






Universitat Autònoma de Barcelona

ADVERTIMENT. L'accés als continguts d'aquesta tesi queda condicionat a l'acceptació de les condicions d'ús establertes per la següent llicència Creative Commons:  http://cat.creativecommons.org/?page_id=184

ADVERTENCIA. El acceso a los contenidos de esta tesis queda condicionado a la aceptación de las condiciones de uso establecidas por la siguiente licencia Creative Commons:  <http://es.creativecommons.org/blog/licencias/>

WARNING. The access to the contents of this doctoral thesis it is limited to the acceptance of the use conditions set by the following Creative Commons license:  <https://creativecommons.org/licenses/?lang=en>

AUTONOMOUS UNIVERSITY OF BARCELONA

DOCTORAL THESIS

**Reliable Training Scenarios
for Dealing with Minimal
Parallel-Resource Language Pairs
in Statistical Machine Translation**

Author:

Benyamin AHMADNIAYEBOSARI

Supervisor:

Dr. Javier SERRANO GARCIA

PhD Program in Electrical and Telecommunication Engineering

*A thesis submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy*

in the

Department of Telecommunication and Systems Engineering
School of Engineering

November 2017

Abstract

Over the years, various changes have been made to Machine Translation (MT) which is mainly applied for Natural Language Processing (NLP). Statistical Machine Translation (SMT) is one of the preferred approaches to MT, and various improvements could be detected in this approach, specifically in the output quality in a number of systems for language pairs since the advances in computational power, together with the exploration of new methods and algorithms have been made.

When we ponder over the development of SMT systems for many language pairs, the major bottleneck that we will find is the lack of training parallel data. Due to the fact that lots of time and effort is required to create these corpora, they are available in limited quantity, genre, and language.

SMT models learn that how they could do translation through the process of examining a bilingual parallel corpus that contains the sentences aligned with their human-produced translations. However, the output quality of SMT systems is heavily dependent on the availability of massive amounts of parallel text within the source and target languages. Hence, an important role is played by the parallel resources so that the quality of SMT systems could be improved. We define minimal parallel-resource SMT settings possess only small amounts of parallel data, which can also be seen in various pairs of languages.

The performance achieved by current state-of-the-art minimal parallel-resource SMT is highly appreciable, but they usually use the monolingual text and do not fundamentally address the shortage of parallel training text. Creating enlargement in the parallel training data without providing any sort of guarantee on the quality of the bilingual sentence pairs that have been newly generated, is also raising concerns. The limitations that emerge during the training of the minimal parallel-resource SMT prove that the current systems are incapable of producing the high-quality translation output.

In this thesis, we have proposed a "direct-bridge combination" scenario as well as a "round-trip training scenario", that the former is based on bridge language technique, while the latter one is based on retraining approach, for dealing with minimal parallel-resource SMT systems.

Our main aim for putting forward the direct-bridge combination scenario is that we might bring it closer to state-of-the-art performance. This scenario has been proposed to maximize the information gain by choosing the appropriate portions of the bridge-based translation system that do not interfere with the direct-based translation system which is trusted more. Furthermore, the round-trip training scenario has been proposed to take advantage of the readily available generated bilingual sentence pairs to build high-quality SMT system in an iterative behaviour; by selecting high-quality subset of generated sentence pairs in target side, preparing their suitable correspond source sentences, and using them together with the original sentence pairs to retrain the SMT system.

The proposed methods are intrinsically evaluated, and their comparison is made against the baseline translation systems. We have also conducted the experiments in the aforementioned proposed scenarios with minimal initial bilingual data. We have demonstrated improvement made in the performance through the use of proposed methods while building high-quality SMT systems over the baseline involving each scenario.

Acknowledgements

I would like to express my deep gratitude to my advisor, Dr. Javier Serrano, who has had a profound influence on my research and studies. He introduced me to the fascinating field of Natural Language Processing and Statistical Machine Translation, and taught me a great deal of valuable research skills. Besides being a knowledgeable teacher, he has also been a helpful friend with a strong personality which was always inspiring.

I want to thank Dr. Gholamreza Haffari, for his guidance and support along this thesis work during my fruitful research-visiting stay in the Faculty of Information Technology at Monash University, Australia. Probably, this doctoral thesis would have not been possible without his help.

At this point, I would like to express my everlasting gratitude to Dr. Mojtaba Sabbagh Jafari (Vali-e-Asr University of Rafsanjan, Iran), and Dr. Nik-Mohammad Balouchzahi (University of Sistan and Baluchestan, Iran). I marvel at their invaluable and insightful technical support, their exceptional generosity, and encouragement shown towards me.

I am also thankful to the past and current members of Transmedia-Catalonia research group, and Telecommunication and Systems Engineering Department at the School of Engineering at Autonomous University of Barcelona, Spain.

Very special thanks to my friend, Shekoofeh Dadgostar (University of Granada, Spain), for all her mental and unconditional support in the most stressful and hardest moments.

Of course, I would like to thank my beloved family members. My dear parents (Farideh and Khosro), and my lovely sister (Tamara), have endowed me with the curiosity about the world, and supported me through my life even from thousands miles away. Their continuing encouragement lightens my path into higher education. Thank you all.

Finally, thanks to all my friends, families, colleagues, reviewers, and everyone else not mentioned here, in Spain, Australia, and Iran, for all their supports in many aspects during the course of the research.

Contents

| | |
|--|-------------|
| Abstract | iii |
| Acknowledgements | v |
| List of Figures | xi |
| List of Tables | xiii |
| 1 Introduction | 1 |
| 1.1 Context and Motivation | 1 |
| 1.2 Thesis Objectives | 6 |
| 1.3 Thesis Outline | 7 |
| 2 Background | 9 |
| 2.1 Computational Linguistics | 9 |
| 2.2 Natural Language Processing | 9 |
| 2.3 Minimal-Resource Languages | 11 |
| 2.4 Machine Translation | 12 |
| 2.4.1 Rule-Based Machine Translation | 14 |
| 2.4.2 Corpus-Based Machine Translation | 15 |
| 2.4.3 Hybrid Machine Translation | 15 |
| 2.5 Statistical Machine Translation | 16 |
| 2.6 Parallel Corpus Alignment | 18 |
| 2.7 Translation Model Training | 19 |
| 2.8 Language Model Training | 20 |
| 2.9 Decoding | 22 |
| 2.10 Evaluation | 22 |
| 2.11 Decoding Software Packages | 24 |
| 2.12 State of the art | 24 |
| 2.12.1 Learning Frameworks for Minimal Parallel-Resource SMT | 25 |
| 2.12.1.1 Semi-Supervised Learning | 25 |
| 2.12.1.2 Active Learning | 29 |
| 2.12.1.3 Deep Learning | 32 |
| 2.12.2 Pivoting Framework for Minimal Parallel-Resource SMT | 33 |
| 2.12.3 Other Research Lines | 34 |
| 2.12.3.1 Bilingual Lexicon Induction for Minimal Parallel-Resource SMT | 35 |
| 2.12.3.2 Monolingual Collocation for Minimal Parallel-Resource SMT | 36 |
| 2.12.3.3 Domain Adaptation for Minimal Parallel-Resource SMT | 37 |
| 2.13 Summary | 39 |

| | | |
|----------|--|-----------|
| 3 | Phrase-Based Translation Models for SMT Systems | 41 |
| 3.1 | Introduction | 42 |
| 3.2 | Classical Phrase-Based Translation Model | 42 |
| 3.2.1 | Noisy-Channel Model | 43 |
| 3.2.2 | Log-Linear Model | 44 |
| 3.2.3 | Feature Functions | 45 |
| 3.2.4 | Phrase Extraction | 47 |
| 3.2.5 | Phrase-Table Induction | 47 |
| 3.2.6 | Learning Weights | 48 |
| 3.2.7 | Solving Search-Problem | 48 |
| 3.2.8 | Classical Model Decoding | 49 |
| 3.3 | Hierarchical Phrase-Based Translation Model | 51 |
| 3.3.1 | Hierarchical Rules | 52 |
| 3.3.2 | Grammars Definition | 52 |
| 3.3.3 | Rules Extraction | 53 |
| 3.3.4 | Rule Parameters Learning | 54 |
| 3.3.5 | Standard Features | 55 |
| 3.3.6 | Hierarchical Model Decoding | 55 |
| 3.4 | Experimental Framework | 56 |
| 3.4.1 | Experimental Set-Up | 57 |
| 3.4.2 | Implementation | 58 |
| 3.4.3 | Results Analysis and Evaluation | 59 |
| 3.5 | Comparative Performance of Phrase-Based Models | 65 |
| 3.5.1 | Experiments Setting | 65 |
| 3.5.2 | Results | 66 |
| 3.6 | Discussion | 72 |
| 3.7 | Summary | 73 |
| 4 | Direct-Bridge Combination for Minimal Parallel-Resource SMT | 75 |
| 4.1 | Introduction | 75 |
| 4.2 | Bridge Language Theory | 77 |
| 4.3 | Bridging Approaches | 77 |
| 4.3.1 | Transfer Approach | 78 |
| 4.3.2 | Synthetic Corpus Approach | 79 |
| 4.3.3 | Triangulation Approach | 79 |
| 4.3.3.1 | Phrase Translation Probabilities | 80 |
| 4.3.3.2 | Lexical Reordering Weights | 81 |
| 4.3.4 | Interpolated Model | 81 |
| 4.4 | Proposed Improvements in Bridge Language Technique | 83 |
| 4.4.1 | Interpolating Bilingual Texts | 83 |
| 4.4.2 | Combining Phrase-tables | 84 |
| 4.5 | Experiments | 84 |
| 4.6 | Results and Evaluation | 85 |
| 4.7 | Proposed Method | 90 |
| 4.7.1 | Optimized Direct-Bridge Combination Method | 90 |
| 4.7.2 | ODBC Method Experiments | 92 |
| 4.7.2.1 | Baseline Systems Evaluation | 93 |
| 4.7.2.2 | Baseline Combination | 93 |
| 4.7.2.3 | Direct-Bridge Combination | 94 |
| 4.8 | Discussion | 96 |
| 4.9 | Summary | 97 |

| | | |
|----------|---|------------|
| 5 | Round-Trip Training Scenario for Minimal Parallel-Resource SMT | 99 |
| 5.1 | Introduction | 100 |
| 5.2 | Bootstrapping Analysis | 102 |
| 5.2.1 | Self-Training Mechanism | 105 |
| 5.2.2 | Co-Training Mechanism | 108 |
| 5.3 | Round-Trip Training Theory | 111 |
| 5.4 | High-Quality Translations Selection | 112 |
| 5.5 | Round-Trip Training Mechanism | 113 |
| 5.6 | Round-Trip Training Optimization | 114 |
| 5.7 | Round-Trip Training Algorithms | 115 |
| 5.8 | Experimental Framework | 118 |
| 5.8.1 | Data Preparation | 118 |
| 5.8.2 | Baseline Phrase-Based SMT Architecture | 118 |
| 5.8.3 | Implementation | 119 |
| 5.9 | Results Analysis and Evaluation | 120 |
| 5.10 | Discussion | 123 |
| 5.11 | Summary | 123 |
| 6 | Conclusions | 125 |
| 6.1 | Overall Review | 125 |
| 6.2 | Future Work Directions | 127 |
| | Bibliography | 129 |

List of Figures

| | | |
|------|--|----|
| 3.1 | Alignment sample of Persian-English parallel sentence | 47 |
| 3.2 | The stack-based beam-search schema | 49 |
| 3.3 | The learning curve of Classical (Moses) and Hierarchical (Cdec) phrase-based translation models for Spanish-English translation and vice versa using 4-gram language models according to BLEU scores | 60 |
| 3.4 | The learning curve of Classical (Moses) and Hierarchical (Cdec) phrase-based translation models for English-Persian translation and vice versa using 4-gram language models according to BLEU scores | 61 |
| 3.5 | The learning curve of Classical (Moses) and Hierarchical (Cdec) phrase-based translation models for Persian-Spanish translation and vice versa using 4-gram language models according to BLEU scores. | 63 |
| 3.6 | The learning curve of Classical (Moses) and Hierarchical (Cdec) phrase-based translation models for Persian-Spanish translation and vice versa using 4-gram language models on Tanzil parallel corpus according to BLEU scores | 64 |
| 3.7 | The performance chart of Spanish-English translation task in different kinds of language models according to BLEU. | 67 |
| 3.8 | The performance chart of English-Spanish translation task in different kinds of language models according to BLEU. | 68 |
| 3.9 | The performance chart of English-Persian translation task in different kinds of language models according to BLEU. | 69 |
| 3.10 | The performance chart of Persian-English translation task in different kinds of language models according to BLEU. | 70 |
| 3.11 | The performance chart of Persian-Spanish translation task in different kinds of language models according to BLEU. | 71 |
| 3.12 | The performance chart of Spanish-Persian translation task in different kinds of language models according to BLEU. | 72 |
| 4.1 | The interpolation method schematic between source, pivot, and target models | 82 |
| 4.2 | Performance comparison of the direct and bridge-based translation systems for both <i>Persian-Spanish</i> and <i>Spanish-Persian</i> tasks. | 87 |
| 4.3 | Comparison of interpolation bilingual texts methods performance for both <i>Persian-Spanish</i> and <i>Spanish-Persian</i> SMT tasks according to BLEU score. | 88 |
| 4.4 | Comparison of combination phrase-tables approaches performance for both <i>Persian-Spanish</i> and <i>Spanish-Persian</i> SMT tasks according to BLEU score. | 89 |
| 4.5 | The performance learning curve of the bridging systems. | 93 |
| 4.6 | The learning curve of the bridging systems comparing the performance of direct, triangulation, and interpolated systems. | 94 |

| | | |
|-----|--|-----|
| 4.7 | The performance of direct, triangulation, and interpolated models versus all types of direct-bridge combination proposed method. | 95 |
| 5.1 | Bootstrapping high-level overview. | 102 |
| 5.2 | Performance comparison of the translation systems. | 122 |
| 5.3 | Learning curve of the translation systems performance. | 123 |

