47.22
    without Heuristic Rules
          default tag: 'DEFAULT': 20.14
          default tag: 'N_SING':    52.78
```

Reference text:

زیبایی	N_SING
پدیده	N_SING
ای	OH
است	V_PRE
که	CON
از	P
مدتها	N_PL
قبل	N_SING
مورد	N_SING
توجه	N_SING
بوده	ADJ_INO
است	V_PRE
.	DELM
هنر	N_SING
اصلی	ADJ_SIM
نشان	N_SING
می	N_SING
دهد	V_SUB
که	CON
زیبایی	N_SING
یک	N_SING
پدیده	N_SING
ذاتی	ADJ_SIM
اما.است	N_SING
در	P
هر	QUA
زمان	N_SING
،	DELM
نوع	N_SING
نگرش	N_SING
انسانها	N_PL
،	DELM
درک	N_SING
متفاوتی	ADJ_SIM
از	P
زیبایی	N_SING
را	P
نشان	N_SING
می	N_SING
دهد	V_SUB
.	DELM
برای	P
مثلا	ADV_EXM
،	DELM
در	P

زمان	N_SING
قدیم	ADJ_SIM
،	DELM
زیبایی	N_SING
در	P
شخصیت	N_SING
،	DELM
رفتار	N_SING
،	DELM
افکار	N_PL
و	CON
روش	N_SING
صحبت	N_SING
کردن	N_SING
بود	V_PA
.	DELM
تاسفانه	N_SING
امروزه	ADV_TIME
،	DELM
لوازم	N_PL
آرایشی	ADJ_SIM
گرانقیمت	ADJ_SIM
،	DELM
لباسهای	N_PL
مد	N_SING
روز	N_SING
،	DELM
و	CON
جراحی	N_SING
های	MORP
مختلف	ADJ_SIM
زیبایی	N_SING
،	DELM
روشهای	N_PL
مردم	N_SING
برای	P
زیباتر	ADJ_CMPR
شدن	N_SING
است	V_PRE
.	DELM
اما	CON
اکثر	QUA
اوقات	N_PL
آنها	PRO
را	P
از	P
ظاهرشان	N_SING
راضی	ADJ_SIM
نمی	N_SING
کند	V_SUB
و	CON
درنتیجه	ADV_NI
پول	N_SING
،	DELM
زمان	N_SING
و	CON
انرژی	N_SING
بیشتری	ADJ_CMPR
صرف	N_SING
آن	PRO
می	N_SING

کنند	V_SUB
.	DELM
دلیلش	N_SING
این	DET
است	V_PRE
که	CON
آنها	PRO
به	P
طور	N_SING
قطع	N_SING
معنی	N_SING
واقعی	ADJ_SIM
زیبایی	N_SING
را	P
گم	ADJ_SIM
کرده	ADJ_INO
اند	DELM
،	DELM
درحالیکه	CON
،	DELM
این	DET
فکر	N_SING
یا	CON
عقیده	N_SING
خوب	ADJ_SIM
است	V_PRE
که	CON
فرد	N_SING
را	P
زیبا	ADJ_SIM
و	CON
دوست	N_SING
داشتنی	N_SING
میکند	V_PRS
.	DELM
این	DET
افراد	N_PL
و	CON
بخصوص	CON
زنان	N_PL
،	DELM
فراموش	ADJ_SIM
کردند	V_PA
که	CON
بانوان	N_PL
ایرانی	ADJ_SIM
به	P
خاطر	N_SING
حفظ	N_SING
حجاب	N_SING
و	CON
متانتشان	N_SING
زیبا	ADJ_SIM
و	CON
مشهورند	N_SING
.	DELM

```
Started at 15:51  |  Finished at 15:51
------------------------------------------------------------------------
-------------

STATISTICS

   TOTAL No. of test words: 162
   --------------------------
   No. of seen words: 98
       No. of correct seen words: 28
   --------------------------
   No. of Unseen words: 64
       No. of correct Unseen words (heuristic rules + default tag:
'N_SING'): 54
       No. of correct Unseen words (heuristic rules + default tag:
'DEFAULT'): 0
       No. of correct Unseen words (default tag: 'N_SING'): 64
       No. of correct Unseen words (default tag: 'DEFAULT'): 0
   --------------------------


PERFORMANCE

   Seen Words Accuracy:  28.57
   --------------------------
   UNSeen Words Accuracy:
       with Heuristic Rules
             default tag: 'DEFAULT': 0
             default tag: 'N_SING':    84.38
       without Heuristic Rules
             default tag: 'DEFAULT': 0
             default tag: 'N_SING':    100
   --------------------------
   Overall
       with Heuristic Rules
             default tag: 'DEFAULT': 17.28
             default tag: 'N_SING':    50.62
       without Heuristic Rules
             default tag: 'DEFAULT': 17.28
             default tag: 'N_SING':    56.79
```