List of Tables

| | | |
|------|--|----|
| 3.1 | Open-Subtitle corpus statistics. | 58 |
| 3.2 | Tanzil corpus statistics. | 58 |
| 3.3 | The performance comparison of Classical (Moses) and Hierarchical (Cdec) translation models for Spanish-English translation task according to different 4-gram language models on Open-Subtitles parallel corpus using BLEU scores. | 59 |
| 3.4 | The performance comparison of Classical (Moses) and Hierarchical (Cdec) translation models for English-Spanish translation task according to different 4-gram language models on Open-Subtitles parallel corpus using BLEU scores. | 60 |
| 3.5 | The performance comparison of Classical (Moses) and Hierarchical (Cdec) translation models for English-Persian translation task according to different 4-gram language models on Open-Subtitles parallel corpus using BLEU scores. | 61 |
| 3.6 | The performance comparison of Classical (Moses) and Hierarchical (Cdec) translation models for Persian-English translation task according to different 4-gram language models on Open-Subtitles parallel corpus using BLEU scores. | 61 |
| 3.7 | The performance comparison of Classical (Moses) and Hierarchical (Cdec) translation models for Persian-Spanish translation task according to different 4-gram language models on Open-Subtitles parallel corpus using BLEU scores. | 62 |
| 3.8 | The performance comparison of Classical (Moses) and Hierarchical (Cdec) translation models for Spanish-Persian translation task according to different 4-gram language models on Open-Subtitles parallel corpus using BLEU scores. | 62 |
| 3.9 | The performance comparison of Classical (Moses) and Hierarchical (Cdec) translation models for Persian-Spanish translation task according to different 4-gram language models on Tanzil parallel corpus using BLEU scores. | 63 |
| 3.10 | The performance comparison of Classical (Moses) and Hierarchical (Cdec) translation models for Spanish-Persian translation task according to different 4-gram language models on Tanzil parallel corpus using BLEU scores. | 64 |
| 3.11 | The performance of applying 3-gram and 5-gram language models through Ken, SRI, and IRST tool-kits on Spanish-English translation task using BLEU metric. | 66 |
| 3.12 | The performance of applying 3-gram and 5-gram language models through Ken, SRI, and IRST tool-kits on English-Spanish translation task using BLEU metric. | 67 |

| | | |
|------|--|-----|
| 3.13 | The performance of applying 3-gram and 5-gram language models through Ken, SRI, and IRST tool-kits on English-Persian translation task using BLEU metric. | 68 |
| 3.14 | The performance of applying 3-gram and 5-gram language models through Ken, SRI, and IRST tool-kits on Persian-English translation task using BLEU metric. | 69 |
| 3.15 | The performance of applying 3-gram and 5-gram language models through Ken, SRI, and IRST tool-kits on Persian-Spanish translation task using BLEU metric. | 70 |
| 3.16 | The performance of applying 3-gram and 5-gram language models through Ken, SRI, and IRST tool-kits on Spanish-Persian translation task using BLEU metric. | 71 |
| 4.1 | Corpus statistics including the source and target languages information. | 85 |
| 4.2 | The BLEU scores comparing the performance of direct translation with bridge-based translation for <i>Persian-Spanish</i> SMT system and back translation through <i>English</i> as bridge language. | 86 |
| 4.3 | The BLEU scores comparing the performance of different interpolating bilingual texts improvements for <i>Persian-Spanish</i> and <i>Spanish-Persian</i> SMT systems through <i>English</i> as bridge language. | 88 |
| 4.4 | The BLEU scores comparing the performance of different interpolating bilingual texts improvements for <i>Persian-Spanish</i> and <i>Spanish-Persian</i> SMT systems through <i>English</i> as bridge language. | 89 |
| 4.5 | Phrase pairs categorization of the portions extracted from the bridge phrase-table. | 91 |
| 4.6 | Transfer method versus triangulation with different filtering thresholds (100/1,000/10,000). | 93 |
| 4.7 | Baseline combination experiments between best bridge baseline and best direct model. | 94 |
| 4.8 | ODBC experiments results. | 95 |
| 4.9 | Percentage of phrase pairs extracted from the original bridge phrase-table for each bridging category. | 96 |
| 5.1 | Translation results using BLEU for Spanish-English and back translation tasks. | 120 |
| 5.2 | Comparing the baseline-SMT and the enlarged-SMT systems using BLEU for Spanish-English (and back translation) and Persian-Spanish (and back translation) tasks. | 121 |

List of Algorithms

| | | |
|---|--|-----|
| 1 | Semi-supervised learning for statistical machine translation | 26 |
| 2 | Active learning for statistical machine translation | 30 |
| 3 | Building phrase-based translation systems | 42 |
| 4 | Triangulation technique | 80 |
| 5 | General bootstrapping | 105 |
| 6 | Self-training | 107 |
| 7 | Co-training | 110 |
| 8 | Round-trip training | 116 |
| 9 | Round-trip training optimization | 117 |

To my family

Chapter 1

Introduction

This chapter presents the context and the motivations that brought us to develop this thesis, and the proposed objectives to be achieved. Section 1.1 sets the context and the motivations of this thesis, Section 1.2 defines the thesis objectives, and finally, Section 1.3 gives an outline of the rest of this thesis manuscript.

1.1 Context and Motivation

Language is a mean of communication. Human beings exchange information between two or more parties using natural languages only. The prime objective of communication is to share information and request or impart knowledge. The information can be specified in written-form or vocal-form (spoken). The most important thing in information content form is the validity of sentences in the given language.

Morphemes, phonemes, words, phrases, clauses, sentences, vocabulary and grammar are the building blocks of any natural language. All valid sentences of a language must follow the rules of that language (grammar). Invalid sentences are not worth and won't be effective to share knowledge, hence out-rightly rejected.

Any natural language consists of countably infinite sentences and these sentences follow basic structure. A sentence structure is perceived hierarchically at different levels of abstraction, i.e. surface level(at the word level), POS(part-of-speech) level to abstract level (phrases: subject, object, verb etc.). The sentence formation strictly depends on the syntactically permissible structures coded in the language grammar rules.

The basic sentence structures broadly depend on the positions of Subject, Object, Verb i.e. their permutations, accordingly *SVO*, *SOV*, *OSV*, *OVS*, *VSO*, *VOS* are possible, but not all (*OVS*, *OSV*) are followed in the grammar of natural languages of the world. These are all referred as word order. Depending the internal phrasal structure of phrases especially the verb phrases, certain clauses, sentences are broadly classified as simple, complex and compound sentence.

Being an effective medium of communication, language speaks for the human mind ideas and expressions obviously. There exists several languages in the world reflecting the linguistic diversity. Undoubtedly, knowing and understanding all the languages of the world would be an absolutely difficult task for an individual. Interesting and unique linguistic challenges are due to studying languages with insufficient resources.

We can get closer towards the goal of an universal translator through providing a solution for these challenges. While there are many languages spoken around the

world, each language does not sit in isolation. Either synchronically or diachronically, languages are often connected with other languages. Accordingly, the methodology of translation was revised in order to communicate the messages from one language to another.

Translation is considered as one of prerequisites of the today's fast world. To meet this requirement in a more rapid pace, the human being thought of automatic translation of one language to another, and several tools, free as well as proprietary, are now available which support translation of text into one or more languages.

In today's era of technology, language engineering focuses on modelling of human languages under Computational Linguistic (CL) research domain. CL is an interdisciplinary field of computer science and linguistics has collaboration with Artificial Intelligence (AI) area, and is concerned with computational aspects of human natural language. Computational linguistics is categorized into applied and theoretical components. Theoretical linguistic deals with linguistic knowledge needed for generation and understanding of language.

CL functions analogous to computational biology or any other type of computational aspect to answering the scientific questions of linguistics. The core questions in linguistics, it develops computational methods involving the nature of linguistic representations and linguistic knowledge, and linguistic knowledge acquisition and adaptation in language production and comprehension.

Answering these questions end in the human language ability description and is likely to help to explain the distribution of linguistic data and behaviour that we actually observe. Usually, we propose formal replies to these core questions, in computational linguistics. Linguists are really asking what and how humans are computing. So we mathematically define classes of linguistic representations and formal grammars which look sufficient for capturing the range of phenomena in human languages. We study their mathematical properties, and devise efficient algorithms for learning, production, and comprehension. Due to the fact that the algorithms are actually able to run, we have the chance to test our models and ascertain whether they make appropriate predictions.

Beyond this core question, linguistics also acknowledge a variety of questions like sociolinguistics, historical linguistics, psycholinguistics, and neurolinguistics. These scientific questions are fair game as well for computational linguists, who are likely to use models and algorithms to make sense of the data. In this case, we do not aim to model the everyday speakers competence in their native language, but rather to automate the special kind of reasoning that linguists do, potentially providing us the opportunity to work on bigger datasets (or even new kinds of data) and draw more solid conclusions. Similarly, it is possible that the computational linguists design software tools to help document endangered languages.

Natural Language Processing (NLP) is a branch of Artificial Intelligence (AI) and theory-motivated range of computational techniques for the automatic analysis and representation of human language. In other words, NLP is the art of solving engineering problems which require to generate or analyse natural language texts. Here, the success is not measured by the fact that whether we designed a better scientific theory or proved that languages X and Y were historically related. Rather, it is analysed according to the fact that whether we provide good solutions on the engineering problem.

For instance, we do not judge *Google Translate* on whether it captures what translation *truly is* or explains how human translators perform their profession. We judge

it on whether it produces reasonably proper and fluent translations for people who need to translate certain things in practice.

The automatic translation community has ways of measuring this, and their main focus is on improving those scores. It is generally believed that NLP is mainly used to help people to navigate and digest large quantities of information which already exist in text form. It is also used to produce improved user interfaces so that humans can communicate more appropriately with computers and other humans as well. By saying that, NLP is engineering, and may be used for scientific ends within other academic disciplines such as political science, economics, medicine, digital humanities, etc.

Machine Translation (MT) known as a very vital area of computational linguistics, and one of the earliest areas of research in NLP, is the process of automatically translating written text or speech in one natural language (source) into another natural language (target). MT is an extremely complicated task with numerous unsettled difficulties. There are several reasons which combine to make high-quality automatic machine translation extremely challenging such as differences in lexical choice, word order and grammatical structure, the use of idiomatic expressions and non-literal translations, and the presence or absence of particular cultural conventions.

On a basic level, MT performs simple substitution of words in one language for words in another, while this fact solely is usually unable to produce a satisfactory translation of a text, since is required to recognise the whole phrases and their closest counterparts in the target language. Solving this problem with statistical techniques is a rapidly growing field which definitely leads to more appropriate translations, handling differences in linguistic typology, translation of idioms, and the isolation of anomalies. There exist different approaches to address the problem of MT. We will now give a rough overview over these different methodologies;

1. The rule-based approach: In rule-based systems, the source language text is analysed and transformed into intermediary representation. The target language text is achieved from this representation. Human experts have devised the rules. For the reason that a large number of rules is needed in order to capture the phenomena of natural language, this is considered a time consuming process. As the set of rules grows over time, it gets more and more perplexing to extend it and ensure consistency (Ahsan et al., 2010).
2. The data-driven approach: In this type, bilingual and monolingual text are used as main knowledge source. Often, a further division is made between the example-based approach (where the basic idea is to do translation by analogy), and the statistical approach¹.

Translation is treated as a machine learning problem by Statistical Machine Translation (SMT). That is to say that a learning algorithm is applied to a large body of previously translated text, known variously as a parallel corpus, parallel text, bilingual-text, or multilingual-text. Afterwards, the learner is able to translate previously unseen sentences.

We can build an MT system for a new language pair within a very short period of time by the aid of an SMT tool-kit and enough parallel text. That is to say that the basic idea in SMT is that we can learn to translate from a corpus of translated text

¹In this thesis, we will follow the statistical approach to machine translation.

through taking a look at translation frequencies. If a word or sentence in one language is consistently paired with the same word or sentence in the other language, this indicates that they are appropriate translations of each other.

Since SMT aims at translating a source language sequence into a target language sequence through maximizing the posterior probability of the target sequence given the source sequence, in state-of-the-art translation systems, this posterior probability is usually considered as a several different models consolidation, such as translation models and lexical models for both translation directions, target language model, word and phrase penalties, etc. A bilingual parallel text represent the probabilities which express correspondences between the words in the source language and the words in the target language and a monolingual text in the target language depicts language model probabilities.

Most of the recent research in the area of SMT has been focused on modelling translation depending on phrases in both the source language, and matching them with their statistically-determined equivalents in the target language (phrase-based translation). This translation approaches said to be used by many modern successful translation machines. Determining a translation model from a word-aligned parallel corpus is regarded as a significantly critical task in a Phrase-based Machine Translation (PBMT) systems.

Frequently, enhance a translation system performance is improved by the larger available training corpus. Whereas the task of finding appropriate monolingual text for the language model is not granted complicated, acquisition of a large high-quality bilingual parallel text for the desired domain and language pair requires a lot of time and effort, and it is highly unlikely for some language pairs. In addition, small corpora represent certain advantages; the possibility of corpus manual creation, possible manual corrections of automatically collected corpus, low-memory and time requirements for the training of a translation system, etc. Accordingly, the strategies for exploiting limited amounts of bilingual data are more and more on the center of attention.

For several NLP tasks such as SMT which rely on the data-driven approach, parallel corpora are proved to be significant resource. The lack of parallel data seems to be especially problematic for the SMT systems, in as much as they require a considerable amount of training dataset for producing reliable models.

The translation performance of the SMT systems directly depends on not only the quantity but also the quality of the available parallel data. Unfortunately, parallel corpora are not readily available in desired quantities. These corpora are limited in quantity, genre, and language coverage as a result of the special effort required to create them, which is time consuming and costly. Large parallel corpora are said to be only available for a handful of language pairs such as *Spanish-English*, *Chinese-English*, *Arabic-English*, etc. The majority of this data comes from parliamentary proceedings such as *European Parliament* (Europarl) or the *United Nations* (UN), and a limited amount of news-wire text is also available. For a vast majority of other language pairs, there is a severe dearth of publicly available parallel corpora. On the other hand, there are a large number of languages that are considered low-density, either because the population speaking the language is not very large, or even if millions of people speak the language, insufficient online resources are available in that language.

So, having a machine learn how to translate from a source language to a target language, is one of the holy grails of AI community. However, the lack of bilingual training data is not just specific to low-density language pairs. It may also happen

because of change in the domains of training and test data. For instance, consider a case where we have bilingual training sentences from the *News* domain, and we want to build an SMT system to translate text from the *Economics* domain. A statistical translation system can be improved and adapted by incorporating new training data in the form of parallel text in cases where there is lack of bilingual training data.

This thesis focuses on the bottleneck of data scarcity relevant to training the SMT systems in the case of small-size bilingual texts available between the source and target languages. In simple words we propose the methods to overcome the shortage training parallel corpus limitation that is usually experienced during the training of minimal parallel-resource SMT systems. This proposal is put forward by us so that we could contribute in the improvement of the translation systems' performance that has been mentioned in the thesis. This will help us in achieving the high-quality translation output in comparison with the baseline ones.

We have proposed two interesting scenarios in this thesis so that we can treat the lack of bilingual training dataset for minimal parallel-resource SMT task;

- The first scenario is the *Direct-Bridge Combination*. This approach is mainly based on the bridge language technique for minimal parallel-resource SMT systems. This approach has been proposed so that a direct and a bridge model built from a given parallel corpora to achieve better coverage and overall translation quality, could be effectively combined. During the application of this technique, we have chosen the portions of bridge model and maximized the information gained from them. The portions we chosen do not interfere with trusted direct model in any way. It can simply be said that the main goal of this scenario is to smartly combine the direct and bridge models so that information gain could be maximized. In order to fulfil our goal, we have pondered over the idea of categorizing the bridge phrase pairs into different categories. These categories are based either on the existence of source or/and of the target phrases in the direct model. We have also demonstrated that optimized direct-bridge combination can result in a large reduction of the bridge model without incurring any sort of effect on the performance. There are also some cases where it has improved the performance. We have further analysed and answered the question that "how by doing a smart choice of only relevant portion of the bridge phrase-table, the quality could be improved?".
- The second scenario is the *Round-Trip Training*, which is based on the retraining technique for minimal parallel-resource SMT systems. This approach has been proposed so that a learning framework could be developed, which is strong enough to compel the minimal-resource SMT system to automatically learn from unlabelled data through a round-trip communication game. In this technique we use two independent translation systems so that model could be represented for either both outbound-trip and inbound-trip translation tasks. Later on, they are asked to learn from each other with the help of a round-trip learning process. The greatest benefit of using this strategy is that these two translation tasks can merge into a closed loop, and generate informative feedback signals. So that it could be easy for them to provide training to the translation model without having a human labeller involved. The main idea behind this scenario is to seek advantage of the readily available generated text for the purpose of building a high-quality SMT model in an iterative manner. Apart from this main idea, the mentioned scenario also helps in making a selection of the high-quality subset of the generated sentences in target-side, it

provides assistance in the preparation of their high-quality correspond source sentences, and it keeps them paired so that they could be used together along with the original initial small bilingual text for the purpose of retraining the SMT model. The most prominent points behind this proposed scenario is the urge to obtain high-quality while using the fewer training dataset. However, this could only happen if it is allowed to select the data from which it learns.

As a result of the research conducted in this thesis, we have ended up gaining the fully automatic methods so that SMT systems could be improved for the situation where the size of bilingual training data is small.

1.2 Thesis Objectives

After analysing the context and identifying the main issues, we can define the thesis objectives. The main goal of this thesis is treating the training data scarcity for minimal-resource language pairs through SMT model to improve translation quality. From this global contribution, several objectives can be derived as follows:

- Analysing the performance of the Classical and Hierarchical as two phrase-based translation models for SMT on *Spanish*↔*English* translation task as well as *English*↔*Persian* and *Persian*↔*Spanish* translation tasks, and investigating the impact of different statistical language models on both Classical and Hierarchical translation models by applying different *n-gram* language models. We first implement *Moses* and *Cdec* as baseline translation systems, then compare the translation systems' outputs under equal conditions, after that we apply *Ken* language model as well as *SRI* and *IRST* language models on each translation task in order to investigate their effects on the output results, finally we identify the suitable translation model in each translation direction. The comparative performance between our case-study language pairs based on Classical and Hierarchical phrase-based translation models will be conducted to set as state-of-the-art for further researches on phrase-based SMT.
- Providing a direct-bridge combination model by developing a smart approach to combine bridge and direct translation systems based on bridge language technique. We first investigate the performance of bridge language strategies, and based on the given conclusion extracted from this in-depth investigation, we propose the direct-bridge combination scenario to maximize the information gain by choosing the relevant portions of the bridge-based model that do not interfere with the direct-based model which is trusted more indeed. Bilingual text interpolation and phrase-table combination as recent improvements on bridge language technique to enhance the minimal parallel-resource SMT systems' performance are presented as well.
- Proposing round-trip training scenario by developing a learning framework based on retraining approach. First, we analyse self-training as well as co-training to show how these weakly supervised learning algorithms can improve translation systems' performance, and how they affect on our proposed scenario as well. Then, we explore how the proposed round-trip training scenario can automatically learn from unlabelled data through a training communication game. Finally, we implement baseline translation system and improve it during each step of our round-trip training game. As a result, we will have obtained the optimised and high-quality translation outputs.

We are interested in getting a high-quality SMT system with competitive performance, without the requirement of large-size training dataset especially for minority languages.

1.3 Thesis Outline

In addition to this introductory chapter, this thesis is structured into six more chapters:

- **Chapter 2** introduces relevant theoretical and mathematical background from the machine translation and the statistical machine translation literature. From the machine translation perspective, this chapter gives an introduction to machine translation approaches, benefits, and limitation issues. For statistical machine translation, this chapter provides background on "how the models' feature functions and parameters are learned?", "how the models' output is evaluated?", and "which tools and software packages are going to be used in the experiments and evaluation process?". Furthermore, this chapter presents a review of the state-of-the-art in minimal parallel-resource statistical machine translation. The basic concepts and modules related to minimal-resource statistical machine translation are reviewed. Then more recent research lines are examined. Putting a special attention on the two main topics of this thesis; dealing with learning frameworks as well as pivoting (bridging) framework.
- **Chapter 3** provides a detailed comparison between the performance of Classical and Hierarchical phrase-based translation models in statistical machine translation systems, through *Moses* and *Cdec* open-source translation systems under the same conditions. During the experimental framework three language pairs are evaluated; *Spanish*↔*English* as well as *English*↔*Persian* and *Persian*↔*Spanish*. For the purpose of detailed comparison we apply various *n-gram* statistical language models on each translation direction, and the comparative results and performances determine the best possibility for each translation direction.
- **Chapter 4** as one of the contributions of this thesis, addresses general framework of bridge (pivot) language technique as a common solution to the lack of parallel data. In this chapter we discuss various strategies of bridge language technique as well as proposed improvements to improve the bridge language technique performance. Also we propose a combination technique of direct and bridge statistical machine translation models to enhance translation quality. Our experimental results show that our proposed combination scenario can lead to a large reduction of the bridge model without affecting the performance if not enhancing it.
- **Chapter 5** presents the main contribution of this thesis. This chapter explores the use of monolingual data for training the translation system in order to improve translation quality by proposing the round-trip training scenario. First, an in-depth analysis of bootstrapping methods is conducted; we investigate the behaviour of self-training mechanism as well as co-training mechanism, and link them to our proposed scenario. Then, the proposed optimized round-trip training theory, mechanism, and algorithms are provided, respectively. Finally, detailed experimental evaluations on the *Spanish*↔*English* (as a well

parallel-resource language pair), and *Persian*↔*Spanish* (as a low parallel-resource language pair) translation tasks are provided.

- **Chapter 6** summarises the major contributions and results of this thesis and proposes research lines for future work.

Chapter 2

Background

In this chapter, we introduce relevant technical background from the Statistical Machine Translation (SMT) literature. From the Artificial Intelligence (AI), Computational Linguistics (CL), and Natural Language Processing (NLP) perspective, this chapter mainly introduces variant approaches on Machine Translation (MT), and implements background on how the model's feature functions and parameters are acquired, how the model's output is assessed, and which tools and software packages are going to be used in the experiments of this thesis.

2.1 Computational Linguistics

A new branch of applied linguistics arose, namely the computational, or engineering, linguistics in the latest half of the twentieth century when the booming availability of machine-readable corpora has represented new methods for studies in different areas such as lexical knowledge acquisition, grammar construction, and machine translation. Although, it was very common in the speech community, statistical and probabilistic methods used to discover and organize data were relatively new to the field at large. Therefore, various actions were undertaken to locate and collect machine-readable corpora in order to recognize the potential use of this data and also to work toward making these materials accessible for the research community.

Intelligent NLP is based on the science called computational linguistics which is relatively connected with applied linguistics and linguistics in general. CL which is broad field incorporating research and techniques for processing language, can be considered as an equivalence of automatic processing of natural language since the main task of computational linguistics is just the construction of computer programs to process words and texts in natural language.

2.2 Natural Language Processing

The field NLP targets on the interactions between human language and computers. It sits at the intersection of computer science, artificial intelligence, and computational linguistics. NLP works as a method for computers helping them to analyse, understand, and derive meaning from human language in a smart and useful way. Through exploiting NLP, developers are able to organize and structure knowledge to perform different tasks such as automatic summarisation, translation, named entity recognition, relationship extraction, sentiment analysis, speech recognition, and topic segmentation.

In other words, NLP is an area of research and application that seeks to discover that in what way the use of computers can help in understanding and manipulating natural language text or speech so useful things could be performed. The basic aim

of the NLP researchers is to gather knowledge on how human beings understand and use language so that appropriate tools and techniques can be developed. These techniques and tools will then be used for the purpose of making the computer systems understand and manipulate natural languages so that they could perform the desired tasks. We can easily find the NLP foundations in a number of disciplines such as computer and information sciences, linguistics, mathematics, electrical and electronic engineering, artificial intelligence and robotics, psychology, etc. However, in order to make the application of NLP we must be acquainted with a number of fields of studies including the machine translation, natural language text processing and summarization, user interfaces, multilingual and cross language information retrieval, speech recognition, artificial intelligence and expert systems, and so on.

In order to analyse a text, or to allow machines to understand the human's speak, NLP is the best choice. This type of interaction which is considered as a human-computer one permits real-world applications like automatic text summarisation, sentiment analysis, topic extraction, named entity recognition, parts-of-speech tagging, relationship extraction, stemming, and more. It is widely common to use NLP for text mining, machine translation, and automated question answering. Besides, NLP is known to be a solid dispute in computer science. Human language is rarely precise, or plainly spoken. To understand human language is to understand not only the words, but also the concepts and how they are associated together to create meaning. Despite the fact that language is regarded as one of the simplest areas for humans to learn, its ambiguity is what makes natural language processing a challenging dispute for computers to master.

It is estimated that, currently there exist around 7000 languages spoken in the world (Nettle, 1998), while NLP research focuses on only a small number of those languages such as *English, Chinese, French, German*, etc. Those Languages which have received relatively less attention from NLP are usually less popular, due to their lack of available resources, and are often called low-resource. Languages owning abundance of NLP resources and tools usually do so as a result of political, financial, and social reasons. Though, economically speaking, focusing money and effort on the most widely spoken languages makes sense, it is difficult for researchers to produce significant resources for current low-resource languages without funding. The availability of NLP tools, such as machine translation (MT), may encourage speakers of low-resource languages to continue to use that language rather than abandon it in favor of a majority language. From a commercial, military and political point of view, MT has widespread applications. For example, increasingly, the Web is accessed by *non-English* speakers, reading *non-English* pages. Our language-speaking capabilities is not to bind our ability to find relevant information clearly. Furthermore, there may not exist sufficient linguists in some language of interest to cope with the sheer volume of documents to be translated. Enter automatic translation. MT postures a number of interesting machine learning challenges such as the following:

- The associated models are typically very large, as are the data sets;
- The applied training material is often noisy and plagued with sparse statistics;
- The search space of possible translations is sufficiently large that exhaustive search is not possible.

Advances in machine learning, such as maximum-margin methods, periodically emerge in translation research.

2.3 Minimal-Resource Languages

The documentation and description of the language are basically the tasks that are focused on the collection and analysis of language samples. After collection, these samples are analysed so that properties of the language could be described properly. Generally speaking, these two tasks right in the process of a slow transition to electronic digital storage from older physical storage methods. This is understood to be the domain of linguists field. Even though limited research has been conducted so far on the necessary methods, NLP and CL are considered in a great position for potential collaboration on these tasks.

NLP methods that have been created for the purpose of minimal-resource languages are likely to encounter the similar issues that are faced by the documentary and descriptive linguists working in the minority languages field. It is considered to be highly informational to lay out these issues for the NLP researchers so that they could learn what to expect while they are dealing with these types of languages.

The first and one of the biggest issues with minimal-resource languages is that obtaining the resources is an extremely difficult task. Most of the available language description is present in either paper format or in unpublished form. Whatever little amount that does exist in the electronic format is either in useless or unusual format (Bird and Simons, 2003). Due to which, obtaining and using even the raw text becomes impossible in a minimal-resource language.

The second issue is that standardization of the orthographies for minimal-resource languages is not confirmed. There are certain points that indicate towards this weakness for instance, the word boundaries might not be standardized, spellings could be different, and even the language usage itself might not be consistent where multiple speakers are involved. Although there are various languages that include a specific part of the dictionary or certain word lists, still we must keep it in mind that most of these dictionaries and word lists have been created by the foreign linguists so they do not represent a standardized spelling.

The third issue is that even though the link between the dialects and their relationships is well-understood in relation to the most prominent languages of the world, they are not clear while dealing with less studied languages. The important point is that mutual intelligibility is mostly used as a measuring stick so the link between two languages could be determined (whether separate languages or just the separate dialects), even this step can be considered subjective up to a certain extent.

As a result of the above mentioned issues, it can be understood that we can easily detect substantial differences in two different sources of text belonging to the same language. These differences can put NLP researchers into confusion and they could question themselves whether their tools are even applicable to the presented texts or not.

The greatest benefit of the text-based methods is basically the amount of written text available even in minimal-resource languages; still it must be kept in mind that most of the world's languages are available in unwritten form. Linguists tend to benefit from some of the languages that possess the writing system but at the same time, the literacy rates of native speakers remains low. Thus, we can say that using any text-based approach would not be of any help as it will be fundamentally limited in its scope.

All the work done on NLP for minimal-resource languages till now, can be distributed into two major categories:

1. The approaches that emphasize on a small set of languages.
2. The approaches that are applied to a large set of languages.

During the first approach, emphasis is put on a single language or a small set of related languages. The application of this approach starts with a data collection phase at which point the text or speech are compiled into the languages of interest. An NLP tool is also produced during the entire procedure. The benefit of these approaches is that they do give positive outcome, but the biggest issue is that they require aid from an expert. Also they are not immediately applicable to the other languages.

If we search for the most effective example of an approach that is focused on a very large set of languages then it could easily be found inside the *Kevin Scannell Crubadan* project (Scannell, 2007). After they crafted the Web search queries that were designed for the purpose of returning Web-pages in specific minimal-resource languages, they succeeded in building corpora for (1872) different languages. A number of different tools and resources for minimal-resource languages were significantly developed by these corpora such as thesauri, diacritic restoration, and automatic translation. However, there is one weakness that we can easily find in the Scannell's approach and that is it heavily relies on the manual effort of expert volunteers. Without getting support from this manual effort, these resources and tools cannot be created.

Some of the other examples for the many language approach include:

- **The proposed human language project:** This project provides a complete description of the common format that is used for the purpose of annotated text corpora. Also, it issues the challenge for creation of the universal corpus that must contain all of the world's languages (Abney and Bird, 2010).
- **The Leipzig corpora collection:** This project has created the corpora for (124) distinct languages. Further it offers statistics about each of these languages for instance, word frequencies and contexts as well as dictionaries even though the texts can't be distributed due to copyright (Biemann et al., 2007).

2.4 Machine Translation

Machine translation (MT) may be defined as an example of a computer use to translate a text from one natural language (the source language) into another one (the target language). MT is regarded difficult mainly for the reason that natural languages are proved to be highly ambiguous and also since two languages are not always permitted to express the same content. MT carries plentiful advantages over traditional professional human translation. Ordinarily, it is very easy to apply MT systems. Since translation is performed quickly and usually on-demand, it is much more convenient to use this method rather than a human translator. On top of this, the cost factor should also be regarded; professional human translation is usually costly, not available on-demand, and when it is, requires much more time to complete.

General MT systems drawbacks include the fact that translation output (to some degree) is usually lacking in accuracy, especially when focused on a particular domain. Translations made for technical or scientific writings are frequently inaccurate, unless the system has been trained on how to handle the data from that particular domain.

The MT accuracy is not guaranteed. If a poor quality translation is generated, there is generally no real way of recognizing, unless someone speaking both source and target languages compares and evaluates them. This fact can postures some problems, particularly when translating sensitive or private documents.

Spoken language sentences seem to be long and complex, and often consist of grammatically unpredictable constructions. They may even hold 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 in order to find the unknown words meanings in context is said to be also a denouncing problem in both text and speech translation. Many words present various meanings and, consequently, different possible translations. In some languages such as *Japanese* or *Korean*, not even the word boundaries are given. That is to say that certain grammatical relations in one language might not exist in another language, and sentences involving these relations are to be significantly reformulated. Furthermore, 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.

For accurately performing MT, various 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. Linguistically speaking, various types of dependencies are to be considered; morphologic, syntactic, semantic and pragmatic dependencies (Jurafsky and Martin, 2000).

More specifically, there are dependencies relating source and target language words, describing that certain words or phrases have the capability of being translated to 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 affords to deal with the weak and vague dependencies. Having acquired such a framework, certain methods efficiently obtaining the large amount of relevant dependencies must be developed (Och and Ney, 2002).

Large-scale NLP demands the vast amounts of lexical, grammatical, and conceptual knowledge integration. A robust generator is expected to operate well, even when some of knowledge pieces are missing; it must also be resistant to incomplete or inaccurate input. There exist two basic issues which machines encounter while dealing with natural languages. The first one is related to context and cultural issues; Computers, definitely, are unable to perceive the contextual and pragmatic information which humans can. Similarly, they are unaware of cultural differences which often surface in linguistic exchanges. The second issue relates to the language function. Conveying the meaning is just one of the application of human language; there are many others as well, 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 even difficult to translate for humans. It follows that computers encounter great difficulty while providing quality translations for the so-called pieces. Ambiguity, idioms, differences in vocabulary, collocations, and structural and lexical differences between the source and target languages are also difficulties which an MT system has to deal with (Gross, 1992).

We can identify the kinds of linguistic errors that might be expected in the raw output yielded by fully automated machine translation 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 making a change or subtracting from the intended meaning.

Despite some of the negative aspects of machine translation, they also retain diverse qualities that make machine translation very attractive. Machines are usually constant both in interpretation and vocabulary; they do not omit words or paragraphs accidentally, and do not make the erroneous conclusions that can be made even by competent human translators. According to (Gross, 1992), comparing to human translators, machines, for the most part, have potential to be faster, more economical, and provide translations with a greater degree of accuracy. This is particularly true if the machines are limited to a specific subject domain, just as human translators are. People's overall mistrust and uncertainty on computers and technological advances may, he states, be the major reason behind their scepticism towards machine translation, rather than criticizing MT legitimately.

The different ways in which the MT problem has been approached may be classified according to the nature of the knowledge used in the development of the MT system. From this point of view, one can distinguish between rule-based and corpus-based approaches; although hybrid approaches are possible between them too.

2.4.1 Rule-Based Machine Translation

This approach applies knowledge in the form of rules explicitly coded by human experts trying to describe the process of translation. In a Rule-Based Machine Translation (RBMT) system, first the original text is analysed morphologically and syntactically for the sake of obtaining a syntactic representation. The so-called representation can then be refined to a more abstract level, emphasizing on the relevant parts of translation and ignoring other types of information. The transfer process then converts this final representation to a representation of the same level of abstraction in the target language. An RBMT system may be regarded as either an *Interlingua* or a *Transfer-Based* MT system.

One of the more classic approaches to machine translation is considered to be interlingua MT. In this approach, the source language is transformed into an interlingua, that is, an abstract language-independent representation. The target language is then developed from the interlingua.

In the transfer-based approach, the source language is transformed into an abstract, less language-specific representation. Those linguistic rules 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 developed.

Throughout the development, rule-based MT systems are easier to diagnose and the translation errors produced by them have a repetitive nature, making them more predictable and easier to post-edit, and consequently, better suited for dissemination purposes. Within the rule-based MT paradigm, the interlingua approach can be an alternative to the *Direct*¹ and transfer approaches (Leavitt et al., 1994).

2.4.2 Corpus-Based Machine Translation

Corpus-Based Machine Translation (CBMT) approaches use large collections of parallel texts (corpus) as the source of knowledge through which the engine learns how to perform translations. Three basic types of corpus-based approaches to the MT problem have been set;

1. Example-Based Machine Translation (EBMT).
2. Statistical Machine Translation (SMT).
3. Neural Machine Translation (NMT).

In EBMT (Nirenburg et al., 1994) the translation is performed by analogy; given a source sentence and a parallel corpus. EBMT aims to find the best match for the source sentence in the parallel corpus and retrieves its target part as the translation.

In SMT (Lopez, 2008) translations are developed on the basis of statistical translation models whose parameters are learned from parallel corpora.

In NMT (Bahdanau et al., 2014) the key role is played by the neural networks while developing the translation.

Corpus-based MT systems require large amounts, in the order of millions of sentences, of parallel corpora to come to a reasonable translation quality in open-domain tasks. Basically, such a vast amount of parallel corpora is not available for most minimal parallel-resource language pairs demanding MT services such as *Persian-Spanish*.

2.4.3 Hybrid Machine Translation

This approach integrates more than one MT paradigm are receiving increasing attention. The *METIS-II* MT system (Dirix et al., 2005) is an example of hybridization around the EBMT framework; it avoids the usual need for parallel corpora via applying a bilingual dictionary and a monolingual corpus in the target language.

An example of hybridization around the RBMT paradigm is given by Oepen et al. (2007); they integrate statistical methods within an RBMT system to choose the best translation from a set of competing translations developed applying rule-based methods.

In SMT, Koehn et al. (2003) integrates additional annotations at the word-level into the translation models in order to better learn some aspects of the translation that are best explained on a morphological, syntactic or semantic level.

In this issue of step-by-step thesis, we focus on the statistical machine translation, and we explain how the SMT works.

¹In the direct approach words are translated directly without passing through an additional representation.

2.5 Statistical Machine Translation

In recent years, MT has been dominated by statistical approaches, which aim to learn such segmentation or tokenization, translation, and recombination decisions by learning them from a large collections of previously translated texts.

The basic idea in SMT is that, we can learn to translate from a corpus of translated text by looking at translation frequencies. If a word or sentence in one language is consistently paired with the same word or sentence in the other language, this indicates that the two are good translations of each other. We formalize this expectation that frequent translations are good translations through probabilistic models.

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, 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 (Brown et al., 1990). Most other MT 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.

SMT 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 SMT, it is typically necessary to combine many knowledge sources.

In SMT, 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 MT 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 SMT 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 is required to be performed in order to implement it on other language pairs or text domains is to train it on new data. While SMT has proved to yield promising results for large amounts of language, it can be shown that rule-based systems function more efficient where the source sample is shorter.

SMT involves two individual processes known as *Training* and *Decoding*. In the training process, from an aligned parallel corpus, a statistical translation model is excerpted, and another separate statistical model is extracted from a monolingual corpus in the target language. The decoding process can be defined as the one which generates the translation. The decoder receives the input, a phrase, a sentence or sentences, which searches through all the input available translations produced by the translation model. Afterwards, the translation with the highest probability produced by the language and translation models, is nominated as the most likely precise translation, and is output in the target language.

The two important features of SMT are the Translation Model (TM) and the target Language Model (LM). These features are multiplied together. An independent modelling of the target language model and the translation model is gained via the *Noisy-Channel* model. The well-formed target language sentence is expressed via the target language model. The source language sentences and the target language sentences are connected to each other via the translation model links. The translation model score, broadly based on lexical correspondences, depicts how well the meaning of the source sentence is captured in the translation. The language model score is based on frequency of occurrence of sub-strings in a monolingual corpus of the target language, and acts independently whether the original meaning of the source language sample is captured. The score simply represents the probability of the translation being a valid sentence in the target language.

The two common approaches for language modelling are *n-gram* models, and recently *neural networks* models. There are various approaches for the translation modelling. These models are categorized as follow:

- **Word-based translation model:** The early SMT models are word-based which use words as translation units, proposed by Brown et al. (1993) in IBM. They proposed (5) models called IBM Models (1-5) (Brown et al., 1990; Brown et al., 1993). Later Och F. J. and H. Ney (2003) proposed model (6).
- **Classical phrase-based translation model:** Och (1999) first proposed to use a sequence of words (i.e. phrases) as translation units rather than single words. These models are called phrase-based models which have been shown to produce remarkably better translations (Koehn et al., 2003; Marcu and Wong, 2002; Och F. J. and H. Ney, 2004). Although phrase-based models address the problem of local reordering and idiomatic expressions in word-based models, they are unable to model long-distance reordering.
- **Hierarchical phrase-based translation model:** These models try to address the problem of complex reordering in phrase-based models by considering the structure of sentences. Chiang (2005) first proposed to use hierarchical phrases in the form of Synchronous Context-Free Grammar (SCFG) (Aho and Ullman, 1969). This approach become dominant in translation of language pairs with complex reordering.

- **Syntax-based translation model:** These models incorporate the linguistic syntax of sentences in translation. Syntax-based models can be divided into two groups: synchronous-grammar-based models and tree-transducer-based models. Many synchronous grammars have been used in MT: SCFG (Zollmann and Venugopal, 2006), synchronous tree-substitution grammars (Eisner, 2003), synchronous tree-adjoining grammars (Deneefe and Knight, 2009), and generalized multi-text grammars (GMTG) (Melamed et al., 2004). On the other hand, tree transducers have been used to create several syntax-based SMT models (Galley et al., 2006a).
- **Tree-based translation model:** This model is basically used in the case of synchronous grammar where syntactic tree is used for the purpose of providing assistance on the mapping of various linguist structure and contextual word translation. However, tree-bank (Tinsley et al., 2009) is required by the Tree-based model so it could be used as a resource for the total translation process. This is the reason that proposal was presented for less informative model such as tree to string (Liu et al., 2007) and string to tree (Neubig and Duh, 2014), or any model that is without the linguistic information including hierarchical phrase-base model. Effective planning could be done for implementing a tree-based model that includes full linguistic information in the case where rich-resource languages with comfortable tree-bank are involved. Otherwise, settling for the less informative model or any model that is available without the linguistic information for the purpose of minimal-resource languages is needed.

Generally speaking, the translation model score influences the final score twice more than that of the language model, due to the fact that it is a more significant parameter. From the combination of the translation model score and the language model score we obtain to the final score, depicting the best scores combination to give the optimum target sentence.

2.6 Parallel Corpus Alignment

Alignment is generally classified according to the level it is performed. For instance, a parallel corpus aligned on sentence level refers to the alignment of sentences. The number of phrases and words may vary between the two languages, but the sentences themselves are linguistically equivalent. Alignment on word level refers to equating words, and alignment on phrase level refers to equating phrases.

The word alignment process refers to linking words, phrases or sentences of equivalence between the two sides of a parallel corpus. Aligning a parallel corpus is required before a training model generation.

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.

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 et al. (1977) developed the Expectation Maximization (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 maximization, *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.

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 source word to single target word, and 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. In this way, single-to-multi alignments are extracted in both directions. Multiple-to-multiple alignments are extracted using phrase-alignment heuristics (Och, F. J., 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 bilingual text. The word alignment sets are refined by removing alignments occurring only on one set.

2.7 Translation Model Training

The translation model, $P(t_1^I, s_1^J)$, represents the source and target sentences probability, being linguistically equivalent, in other words, the target sentences define the source sentences meaning accurately. The translation consist of the source-target training corpus² model, and an equivalence calculating algorithm for source and target sentences.

It is required to extract a translation model from a parallel corpus involving word-aligning the data using *GIZA++* (Och F. J. and H. Ney, 2003), *MGIZA* (Gao and Vogel, 2008), or *fast-align* (Dyer et al., 2013) tool-kits, and extending those alignments to cover phrases. Phrase pairs are then extracted to give phrase lengths of (1) to n^3 words. In many cases the number of words in each aligned phrase may vary between the source and target language, depending on how each language represents the meaning of the phrase.

SMT relies on two main resources which are its parallel and monolingual corpora. In the target language, for generating a language model the monolingual corpus is applied, while the parallel corpus is needed to generate the training model,

²This corpus is aligned on sentence level.

³ n is chosen as a maximum such that the system is presented with phrases that are actually feasible to work with

which is searched by the translation model $P(t_1^I | s_1^J)$ for aligned phrases. The training data is referred to parallel and monolingual corpora collectively. Having finished the training process, the corpora themselves are no longer required for any further process.

2.8 Language Model Training

In order to determine the fluency and validity of considered target phrases or sentences the language model (LM) is applied through the decoder. By doing so, based on the language model, the probability, $P(t_1^I)$, of target phrase or sentence can be checked. The language model, extracted from the corpus, gives the frequency of sub-strings in that corpus. When the input sentence's probability is determined, it is based on the sub-strings of that sentence compared to those in the model. Different language models are defined as follows:

- **Uni-gram language model:** According to corpus word tokens, one basic language model, known as the *uni-gram* model, may be simply composed of sub-strings alone. 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, this model contains significant limitations. Since only single word types are operated by it, 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 words. On the other hand, a zero probability 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 it is highly unlikely to ensure 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 et al., 1978) are also applied. 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 to them. In this way each phrase and sentence is guaranteed a non-zero score.
- **Bi-gram language model:** A *bi-gram* LM refers to a model consisting of all *bi-grams* found in the corpus. Such language models operate based on word sequences. Probabilities are defined by determining the likelihood of the *bi-gram* second word occurring, given the first word. Throughout 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*, the probability is calculated.
- **N-gram language model** Larger models, known as *n-gram*, are based on the same logic as *bi-gram*, with *n*-length sub-strings, or *n-gram*. The *n-grams* are strings of length *n* generated from words in texts. In traditional vector space approaches, words or phrases that occur in the collection are considered as dimensions of the document space for a given collection of documents. Contrarily, in the *n-gram* approach, *n-grams* are dimensions of the document space, namely, strings of *n* consecutive characters extracted from words. Since the

number of possible strings of length n is distinctively smaller than the number of possible single words in a language, n -gram approaches, therefore, have smaller dimensionality (AleAhmad et al., 2009). 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 languages in which documents are written. The n -gram length n and the method of extracting n -grams from documents is different from one author and application to another (Mustafa, 2005). Both bi -gram or n -gram LMs operating on any string length, still encounter issues in this particular case of unknown n -grams. Generally speaking, 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, it is highly unlikely to determine whether the individual words in a previously unseen n -gram have already occurred in the training corpus even with smoothing techniques. 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, assorted n -gram models are applied, each with different weights, the scores of which are combined. In this way, it is more possible to obtain a given sentence.

- **Neural networks language model:** A neural network language model is a language model based on neural networks, exploiting their ability to learn distributed representations to reduce the impact of the curse of dimensionality. In the context of learning algorithms, when the number of input variables increases, the number of required examples can grow exponentially. In the context of language models, the problem comes from the huge number of possible sequences of words, e.g., with a sequence of (10) words taken from a vocabulary of (100,000) there are (1050) possible sequences.

A statistical language model is simply understood as the probability distribution over sequences of words. In order to understand this sequence, say its length m , a probability $P(w_1, \dots, w_m)$ is assigned to the whole sequence. It is very useful in NLP application to estimate the relative likelihood of different phrases especially those that contribute towards generating text as an output. In an n -gram model, the probability $P(w_1, \dots, w_m)$ of observing the sentence (w_1, \dots, w_m) is approximated as:

$$P(w_1, \dots, w_m) = \prod_{i=1}^m P(w_i | w_1, \dots, w_{i-1}) \approx \prod_{i=1}^m P(w_i | w_{i-(n-1)}, \dots, w_{i-1}) \quad (2.1)$$

At this point, we must make the assumption that the probability of observing the i^{th} word w_i in the context history of the preceding $(i - 1)$ words can only be estimated by the probability where it is observed in the shortened context history of the preceding $(n - 1)$ words. The n -gram model frequency counts must be calculated with the help of conditional probability:

$$P(w_i | w_{i-(n-1)}, \dots, w_{i-1}) = \frac{\text{Count}(w_{i-(n-1)}, w_{i-1}, w_i)}{\text{Count}(w_{i-(n-1)}, \dots, w_{i-1})} \quad (2.2)$$

The n -gram language models with $(n = 2)$ and $(n = 3)$ are respectively denoted by the words bi -gram and tri -gram language models.

2.9 Decoding

In SMT, decoding as a search process, is the task of finding the best translation for a given source sentence from all possible translation according to the translation model. In other words, 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 sub-strings are matched with translation model sub-strings, each individual translation is retrieved, and those translations are concatenated to produce the full translation. It is vital to maximize the number of probable hypotheses generated, and avoid producing hypotheses unlikely to be chosen, as only a certain number of hypotheses may be generated in a given amount of time (Al-Onaizan et al., 1999). In summary, it is necessary to find the most probable translations in the given amount of time.

Since decoding is an online task thus preserving high translation accuracy with low translation time is an essential for decoding algorithms. The decoders use language models to ensure that the output translation is grammatically correct, hence computing the language model score is a crucial part of the process, but the most expensive one as well.

Currently, a beam-search decoder implements the most leading decoding methods (Koehn, 2004). In this method, the system run-time is managed by setting a number of hypotheses to be generated, known as a beam stack. During the decoding process this number is maintained. 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 partly based on many factors such as the log-linear, 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 (Senellart and Koehn, 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 *k-best* list of candidate translations, where *k* is the source sentence length. The final sentence with the highest probability is selected and output as the chosen translation.

2.10 Evaluation

It is obvious the human judgement of the machine translation output is expensive and subjective, so automatic evaluation measures become vital. However, it is a challenging task to apply automatic evaluation of the MT output. In most cases, there is no single correct translation. Furthermore, it might be the case that two correct translations of the same sentence can have completely different words and sentence structure.

An area of active research in SMT community is said to be showing the deficiencies of the current automatic evaluation measures and proposing new measures has been. A quality metric should ideally fulfil several requirements. Ideally, a metric should be reliable, objective, give repeatable results, and, most importantly, produce

results that are meaningful in regards to quality characteristics that are evaluated.

BLEU (bilingual evaluation understudy) (Papineni et al., 2002) is the most commonly used evaluation metric, which was developed by a team at IBM. The BLEU system awards a score between 0 and 1 depending on how close an MT output is to that produced by a professional human translator. The BLEU evaluates MT 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 :

$$p_n = \frac{|C_n \cap r_n|}{|C_n|} \quad (2.3)$$

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 .

While n increases n-gram precision scores can decrease rapidly, since the likelihood of longer word sequences occurring in both the MT output and the reference translation decreases. This may end 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. Determining the combined p_n score is performed by the following equation:

$$P_n = \exp \left(\sum_{n=1}^N \frac{1}{N} \log(p_n) \right) \quad (2.4)$$

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

Where the reference translation is longer than an output 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. Calculation of the brevity penalty is performed by using the following equation:

$$BP = e^{\max(1 - \frac{\text{length}(R)}{\text{length}(C)}, 0)} \quad (2.5)$$

Where R is the reference set, and C is the candidate set. The final score is given by:

$$BLEU = BP \cdot P_n \quad (2.6)$$

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

$$\log(BLEU) = \left(1 - \frac{\text{length}(R)}{\text{length}(C)}, 0 \right) + \sum_{n=1}^N \frac{1}{N} \log(P_n) \quad (2.7)$$

2.11 Decoding Software Packages

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

Moses is a fully-featured, open-source phrase-based SMT system developed at the University of Edinburgh (Koehn et al., 2007), which allows one to train translation models using *GIZA++*⁴ for any given language pair for which a parallel corpus exists.

Dyer 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 such as rule-based, phrase-based and hierarchical phrase-based models. Several features of *Cdec* makes it advantageous over other open-source decoders. Being written in C++, it has the benefit of efficient memory usage and superior run time performance. While not being limited to extraction of just *k-best* translations, it is also capable of extracting alignments to references.

Where most MT models use Finite-State Transducers (FSTs) phrase-based models such as that used in *Moses*, lexical models, or SCFGs hierarchical phrase-based models such as that used in *Joshua* (Li et al., 2009), or *Jane* (Vilar et al., 2010), *Cdec* implements both these classes and maximizes on their benefits individually.

Dyer et al. (2010) believes on the significant lack of both phrase-based and hierarchical models in specific areas, specifically not being able to enhance conveniently to new algorithms and models. They admit this to be true as the translation, language model integration, and pruning algorithms are too closely associated, ending in either difficulty or inability to examine different translation models.

2.12 State of the art

This section gives an overview of research conducted in the topic of under-resource statistical machine translation training. (Lopez, 2008; Wu and Wang, 2007; Ueffing et al., 2007; Haffari et al., 2009; Irvine and Callison-Burch, 2015) are excellent papers which provide in-depth reviews of the state-of-the-art in SMT as well as training minimal parallel-resource SMT. For this reason, the review provided here is not as complete, but more focused on the particular aspects that this thesis aims to cover. In this section we review some of the most possibilities to minimal parallel-resource SMT. During this review, we have also highlighted various approaches such as semi-supervised learning, active learning, deep learning, and pivot language technique.

Corpus-based approaches to automatic translation for instance SMT systems use huge amounts of parallel data. This parallel data is basically created by humans for the purpose of training the mathematical models so it could be used in automatic translation. If the parallel data is required to be generated at large scale then intensive human effort and fluent bilingual translators are needed for new language pairs. Therefore it is nearly impossible to provide state-of-the-art SMT systems for the purpose of rare languages.

When it comes to SMT, both of the hierarchical and classical phrase-based translation models outperform the word-based model used for translation. If these SMT systems are provided with large parallel training corpora then they could provide high-quality translations.

⁴For the purpose of our experiments in this thesis, we apply *fast-align* tool-kit instead.

Due to the fact that using human annotation to create new parallel corpora that is enough to build the good translation system is so expensive, such data is available only for a limited language pairs. As they possess small quantities of training data set, these systems mostly manage to produce inferior translation output.

The most unfortunate point is that large quantities of parallel data are not available for a certain number of language pairs. However, there are various resources of text available for different languages. This has triggered a new research challenge in SMT related to training a SMT system where the parallel text is not sufficiently available.

2.12.1 Learning Frameworks for Minimal Parallel-Resource SMT

The problem of learning from insufficiently labelled training data has been resolved within the machine learning community with the help of two general frameworks i.e. Semi-Supervised Learning (SSL), and Active Learning (AL).

The main idea of the semi-supervised learning is to gather the cheap and excessive (unlabelled) data, blend it together with the labelled data and construct a high-quality mapping with labels. On the other hand, active learning is focused on reducing the amount of labelled data that is used for learning the high-quality mapping by asking the user to label those examples that are informative ones so that mapping could be learned with the help of lesser examples.

2.12.1.1 Semi-Supervised Learning

Supervised learning is a sort of machine learning task that infers a function from supervised training data. The training data includes a set of training texts. In supervised learning, each text involves a pair that includes an input object (typically a vector) and a desired output value (also called the supervisory signal).

A supervised learning algorithm conducts an analysis of the training data and produces an inferred function. This inferred function is commonly known as a classifier or a regression function. Before moving forward, the inferred function should suggest the correct output value related to any valid input object. Though this way, the learning algorithm will be required to generalize from the training data to unseen situations. The parallel task in human psychology is simply known as the concept learning.

Semi-supervised learning is focused on utilizing both labelled and unlabelled data so that learning performance could be improved. Recently, the issue of learning while in the presence of labelled and unlabelled data has created quite a stir. The SSL algorithm plays its function in the following ways:

- First of all, the estimation of translation model is conducted based on the sentence pairs in the bilingual training data (L).
- Secondly, the translation of the set of source language sentences (U) is done depending on the current model.
- Later on, a subset of good translations and their sources, (T_i) is selected in each iteration and then included in the training data.

Algorithm 1 shows the procedures of semi-supervised learning technique for SMT systems:

Algorithm 1 Semi-supervised learning for statistical machine translation

Input: Training set of parallel sentence pairs (L), Unlabelled set of source text (U), Development dataset (C), Number of iterations (R), Size of k -best list (K), Additional bilingual training data (T_i).

- 1: **repeat**
- 2: *Training step:* $\pi^{(i)} := \text{Estimating}(L, T_{i-1})$.
- 3: $X_i := \{\}$. // The set of generated translations for this iteration.
- 4: $U_i := \text{Filter}(U, C, i)$. // The i^{th} chunk of unlabelled sentences.
- 5: **for** sentence $s \in U_i$ **do**.
- 6: *Labelling step:* Decode s using $\pi^{(i)}$ to obtain K best sentence pairs with their scores.
- 7: $X_i := X_i \cup \{t_k, s, \pi^{(i)}(t_k|s)\}_{k=1}^K$
- 8: **end for**
- 9: *Scoring step:* $S_i := \text{Scoring}(X_i)$. // Assign a score to sentence pair (t, s) from X_i .
- 10: *Selection step:* $T_i := T_i \cup \text{Selecting}(X_i, S_i)$. // Choose a subset of good sentence pair (t, s) from X_i .
- 11: $i := i + 1$.
- 12: **until** $i > R$.

The replacement of the selected sentence pairs is done in each iteration. Bilingual training data is the only data that is kept fixed at every step of the algorithm. The certain processes such as the generation of sentence pairs, selection of subset of good sentence pairs and the updating of model are continued until a certain condition is met. This condition is the most decisive point as it is known as the stopping condition.

We must keep in mind that this algorithm is functioned in a *transductive* setting. It means that the set of sentences is either derived from test set that is used to evaluate the SMT system or the development set. If the definition of *Estimating*, *Scoring*, and *Selecting* is changed in this algorithm then it will create different sort of SSL algorithms.

While keeping the probability model $P(t|s)$ in mind, we must consider the distribution over all possible valid translations t linked to a specific input sentence. This probability distribution can be initialized to the uniform distribution of each sentence s in the form of unlabelled data. Thus, it can be said that this distribution linked to the translation of sentences from U will contain the maximum entropy. The base of this algorithm lies in minimizing the entropy of distribution over translations of U . However, this theory proves true only in the case when *Estimating*, *Scoring*, and *Selecting* functions carry prescribed definitions.

This technique is followed for a limited number of iterations but is mostly focused on finding the useful definitions for *Estimating*, *Scoring*, and *Selecting*. There are certain chances that these definitions might help in improving the SMT performance. In following we investigate the mentioned functions in-depth:

1. **The Estimating Function:** Definitions below are followed for the Estimating function according to Algorithm 1:
 - **Complete retraining of all translation models:** If Estimate (L, T) keeps an estimate of the model parameters based on $(L \cup T)$, then we will get an SSL algorithm that serves the purpose of retraining the model on the original training data L . This also includes the sentences that have been

decoded in the last iteration. If there is an issue of the size of L then it can be changed by filtering the training data.

- **Additional phrase-table:** In the case where a new phrase translation table is learned on T only and then included in the log-linear model as a new component, we can get an alternative to the full retraining of the model on labelled and unlabelled data. It could prove really expensive if the L is large. This additional phrase-table is mainly smaller in size and is very particular to the test set it is trained on. Even though, it overlaps with the original phrase-tables, still it carries various new phrase pairs.
- **Mixture model:** Another alternative for the Estimate can be provided in the case when the phrase table probabilities are blended with the new phrase table probabilities:

$$P(s|t) = \lambda \cdot L_p(s|t) + (1 - \lambda)T_p(s|t) \quad (2.8)$$

In this case, L_p and T_p stand for the phrase-table probabilities that are estimated on L and T , respectively. Where the new phrase pairs are learned from T , they end up becoming a part of the merged phrase-table.

2. **The Scoring Function:** This function plays the role of assigning score to each of the translation hypothesis t . These scoring functions are commonly used:

- **Length-normalised score:** Each translated sentence pair (t, s) is provided with score according to the model probability $P(t|s)$ which is normalised by the length ($|t|$) of the target sentence:

$$Score(t|s) = P(t|s)^{\frac{1}{|t|}} \quad (2.9)$$

- **Confidence estimation:** When it comes to calculating the confidence score of the target sentence t , then it is calculated as a log-linear combination of word posterior probabilities, phrase posterior probabilities, and a target language model score.

Translation Error Rate (TER) (Snover et al., 2006) helps in optimizing the weights of the different scores. When the sentence probabilities of all translation hypotheses that contain the phrase pair in the k -best list are summed, they help in determining the phrase posterior probabilities.

It is the decoder that conducts the segmentation of sentence into the suitable phrases. Later on, this sum is normalized with the help of total probability mass of the k -best list. In order to gain the score for the whole target sentence, we will multiply the posterior probabilities of all target phrases. The calculation of the word posterior probabilities is done on the basis of normal alignment that lies between the considered hypotheses and all the other translations contained in the k -best list. Again we will take the single values and multiply them so that the score for whole sentence could be derived.

3. **The Selecting Function:** This function is mainly used to create the additional training data T_i . This additional training data is used in the next iteration ($i + 1$) under the Estimate so it could increase the original bilingual training data. Here are some of the selection functions that will be used in this process:

- **Importance sampling:** The labelling step generates a list of translations known as k -best list for each sentence s present in the set of unlabelled

sentences U . The subsequent scoring assigns a score for each translation in the k -best list. The set of generated translations containing all the sentences in U are recognized as the event space. Probability distribution is instilled over this space through the scores just for the purpose of renormalizing them. The main idea of importance sampling is to select N translations from this distribution. Replacement strategy is followed in order to do the sampling; it means that selection of same translation may be done more than one times. The combination of N sampled translation and the source sentences linked to them result in the additional training data T_i .

- **Selection through a threshold:** In this method, the score of every 1 -best translation is compared to a threshold. If the score of the translation exceeds the threshold then it is considered to be reliable. As a result, it is included into the set of T_i . Otherwise, it is eliminated and not made a part of the additional training data. The optimization of the threshold is done according to the development beforehand. Due to the fact that scores of translations vary in each iteration, the size of T_i also varies.
- **Keep all:** This method does not involve any sort of filtering. It is assumed in this method that all the translations in the set X_i are reliable so this is the reason that none of them are eliminated. However, the result of selection step in each iteration will come out as ($T_i = X_i$). The main idea of this method was to make a healthy comparison with the other methods.

If we consider it generally then having excessive amount of training data brings improvement to the quality of the trained models. However, when we reach the procedure where a particular test set is being translated then we do have to ponder over the issue whether all of the available training data is suitable for translation or not. We must also keep in mind that excessive computational power is required when we work with large amounts of training data. The computational complexity could be decreased if the subset of training could be identified and used to retrain the models.

This identifies the parts that are related to the test set by proposing to filter the training data. It filters the data either in form of monolingual text or either bilingual text. This filtering usually takes place according to the n -gram coverage. If we talk about the source sentence s in the training data, we will come to know that its n -gram coverage over the test set is actually computed. The average over some of the n -gram lengths is used as a measure of relevance in the case of training sentences. This base is then used to select the top N source sentences or sentence pairs.

The issue that SMT system can learn something from its own output and then gets improved by semi-supervised learning, raises a lot of intuitive doubts. It could be said that there are two main reasons that lead to such improvement:

1. The selection step provides important feedback to the system. If we take the example of confidence estimation, it discards translations with low language model scores or posterior probabilities. The selection step not only reinforces the high-quality phrases but it also discards the translations done by bad machines. As a result of this selection step, the probabilities of low-quality phrase pairs degrade. These probabilities of low-quality phrase pairs include overly confident singletons or noise in the table. According to the research conducted by Ueffing et al. (2007), it became clear that selection outperforms all the methods that are used to save the generated translations in the form of additional

training data. The selection methods that were investigated during his research proved to be well-suited enough to boost performance of SSL for SMT.

2. This algorithm adapts the SMT system to a new style or domain without any sort of requirement for the development data or the bilingual training. The phrases that are suitable for translation of new data and are present in the phrase-tables are reinforced. This is the reason that probability distribution over the phrase pairs get highly-focused on the parts that are related to the test data.

Semi-supervised learning previously has been applied so that word alignments could be improved. During the research conducted by Callison-Burch et al. (2004), unsupervised learning on parallel data was used for the purpose of training the generative model for word alignment. Other than that, another model that included the small amount of hand-annotated word alignment data was also trained. The important thing about mixture model is that it provides probability for word alignment. Experiments show that if we put a large weight on the model that has been trained through labelled data then it starts performing best.

Almost similar research was conducted by Fraser and Marcu (2006) during which a generative model of word alignment was combined through the help of a log-linear discriminative model. This model was trained on a small set of hand aligned sentences. The word alignments strongly contribute towards increasing the translation quality by training the standard phrase- based SMT system.

2.12.1.2 Active Learning

In active learning a few labelled texts are provided with a set of unlabelled texts which is very large. The main idea is to arrange the set of texts in an optimal order so they could be labelled for external oracle. Then rerun is used as an underlying system to improve the performance. This continues in the iterative fashion for the purpose of convergence which is a typical threshold on the achievable performance before all the unlabelled data set could be exhausted.

If we talk about the SMT model that is initially trained on the bilingual data, then it becomes clear that the main problem is to minimize the human effort during the translation of new sentences. These new sentences will then be made a part of the training data so that retrained SMT model could achieve an improved level of performance. Thus, given a bilingual text L and a monolingual source text U , the goal is to select a subset of highly informative sentences from U to present to a human expert for translation. Highly informative sentences are basically those sentences that enable the retrained SMT model to reach an improved transaction level with the help of their translations.

The initial SMT system on the bilingual corpus L is not only trained but it is also used for the translation of all the monolingual sentences in U . The sentences in U are then branded together by denoting their translations as U^+ . After that, the retraining of SMT system on $(L \cup U^+)$ is done and the model that occurs after the retraining is used to decode the test set. Later on, it becomes easy to make a selection and removal of the subset of highly informative sentences from U . These sentences can then be add together to L through the help of their human-provided translations.

Two types of phrase-tables could be learned during the retraining of models:

1. Phrase-table is learned from L .

2. Phrase-table is learned from U^+ .

The phrase-table obtained from the U^+ is included in the log-linear translation model as a new feature. The alternate option is to ignore the U^+ as is done in the conventional AL setting. If the phrase-tables are not useful then they get the (0) score in the Minimum Error Rate Training (MERT). So this method is considered to be more beneficial.

Also, this method has empirically proved to be more effective (Ueffing et al., 2007) than (1) using the weighted combination of the two phrase-tables from L and U^+ , or (2) combining the two sets of data and training from the bi-text ($L \cup U^+$). The procedure helps to investigate that how one can take maximum advantage of human effort spent in sentence translation while learning the SMT model from the data that is available. This investigation also includes monolingual and bilingual text.

Algorithm 2 shows the procedures of active learning technique for statistical machine translation systems:

Algorithm 2 Active learning for statistical machine translation

Input: Training set of bilingual sentence pairs (L), Training set of monolingual dataset (U).

- 1: $M_{S \rightarrow T} = \mathbf{train}(L, 0)$.
 - 2: **for** $t = 1, 2, \dots$ **do**.
 - 3: $U^+ = \mathbf{translate}(U, M_{S \rightarrow T})$
 - 4: Select k sentence pairs from U^+ , and ask from someone for their true translations.
 - 5: Remove the k sentences from U , and add the k sentence pairs to L .
 - 6: $M_{S \rightarrow T} = \mathbf{train}(L, U^+)$.
 - 7: Evaluate the performance on the test set T .
 - 8: **end for**
-

The strategies of sentences selection can be divided into two different types of categories:

1. Those which look into the source language and are considered independent of the target language.
2. Those which are considered a part of the target language.

This considered AL scenario is mainly based on the second category.

- **The utility of translation units:** Phrases are basically the units of translations present in the phrase-based SMT models. The phrases that have been potentially derived from any particular sentence indicate the informativeness of that sentence. If a sentence offers more new phrases then it means that it is more informative. It is necessary that we take accurate estimate of the phrase translation probabilities because of the fact that sentences containing rare phrases are also quite informative. When we select the new sentences for the purpose of human translation, we need to keep in mind the trade-off between exploitation and exploration. This trade-off between exploitation and exploration means selecting the sentences for the discovery of new phrases versus making the accurate estimate of phrase translation probabilities. Here we need to make a similar argument so that complete emphasis could be put on importance of words instead of SMT model phrases. Also we must keep in mind

that smoothing is a means through which the accurate estimate of translation probabilities could be made especially when the events are rare.

- **Similarity to the bilingual training data:** The simplest possible way for the expansion of the lexicon set is to select the sentences from U . However, these sentences must be as dissimilar as possible to L . This is the method where measurement of similarity is conducted with the help of weighted n -gram coverage (Ueffing et al., 2007).
- **Confidence of translations:** The decoder produces an output translation t with the help of the probability $P(t|s)$. This probability is usually treated as a confidence score for the translation in order to make the confidence score for sentences carrying comparatively different lengths. If required then normalising can be done using the sentence length (Ueffing et al., 2007).
- **Feature combination:** The idea is to gather the information from several simpler methods, while producing the final ranking of sentences. We can either use the strategy to blend the output rankings of those simpler models, or we can also use the scores that they generate so we could use them as input features for a higher level ranking model. A linear model is used here:

$$W(s) = \sum_n \lambda_n \gamma_n(s) \quad (2.10)$$

The λ_n used in this linear model are known as the model parameters. The γ_n are the features functions that belong to the confidence score and are the score for utility of the translation units.

- **Reverse model:** Due to the fact that a translation system $M_{S \rightarrow T}$ is built from language S to language T , we also get to build this translation system in the reverse direction $M_{T \rightarrow S}$. In order to measure the informative nature of the monolingual sentence s , we should translate it to t by $M_{S \rightarrow T}$ and then we must convert the translation back to S through $M_{T \rightarrow S}$. Next step is to denote this reconstructed version of S sentence with s' . If we make a comparison between s and s' using BLEU or other measures then we can easily find out that how much information got lost when we used the direct and reverse translation systems. Later on, human selects the sentences with higher information loss for the purpose of getting them translated.

Active learning framework for SMT makes use of both type of data including the labelled and unlabelled. If we need to make a perfect sentence selection in the SMT active learning then we must learn to pay attention to the units of translations i.e. words and candidate phrases.

It is important to improve the coverage of the bilingual training data. However, we must keep in mind that this is not the only crucial factor. For instance, decoder confidence for sentence selection possesses low coverage still it performs well when it is in the domain adaptation scenario. Otherwise, its performance is not so acceptable.

We can found little amount of published work on active learning for SMT for domain adaptation and minimal-resource languages even though various promises have been made on the subject. Mohit and Hwa (2007) introduced a technique

through which they classified the phrases that were difficult to translate, and further they incorporated human translations for these phrases. The approach introduced by Mohit and Hwa (2007) slightly differs from AL. During their research, they sought help from the human translations improve translation output in the decoder could be improved.

Further studies are also available on sampling sentence pairs for SMT (Eck, M. and S. Vogel and A. Waibel, 2005). However, researchers have aimed to limit the amount of training data so that memory footprint of the SMT decoder could be reduced. Eck, M. and S. Vogel and A. Waibel (2005) used *n-gram* features for the purpose of computing this score. We must keep this in mind that those features were very different from the *n-gram* features proposed during this work.

2.12.1.3 Deep Learning

Deep learning is a recently used approach for MT. The Neural Machine Translation (NMT), unlike the traditional MT, is a better choice for more accurate translation and it also provides better performance. Deep Neural Networks (DNNs) with more than one hidden layer can be used to improve traditional systems in order to make them more efficient. These networks first enter into the training phase then implemented to solve the problem (Guzmán et al., 2017).

Different deep learning techniques and libraries are required for developing a better MT system. Recurrent Neural Networks (RNNs) (Zhang et al., 2016), Long-Short Term Memories (LSTMs) (Sutskever et al., 2014) etc. are used to train the system which will convert the sentence from source language to target one. Adapting the suitable networks and deep learning strategies is a suitable choice because it tuned the system towards maximizing the accuracy of the translation system as compare to others.

MT is a method to convert the source sentence from one natural language to other natural language with the help of computerized systems and human assistance is not necessary. Different approaches are available to create such type of systems but a more robust technique is required to create better system than existing systems. A well-trained network leads the system towards its goal, which is to generate more efficient translation system that is capable in providing suitable accuracy.

Deep learning is a new technique, widely use in different machine learning community. It enables the system to learn like a human and to improve the efficiency with training. Deep learning methods have the capability of feature representation by using supervised learning as well as unsupervised learning even there exist higher and more abstract layers. Deep learning currently used in big data, image applications, speech recognition, machine translation etc.

Deep learning attracts researchers for using it in MT. The main idea behind this is to develop a system that works as translator. With the help of history and past experiences, a trained DNN translates the sentences without using large database of rules.

MT consists some other related processes like word alignment, reordering rules, language modelling etc. Each process in text processing has appropriate DNN solutions. After preprocessing (sentence segmentation, translation process starts with word alignment followed by reordering and language modelling.

2.12.2 Pivoting Framework for Minimal Parallel-Resource SMT

A common solution to the lack of parallel data is using pivot (bridge or intermediary) language technique. This technique is used to generate a systematic SMT when a proper bilingual corpus is lacking or the existing ones are weak (Ahmadnia et al., 2017).

This issue becomes significant when there are languages with inefficient NLP resources to be able to provide an SMT system. However, there are sufficient resources between them and some other languages. Though it is claimed that, the intermediary languages do not lead to an improvement in general case, this idea can be employed as a simple method to enrich the translation performance even for existing systems (Matusov et al., 2008).

This idea brings a third language for translating between a source and target languages with limited bilingual text, this third language called the pivot language. For the language pairs source-pivot and pivot-target, there exist large bilingual corpora. Using only source-pivot and pivot-target bilingual resources, a translation model is built for source-target language pair.

The advantage of this technique is the fact that translation between source and target can perform even if there is no bilingual corpus available for this language pair. This point is vital because this technique provides a means of translation between many pairs of languages for which only few parallel data exist.

There is a substantial amount of work done in the area of pivot strategies for SMT. For instance, De Gispert and Mariño (2006) talked about translation task between *Catalan* and *English* while using *Spanish* as a pivot language. Pivoting is done with the help of two techniques-concatenation of two SMT systems and direct approach in which *Catalan-English* corpus is generated and trained upon.

In Utiyama and Isahara (2007), the authors conducted research on the use of pivot language through phrase translation (phrase-table creation) and sentence translation.

Wu and Wang (2007) discussed three methods for pivot strategies namely:

1. **Multiplication:** This method combines the corresponding translation probabilities of the translation models for the source-pivot and the pivot-target languages, thus generating a novel model for the source-target translation.
2. **Cascade:** This method translates the text in the source language to the pivot through employing a source-pivot translation model, and subsequently translate it to a target language utilizing a pivot-target translation model.
3. **Synthetic Corpus:** For the purpose of obtaining source-target corpus, there are two ways; First is, we can translate pivot-language sentences from the source-pivot corpus into target-language sentences using the pivot-target system. Second is, translation of pivot sentences from the pivot-target corpus into source sentences using the pivot-source system.

Assume that a small bilingual text is available for source-target language pair, a standard phrase-based translation model can be built and then an improved translation model for source-target language pair is built accordingly by performing linear interpolation on the standard model and the pivot-based model. Thus, the interpolated model can employ both the small source-target text and the large source-pivot and pivot-target corpora to improve the translation quality.

In the previous researches, we have explored the idea of using pivot languages to overcome data sparseness. A research conducted by Callison-Burch et al. (2006) during which the researchers used paraphrases so the unseen source phrases could be dealt with. They used certain methods to acquire the paraphrases by conducting an identification of the candidate phrases in the source-language. After identification, these candidate phrases were then translated into the multiple intermediate languages, and sent back to the source. Later on, the new source phrases are treated as the potential paraphrases of the originals. During this procedure, the source phrases that are left unknown are then substituted with the paraphrases and then translation applied on these paraphrases.

Pivot language approaches are also used so that the word alignments could be improved. A research was conducted by Borin (1999) during which he used multilingual corpora so that alignment coverage could be improved. During another research that was conducted by Wang et al. (2006), the researchers improved the word alignment quality by inducing alignment models through two additional bilingual corpora. Further researches were focused on Cross-Language Information Retrieval (CLIR) (Kishida and Kando, 2003), translation dictionary induction (Schafer and Yarowsky, 2002), word-sense disambiguation (Diab and Resnik, 2002), and so on through the use of pivot language methods.

A team of researchers also organized a shared task on word alignment during the *ACL (2005) Workshop on Building and Using Parallel Texts* (Martin et al., 2005). While organizing this shared task, they kept their ideas focused on the languages with limited resources. Various researchers (Lopez and Resnik, 2005; Tufis et al., 2005) carried out their experiments in relevance to the subtask of unlimited resources, during which they used language-dependent resources such as a dictionary, a thesaurus, and a dependency parser to improve word alignment results.

Nakov and Ng (2012) tried to research the similar points between resource-poor and resource-rich languages for the translation task. Dabre et al. (2015) used Multiple Decoding Paths (MDP) in order to overcome the limitation of small-sized corpora. Paul et al. (2013) conducted research over criteria to be considered for selection of good pivot language. Kunchukuttan et al. (2014) demonstrated the use of source-side segmentation as preprocessing technique. Goldwater and McClosky (2005) investigated several methods for incorporating morphological information in order to achieve better translation from *Czech to English*.

2.12.3 Other Research Lines

The previous sections give an overview of the most important relevant classic approaches to scarce-resource SMT used by the community. This section reviews the other recent efforts that were carried out in order to overcome the training data scarcity limitations of baseline systems.

It is a general perception in the SMT systems that if the available training corpus is larger, then performance of the translation system will be better. Whereas the task of finding appropriate monolingual text for the language model is not considered as difficult, acquisition of a large high-quality bilingual parallel text for the desired domain and language pair requires a lot of time and effort. It happens because acquisition of a large high-quality bilingual parallel text is not even possible for some language pairs. In addition, small corpora have certain advantages such as:

- Manual creation of the corpus becomes possible.
- Automatically collected corpus could be manually corrected.

- Low-memory and time requirements for the training of a translation system.

These are the reasons that strategies for exploiting limited amounts of bilingual data are receiving more and more attention.

2.12.3.1 Bilingual Lexicon Induction for Minimal Parallel-Resource SMT

Exploiting bilingual lexicon induction techniques (Irvine, A. and C. Callison-Burch, 2016) learn translations from monolingual texts in two languages. This is done in order to build an end-to-end SMT system without the use of any bilingual sentence-aligned.

Parallel corpora are one of the possibilities that could help overcome the limitation of training data scarcity. Bilingual lexicon induction describes the class of algorithms that attempts to learn translations from monolingual corpora.

The most prominent problem that arises when an MT system has access to limited parallel resources is the fact that there are many unknown words that are Out-Of-Vocabulary (OOV) with respect to the training data, but which do appear in the texts that we would like the SMT system to translate.

Bilingual lexicon induction can be used to try to improve the coverage of under-resource translation models, by learning the translations of words that do not occur in the parallel training data. Although past research into bilingual lexicon induction has been motivated by the idea that it could be used to improve SMT systems by translating OOV words, it has rarely been evaluated that way.

Some of the notable exceptions of past researches that did evaluate bilingual lexicon induction in the context of SMT through better OOV handling include Dou and Knight (2013) and Dou et al. (2014).

Despite the above mentioned researches, the majority of prior work in bilingual lexicon induction has treated it as a standalone task, without even thinking about integrating induced translations into end-to-end SMT. Instead the evaluation was done by holding out a portion of a bilingual dictionary and analysing that how well the algorithm learns the translations of the held out words.

Bilingual lexicon induction uses monolingual or comparable corpora, usually paired with a small seed dictionary, to compute signals of translation equivalence. Consider for a second that bilingual dictionaries and only a small amount of parallel training data are available:

- In the first case, a baseline system that produces a simple dictionary gloss with additional translations that are learned using monolingual corpora in the source and target languages is generated.
- In the second case, a baseline statistical model learned over small amounts of parallel training data with additional translations and features estimated over monolingual corpora is generated as well.

The idea is making effective use of bilingual lexicon induction, which allows learning translations from independent monolingual texts or comparable corpora that are written in two languages. SMT typically uses sentence-aligned bilingual parallel texts to learn the translations of individual words (Brown et al., 1990).

Another thread of research has examined bilingual lexicon induction which tries to induce translations from monolingual corpora in two languages. The range of these monolingual corpora starts from being completely unrelated topics and falls

to being comparable corpora. The bilingual lexicon induction is framed as a binary classification problem; for a pair of source and target language words, predicting whether the two are translations of one another or not is required.

Since binary classification does not inherently provide a list of the best translations, taking an additional step is required. For a given source-language word, its best translation or its *k-best* list translations by first using the classifier on all target language words is fined. Then ranking them based on how confident the classifier is that each target-language word is a translation of the source word is applied.

Additional related work on learning translations from monolingual corpora are as follows:

- Carbonell et al. (2006) described an SMT system which produced translation lattices using a bilingual dictionary and scored them using an *n-gram* language model. Their method had no notion of translation similarity aside from being a bilingual dictionary.
- Similarly, Sanchez-Cartagena et al. (2011) supplemented an SMT phrase-table with translation pairs extracted from a bilingual dictionary and gave each a frequency of one for computing translation scores.
- Ravi and Knight (2011) treated SMT without parallel training data as a decipherment task and learned a translation model from monolingual text. They also translated corpora of *Spanish time expressions* and *subtitles*, which both has a limited vocabulary, into *English*. Their method has not been applied to broader domains of text. Most work on learning translations from monolingual texts only examined small numbers of frequent words.
- Daume III and Jagarlamudi (2011) were exceptions that improved SMT by mining translations for OOV items.
- A variety of past researches have focused on mining parallel or comparable corpora from the web (Munteanu and Marcu, 2006; Smith et al., 2010).
- Others used an existing SMT system to discover parallel sentences within independent monolingual texts, and used them to retrain and enhance the system (Chen et al., 2008; Abdul-Rauf and Schwenk, 2009; Lambert et al., 2011).

2.12.3.2 Monolingual Collocation for Minimal Parallel-Resource SMT

Making effective use of the collocation probabilities is one possible way to improve the SMT performance (Liu et al., 2010). The collocation probabilities are estimated from monolingual corpora, in two aspects:

1. Improving word alignment for various kinds of SMT systems.
2. Improving phrase-table for phrase-based SMT systems.

Collocation is generally defined as a group of words that occur together more often than based on chance (McKeown and Radev, 2000). A collocation is composed of two words occurring as either a consecutive word group or an interrupted word group in sentences. In this method the Monolingual Word Alignment (MWA) method (Liu et al., 2010), is used for the purpose of collocation extraction. This method adapts the Bilingual Word Alignment (BWA) algorithm to MWA scenario to extract collocations only from monolingual corpora.

Statistical bilingual word alignment (Brown et al., 1993) is the base of most SMT systems. As compared to single-word alignment, multi-word alignment is more difficult to be identified. Although many methods were proposed to improve the quality of word alignments (Marcu and Wong, 2002; Cherry and Lin, 2003; Huang, 2009), the correlation of the words in multi-word alignments was never fully considered.

In phrase-based SMT the phrase boundary is usually determined based on the bi-directional word alignments, but few previous studies exploited the collocation relations of the words in a phrase. Some researchers used soft syntactic constraints to predict whether source phrase can be translated together (Marton and Resnik, 2008; Xiong et al., 2009). However, the constraints were learned from the parsed corpus, which is not available for many languages.

The idea to use the monolingual collocations was the first one to identify potentially collocated words and estimate collocation probabilities from monolingual corpora using a MWA method which does not need any additional resource or linguistic pre-processing. Plus, it outperforms previous methods on the same experimental data. Furthermore, the collocation information is employed in order to improve BWA for various kinds of SMT systems and to improve phrase-table for phrase-based SMT.

In order to improve BWA, re-estimating the alignment probabilities by using the collocation probabilities of words in the same cept⁵ is required. An alignment between a source multi-word cept and a target word is a many-to-one multi-word alignment. In order to improve the phrase-table, calculating phrase collocation probabilities based on word collocation probabilities is required. Later on, the phrase collocation probabilities are used as additional features in phrase-based SMT systems.

2.12.3.3 Domain Adaptation for Minimal Parallel-Resource SMT

Domain adaptation has recently gained interest in SMT to cope with the performance drop observed when testing conditions deviate from training conditions. The basic idea is that in-domain training data can be exploited to adapt all components of an already developed system.

The main idea is exploit large but cheap monolingual in-domain data, either in the source or in the target language. This idea proposes to synthesize a bilingual corpus by translating the monolingual adaptation data into the counterpart language.

This approach focuses on the issue of adapting an already developed phrase-based translation system in order to work properly on a different domain. There is almost no parallel data available for this phrase-based translation system but only the monolingual texts. In this technique, a lexicalized reordering model is also exploited to control reordering of target words. This model is also learnable from parallel data.

Assuming some large monolingual in-domain texts are available. In such a case, we can only pursue two basic adaptation approaches:

1. Generating synthetic bilingual data with an available SMT system, and use this data to adapt its translation and reordering models.
2. Using synthetic or provided target texts to also, or only, adapt its language model.

⁵A cept is the set of source words that are connected to the same target word.

Once monolingual adaptation data is automatically translated, the synthetic parallel corpus can be used to estimate new language, translation, and reordering models. Such models can either replace or be combined with the original models of the SMT system.

In the last years various publications have dealt with the issue of sparse bilingual corpora.

- In (Niesen et al., 2004) the impact of the training corpus size for SMT from *German* into *English* was investigated where the use of a conventional dictionary and morpho-syntactic information for improving the performance was proposed. They used several types of word reordering as well as a hierarchical lexicon based on the Part-Of-Speech (POS) tags⁶ and base forms of the *German* language. They reported results on the full corpus of about (60,000) sentences, on the very small part of the corpus containing five thousand sentences and on the conventional dictionary only.
- Morpho-syntactic information yields significant improvements in all cases and an acceptable translation quality is also obtained with the very small corpus. SMT of spontaneous speech with a training corpus containing about (3,000) sentences has been dealt with in (Matusov et al., 2004). They proposed the acquisition of additional training data using an *n-gram* coverage measure, lexicon smoothing and hierarchical lexicon structure for improving word alignments as well as several types of word reordering based on POS tags. The *Spanish-English* and *Catalan-English* SMT with sparse bilingual resources in the tourism and travelling domain were investigated in (Popovic and Ney, 2005). The use of a phrasal lexicon as an additional language resource was proposed as well as introducing expansions of the *Spanish* and *Catalan* verbs. With the help of the phrasal lexicon and morphological information, a reasonable translation quality is achieved with only (1,000) sentence pairs from the domain.
- The *Serbian-English* SMT was investigated in (Popovic et al., 2005). A small bilingual corpus containing less than (3,000) sentences was created and SMT systems were trained on different sizes of the corpus. The obtained translation results are comparable with results for other language pairs, especially if the small size of the corpus and rich inflectional morphology of the *Serbian* language are taken into account. Morpho-syntactic information is shown to be very helpful for this language pair. The *Czech-English* SMT and the impact of the morphological information were investigated in (Goldwater and McClosky, 2005). As with *Serbian-English*, morphological transformations have an important role for the translation quality. The problem of creating word alignments for languages with scarce resources i.e. *Romanian-English* and *Hindi-English* had been addressed in (Lopez and Resnik, 2005; Martin et al., 2005).
- A shared task on word alignment was organized as part of the ACL 2005⁷ Workshop on Building and Using Parallel Texts (Martin et al., 2005), focusing on languages with scarce resources.

⁶In corpus linguistics, POS tagging is the process of marking up a word in a corpus as corresponding to a particular part of speech, based on both its definition and its context.

⁷The 43rd annual meeting of the Association for Computational Linguistics, Ann Arbor, USA, June 2015.

- For the subtask of unlimited resources, some researchers (Aswani and Gaizauskas, 2005; Lopez and Resnik, 2005; Tufis et al., 2005) used language-dependent resources such as a dictionary, a thesaurus, and a dependency parser to improve word alignment results.

2.13 Summary

During this chapter, we reviewed all the theoretical and mathematical background knowledge needed to follow the rest of the thesis, including machine translation concepts, translation procedures and requirements based on statistical approach of machine translation, evaluation metric and decoders, and state-of-the-art based on widely-used approaches based on improving the translation performance of minimal-resource language pairs such as semi-supervised learning, active learning, deep learning, pivot language technique, and some other recent research lines.

The advantages of machine translation over human translators are becoming more numerous with research advances in natural language processing. Machine translation has seen significant development in the last years with the most popular approach now tending towards statistical machine translation because of the numerous advantages this approach holds over others.

Very generally, the task of statistical machine translation is based on the log-linear model form. A baseline translation system consists of a training model generated from the parallel corpus aligned on phrase level, a language model from the monolingual corpus in the target language, and a translation model. The translation mode determines the probability of target sentence t being linguistically the equivalent of source (input) sentence s . 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.

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

There are several open-source decoders that can be used for statistical machine translation, such as *Moses* and *Cdec*. Individual differences in these decoders have certain effects on the system and its output as a whole, and will be covered in subsequent chapters.

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 metric is BLEU. The BLEU scores output by comparing parameters of translation length, word choice, and word order to a reference text.

Based on state-of-the-art, the semi-supervised learning algorithm that we saw in this chapter is a kind of learning framework which learns from its own output in an iterative manner. This is similar to active learning with one major difference; in active learning the labels data are provided by human.

Deep learning as a kind of machine learning methods based on learning data representations, as opposed to task-specific algorithms. Learning can be supervised, partially supervised or unsupervised. Deep learning architectures such as deep neural networks, deep belief networks and recurrent neural networks have been applied to fields including computer vision, speech recognition, natural language processing, machine translation, and etc. where they produced results comparable to and in some cases superior to human experts.

The pivot language approach, as an alternative to overcome the training data bottleneck, discussed in this chapter, can be seen as a practical, time and cost-effective option for producing translations for minimal-resourced and rare language pairs despite the further quality control or translation strategies that may be needed to complete the translation process. The right selection of the pivot language for each translation scenario may play an important role in the quality of the generated translations.

In this chapter more research lines have been reviewed as well including bilingual lexicon induction, monolingual collocation, and domain adaptation.

Chapter 3

Phrase-Based Translation Models for SMT Systems

Research in the field of Statistical Machine Translation (SMT) has made new developments during the last years. The recent research has focused on *Classical* and *Hierarchical* phrase-based translation models. These recently researched translation models incorporated the different levels of linguistic annotation while keeping in account the recursive nature of language.

This chapter makes a detailed comparison between Classical phrase-based translation model and Hierarchical phrase-based translation model while using the statistical approach for MT. For the experiments we have used three language pairs; *Spanish-English*, as well as *English-Persian* and *Persian-Spanish*, in order to analyse the performance quality of the above mentioned translation models. The three pairs of languages that we have used for this investigation vary with each other due to their distinct sentence structure, word order, and quantity of bilingual data.

Our experimental results show the performance of Classical phrase-based translation model as well as Hierarchical phrase-based translation model, in each translation direction and back translation as well. Also our results indicate than which translation model is preferable for our considered language pairs and translation direction. We seek to explain why this is so, and detail a series of experiments with our SMT systems using bilingual corpora each with both tool-kits *Moses* and *Cdec*. The former one is used as a Classical phrase-based platform, and the latter one is used as a Hierarchical phrase-based platform.

Furthermore, in order to prove our hypothesis, we have analysed the performance of Classical phrase-based translation model as well as Hierarchical phrase-based translation model by investigating the impact of different statistical language models; applying three kinds of *n-gram* language models on the aforementioned language pairs in each translation direction. The results confirmed that independent of the applied *n-gram* language models, our hypothesis about the performance of phrase-based translation models on our case-study language pairs is absolutely true. However, the performance of translation models depends on the word order and sentence structures of the considered language pairs. The comparative performance between our case-study language pairs based on Classical and Hierarchical phrase-based translation models will be conducted to set as state-of-the-art for further researches on phrase-based SMT.

3.1 Introduction

Phrase-based translation models are used for the purpose of viewing translation of small text pieces but this is usually done with the help of a slight reordering (Koehn et al., 2003; Och F. J. and H. Ney, 2004).

The recent research that has been carried out on SMT is focused on modelling translation of the phrases found in the source language. These phrases are then matched with equivalents present in the target language that are available in the statistically-determined form. This translation model is regarded as Classical (standard or conventional) phrase-based model. The most critical point in the Classical phrase-based model arrives when translation model is determined from parallel corpus. However, most of the times, Classical phrase-based model fails while capturing the essence of different language pairs (Birch et al., 2008). One of the biggest reasons for this failure is that reordering does not always require that it must be reduced to the level of atom phrase units.

Hierarchical phrase-based translation model even moves one step forward than standard phrase-based model by offering the phrases that have gap between them; these phrases with gaps are regarded as Synchronous Context-Free Grammar (SCFG). When it comes to the original hierarchical implementation, the SCFG model is trained in the same way as Classical phrase-based model.

The calculation of the probabilities of translation involves a sub-sample of occurrences that has been derived from the given source phrase. The determination of the parameters that are related to the phrase translation can be done through the runtime when the target language text and word alignment data will be available for the translation system. Algorithm 3 describes the procedure of forming phrase-based translation models in general:

Algorithm 3 Building phrase-based translation systems

Input: Parallel corpus between source and target languages. // Sentence-by-sentence translations of source language into target language.

- 1: **Aligning:** Learn bi-directional alignments from the parallel corpus.
- 2: **Extraction:** Extract phrase pairs from the alignments, and compute probability-based feature values each translation pair. // This is called the translation model.
- 3: **Tuning:** Learn the weights for the features by maximizing BLEU score on a development set using discriminative Minimum Error Rate Training (MERT) or Merge In-fused Relax Algorithm (MIRA).
- 4: **Decoding:** Using a language model and translation model, and translating a test set.

Output: TM. // A translation model.

3.2 Classical Phrase-Based Translation Model

Phrase-based translation models make an improvement in their performance level with the help of estimating translation probabilities. Several word tokens get translated as an atomic unit during this process which are called a phrase.

Phrases pairs that represent the translated meaning of one another are stored in a phrase-table. In the typical means, the basic requirement of a translation process is word-reordering and word-disambiguation. When we work on the phrase-level,

modelling these things in one particular step becomes easy. There are various decisions involved in this process that need not be included in the phrase-based model for the purpose of direct translation in the one step.

3.2.1 Noisy-Channel Model

The noisy-channel model is an effective way to conceptualize many processes in Natural Language Processing (NLP). As we mentioned in Section 2.5, SMT is a decision problem, hence the phrases in both source and target languages must be decided upon each other. In SMT the input is a source language string $s_1^J = s_1 \dots s_j \dots s_J$, which is to be translated into a target language string $t_1^I = t_1 \dots t_i \dots t_I$. We are told by the statistical decision theory that among all possible target language sentences, we should choose the sentence which minimises the expected loss:

$$\hat{t}_1^I = \arg \min_{I, t_1^I} \left\{ \sum_{I', t_1^{I'}} P(t_1^{I'} | s_1^J) \times L(t_1^I, t_1^{I'}) \right\} \quad (3.1)$$

This is the *Bayes* decision rule for SMT. Here,

$$L_{0-1}(t_1^I, t_1^{I'}) = \begin{cases} 0 & \text{if } t_1^I = t_1^{I'} \\ 1 & \text{else} \end{cases} = 1 - \delta(t_1^I, t_1^{I'}) \quad (3.2)$$

$L(t_1^I, t_1^{I'})$ denotes the loss function under consideration. It measures the loss or errors of a candidate translation t_1^I assuming the correct translation is $t_1^{I'}$. $P(t_1^{I'} | s_1^J)$ denotes the posterior probability distribution over all target language sentences t_1^I given the specific source sentence s_1^J . Note that the Bayes decision rule absolutely relies on the loss function $L(t_1^I, t_1^{I'})$. In case of minimising the sentence or stringing error rate, we are provided with this corresponding loss function: Here, Equation (3.2) denotes the *Kronecker-function*. (0-1) loss is this loss function as it assigns a loss of zero to the correct solution and a loss of one otherwise. By the aid of (0-1) loss, Bayes decision rule can be simplified to:

$$\hat{t}_1^I = \arg \max_{I, t_1^I} \{P(t_1^I | s_1^J)\} \quad (3.3)$$

This decision rule is also called the Maximum A-Posteriori (MAP) decision rule. Thus, we select the hypothesis which maximises the posterior probability $P(t_1^I | s_1^J)$. In the original work on SMT (Brown et al., 1990), the posterior probability was decomposed:

$$P(t_1^I | s_1^J) = \frac{P(s_1^J | t_1^I)}{P(s_1^J)} \quad (3.4)$$

Note that the denominator $P(s_1^J)$ depends only on the source sentence s_1^J and, in case of the MAP decision rule, can be omitted during the search:

$$\hat{t}_1^I = \arg \max_{I, t_1^I} \{P(t_1^I) \times P(s_1^J | t_1^I)\} \quad (3.5)$$

This model is called the *noisy-channel* model, the so-called fundamental equation of SMT (Brown et al., 1993). The decomposition into two knowledge sources is known as an approach to SMT by noisy-channel (Brown et al., 1990). The noisy-channel is a more traditionally-used model, but has been largely replaced with the

log-linear model, as it has been proved to be beneficial comparing to the noisy-channel model in many different fields.

The noisy-channel model consist of two feature scores; $P(s_1^J|t_1^I)$ and $P(t_1^I)$. The feature $P(s_1^J|t_1^I)$ is generally known as the *translation model*, embodying the probability of source sentence s and target translation t when they are considered linguistically equivalent. The feature $P(t_1^I)$ is known as the *language model*, embodying the probability of translation t being a valid sentence in the target language. If we speak intuitively, translation model maintains the content preservation which can be termed as a nominal task while language model maintains the fluency of generated translation.

If we want to generate the target translation t from the source sentence s , the possibility is:

1. Sentence s could be segmented into phrases.
2. The phrases could be translated using the bilingual phrase dictionary.
3. Reordering the phrases that have been translated.

We must keep in mind that a phrase is a sequence of words that is contagious; it is not necessarily required to be a semantic unit or a syntactic unit. Phrase pairs are commonly learned from the substring pairs observed during the bilingual training data. But, they do capture idiomatic terms and local reordering.

3.2.2 Log-Linear Model

This model is distinct from the noisy-channel model since this model is able to express scoring based on an unlimited number of features. From this point of view, it can be defined 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 rules¹. The log-linear model can be derived by the direct modelling of the posterior probability $P(t_1^I|s_1^J)$. Using a log-linear model was proposed in Papineni et al. (1998).

$$P(t_1^I|s_1^J) = \rho_{\lambda_1}^M(t_1^I|s_1^J) \quad (3.6)$$

$$\rho_{\lambda_1}^M(t_1^I|s_1^J) = \frac{\exp(\sum_{m=1}^M \lambda_m h_m(t_1^I, s_1^J))}{\sum_{t_1^{I'}} \exp(\sum_{m=1}^M \lambda_m h_m(t_1^{I'}|s_1^J))} \quad (3.7)$$

Here, we are provided with models and model scaling factors. Again, a normalization factor that depends only on the source sentence s_1^J is considered as the denominator. Consequently, we can omit it while searching in case of the MAP decision rule. The result is a linear combination of the individual models $h(t_1^I, t_1^{I'})$:

$$\hat{t}_1^I = \arg \max_{I, t_1^I} \{P(t_1^I|s_1^J)\} \quad (3.8)$$

$$\hat{t}_1^I = \arg \max_{I, t_1^I} \left\{ \frac{\exp(\sum_{m=1}^M \lambda_m h_m(t_1^I, s_1^J))}{\sum_{t_1^{I'}} \exp(\sum_{m=1}^M \lambda_m h_m(t_1^{I'}|s_1^J))} \right\} \quad (3.9)$$

¹ $\log(A.B) = \log(A) + \log(B)$

$$\hat{t}_1^I = \arg \max_{I, t_1^I} \left\{ \sum_{m=1}^M \lambda_m h_m(t_1^I, s_1^J) \right\} \quad (3.10)$$

Where Equation (3.10) is the final form of the log-linear model. In this equation, M denotes the number of features to be added, and individual scoring is undertaken by multiplying λ_m and $h_m(t_1^I, s_1^J)$, λ_m being an importance-indicating weight, and $h_m(t_1^I, s_1^J)$ 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, in other words, the log-linear model as shown in Equation (3.10) is merely a general solution expressed in the noisy-channel approach.

Log-linear model is considered superior comparing to noisy-channel model for the reason that the features importance in the model can be adjusted for to controlling each feature influence on the overall output. For instance, through controlling the values of λ_m and $h_m(t_1^I, s_1^J)$ this task is performed. The model scaling factors λ_m^M are trained according to the maximum class posterior criterion.

More features may be added to the model and the λ_m and $h_m(t_1^I, s_1^J)$ values defined to suit the particular features function within the model, such as modifying the level of operation of either the translation or language model. Alternatively, these can be trained with respect to the final translation quality which is measured by an error criterion (Och, F. J., 2003). This is the so-called Minimum Error Rate Training (MERT). Being superior and adaptable to different systems, the log-linear model was applied in this system's development.

3.2.3 Feature Functions

A feature function converts a pair of target and source sentences to the score value that is non-negative. We must keep in mind that it can be any function that performs this procedure. Every feature function can be unravelled through local evaluations at two different levels i.e. the phrase level and the word level. Global features are computed with the help of decoding process or entire derivation including the:

1. **Distortion:** These are a certain number of source words that fall between two source phrases. By distortion, we mean the translation of these words into successive target phrases.
2. **Word Penalty:** These are some of the generated target words that can help in controlling the length of translation.
3. **Phrase Penalty:** These are some of the phrase pairs that can be used in derivation $|D|$.
4. **Language Model:** It is basically the logarithm of an n -gram target language model. This feature is known for its restrictive context use surrounding the individual phrase pairs.

$$\log P(t) = \log \prod_{j=1}^T P(t_j | t_{j-1}, \dots, t_{j-n}) \quad (3.11)$$

Here the basic requirement is to present the maintenance history for each position of the n words in the target sentence. Remaining features can be found

on each of the individual phrase pairs such as phrase translation probabilities, lexical weighting and lexical reordering.

5. **Translation Probabilities:** The conditional translation probability of the target phrase is derived using the source phrase.

$$\log \prod_{(t,s) \in d} P(t|s) \quad (3.12)$$

If we are provided with t as the target phrase and s as the source phrase, the equivalent phrase probability is also computed in the opposite direction $P(s|t)$ for the same phrase pair. The entire procedure adheres to the noisy-channel model that practically proved to produce a performance that was comparable to the direct probability $P(s|t)$ (Al-Onaizan et al., 1999). When we are about to build the phrase-table, we will require to take the proper estimation of the individual probabilities that vary on certain points with the phrase alignment model.

$$P(s|t) = \frac{\text{count}(s, t)}{\text{count}(t)} \quad (3.13)$$

Here the numerator of the fraction provides representation of the number of the joint occurrences of the alignment of the both the phrases (s, t) . Whereas, the denominator of this fraction provides representation of the marginal counts of the phrase t^2 .

6. **Lexical Weight:** Translation probabilities that are derived from relative frequency estimation emerging between the phrase pairs come up extremely rough due to the issue of data sparsity. Lexical weighing is basically used as a smoothing method for the infrequent phrase pairs. The probabilities of lexical weighing method are poorly estimated (Foster et al., 2006). Word-to-word translation probabilities smoothing for which statistics are available. The target-source lexical weighting is:

$$\phi(t|s, A) = \log \prod_{j=1}^T \frac{1}{|\{i : (i, j) \in A\}|} \sum_{i:(i,j) \in A} P(s_i|t_j) \quad (3.14)$$

If A refers to underlying word alignment, the reverse lexical weighting $\phi(s|t, A)$ will also be defined in a similar manner. The conditional probabilities $P(ss_i|t_j)$ will be defined in the way similar to phrase conditional probabilities.

7. **Lexicalized Reordering:** These features are derived from the orientation where source phrase is translated with the help of previously translated phrase. Reordering can be used to denote the distance between both of these source phrases. In order to avoid sparsity of any type, the limits of orientation are maintained to some heuristics and categories. The most commonly-used of these categories are monotone with the previously translated source phrase.

² $P(t|s)$ is defined similarly.

3.2.4 Phrase Extraction

The subsequence of every target sentence and every parallel source is commonly termed as a phrase pair candidate. If all of the candidates are kept into the consideration then it becomes clear that the space of all the possible phrase pairs is quite huge and will be termed as a computational bottleneck. Another important point that we need to keep in mind is that not all of these phrase pairs will be used for SMT model as not all of them are suitable. In order to resolve this issue, the size of this space is decreased with the help of different mechanisms.

When we talk about phrase extraction, this technique mainly consists of phrase-based SMT system uses the word alignment between the target sentences and source to prune many of the useless phrase pairs. Intuitively the word alignment that lies between any sentence pair denotes correspondence between the words of two sentences. As showed in Figure 3.1 if we use the word alignment then it would help us to understand that every seventh word in the *English* sentence is basically the translation of the second word in the *Persian* sentence.

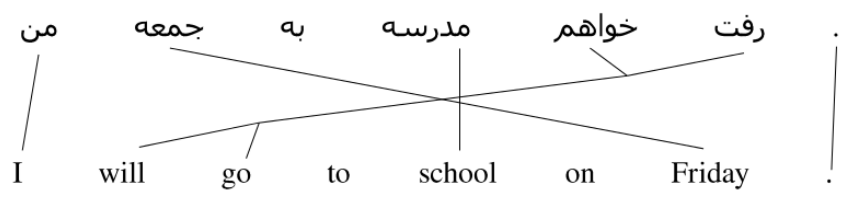


FIGURE 3.1: Alignment sample of Persian-English parallel sentence.

In simple words, the process of producing phrase translation model starts from phrase extraction algorithm. Below is an overview of phrase extraction procedure.

1. Collect word translation probabilities of source-target and vice versa using IBM models³.
2. Use the forward model and backward model from the previous step to align words for source to target and target to source, respectively. Only the highest probability is choose for each word.
3. Intersect both forward and backward word alignment point to get highly accurate alignment point.
4. Fill additional alignment points using heuristic growing procedure.
5. Collect consistence phrase pair from the previous step.

3.2.5 Phrase-Table Induction

The translations candidates that are used for the input source sentence are usually built from the pre-generated set of phrase pairs⁴ commonly known as *bi-lexicon*. In normal circumstances, a phrase alignment is determined for each of the sentence pair and then the selected phrase pairs are gathered from the entire dataset. As this method is quite effective so it is used in most of the state-of-the-art translation systems.

³IBM models are probabilistic models to transform a sentence s in one language to its translation t in another language.

⁴These set of phrase pairs is generated from a sentence-aligned parallel text.

The phrase-table is a sort of data structure that contains all of the phrase pairs that are found in the bi-lexicon. It is mainly used in the phrase-based translation systems. All the features that are used by the model are basically predetermined and stored in the phrase-table. It can also be said that phrase-table is a data structure that provides representation to each source phrase along with the possible translation and values of the associated parameter. A set of feature functions are determined for each of the phrase pair in the phrase-table and then used to score the translation candidates.

3.2.6 Learning Weights

Due to the fact that we provided the feature functions of the log-linear model, we also need to mention the concerned *weights* (λ). Instead of adopting the strategy of directly learning the model parameters so that likelihood could be optimized, the main aim is to learn them through direct optimization in order to ensure the translation quality because likelihood has just a loose relation with the final translation quality on the unseen text (Och, F. J., 2003).

One method is to perform the grid-based search: for this purpose we must start from the random point in \mathfrak{R}^N where N would be considered number of feature functions. Here we would learn one parameter at a time while we will keep the others fixed. While learning that one parameter, we will change the value of that one parameter with the help of some step size.

If we take the small steps, they might lead to improved parameter value but at the same time the convergence will be low. MERT is an algorithm which is called *specialized line search* algorithm. It is used for the purpose of finding the weights of the log-linear model in the Classical phrase-based SMT.

3.2.7 Solving Search-Problem

Making a general search for the best translation candidate for the purpose of given source sentence is considered to be complicated (Knight, 1999). It can be reformulated as a search problem with classic Artificial Intelligence (AI).

Most of the times, researchers use the effective heuristic search algorithms that have been developed in the AI community. The most notable of these algorithms is *beam-search* which is used to address the decoding problem. The main aim of the beam-search algorithm is generating the target sentence by creating translation of the phrases from different parts of the source sentence. Beam-search algorithm conducts this translation on repeated basis. We must stick to these two steps until all of the words have been translated from the source sentence:

1. A phrase must be selected from the untranslated part of the source sentence.
2. This phrase must be translated while keeping in view the phrase-table.

If a source sentence is covered in an incomplete way and its corresponding translation is also incomplete then it will also be called a hypothesis. Each of the iterations will use the above mentioned two steps to transform one hypothesis to several next hypotheses with some costs attached to the feature functions and their weights in the log-linear model such as reordering cost, language model cost, and translation cost.

The biggest feature of the beam-search algorithm is that it discards all the low-quality hypotheses and bounds itself to the part of a search-graph that is promising.

The main idea of the algorithm is to keep a subset of the all the best hypotheses, expand them depending on the previously mentioned operations, and prune the inferior hypotheses after analysing their total costs.

In pruning, while comparing the hypotheses, only those with the same amount of source-sentence words are considered. Algorithmically, this stack-based beam-search works as follows:

1. First we need to create the empty hypothesis and divide the length of the source sentence into different stacks.
2. Then we must start from the stacks that correspond to the smaller number of the source words, as long as the unexpanded hypothesis prevails.
3. Next we must expand the hypothesis and consider the new hypothesis for the purpose of pruning or entering the corresponding stacks.

Figure 3.2 illustrate that a hypothesis is expanded to new hypotheses, and they are placed into new stacks according to the number of source-sentence words which they cover.

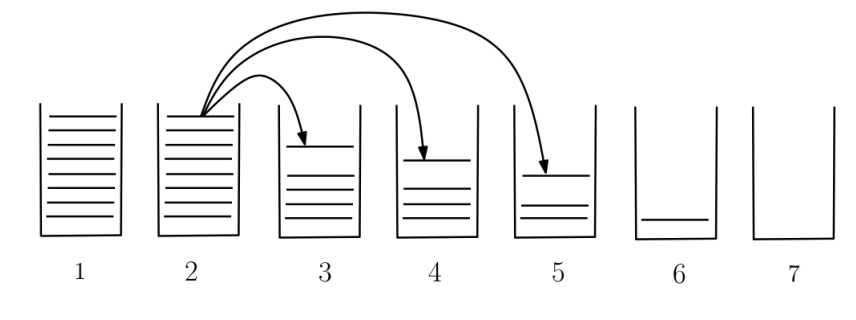


FIGURE 3.2: The stack-based beam-search schema.

This is for the running example in Figure 3.1 in which the source *English* sentence has seven words (tokens). We can either try the histogram pruning or threshold pruning in order to confine the number of hypotheses in a single stack. In the histogram pruning, the maximum capacity of the stack is set in advance. But in threshold pruning, the hypotheses that carry the costs above the range of the multiplicative factor are discarded.

3.2.8 Classical Model Decoding

As we mentioned in Section 2.9, decoding is the process during which the most probable translation for the input sentence is searched in accordance with the out models. Due to the fact that the sentence is unseen in most of the training data cases so we will have to break it into smaller units. For this purpose, we possess or must possess sufficient statistical evidence. Here we must keep in mind that each of the units corresponds to the grammar rule so the main idea behind decoding algorithms is to connect together these rules for the optimal sentence translation.

The biggest issue of the decoding process is the way how units interact with each other. Reordering models in the standard phase-based decoding are the ones that consider the input position of the output phrases neighbouring. However, the *n-gram* language models tie the translated words together more severely. As this tie-up

is generated by several rules so it is not possible to perceive the sentence translation as the independent combination of the applied translation rules.

In simple words it can be said that it is not easy to search for the most suitable rules to apply to a sentence because we have to take certain scoring functions into account. By building the flow of translation from left to right, the conventional Phrase-based decoding might move forward in a sequential way.

For decoding an algorithm that attempts to find $\arg \max_{y \in Y(x)} f(y)$ where assuming $y = p_1, p_2, \dots, p_L$:

$$f(y) = h(e(y)) + \sum_{k=1}^L g(p_k) + \sum_{k=1}^{L-1} \eta \times |t(p_k) + 1 - s(p_{k+1})| \quad (3.15)$$

The problem detected in $\arg \max_{y \in Y(x)} f(y)$ is basically NP-hard⁵ problem if we keep this definition of $f(y)$; in view then the described algorithm is the approximate method that cannot find the optimal solution.

The first critical data structure that is present in the algorithm is known as *state*. A state is basically a tuple (e_1, e_2, b, r, α) , where e_1 and e_2 are *English* words, b is a bit-string of length n^6 , r is an integer specifying the end-point of the last phrase in the state, and α is the score for the state.

Any sequence of the phrases can be adjusted according to the corresponding state. For instance, $y = (1, 3, we\ must\ also), (7, 7, take), (4, 5, this\ criticism)$, would be adjusted to the state $(this, criticism, 1111101, 5, \alpha)$. The state records last and the second last word in the translation underlying this sequence of phrases, namely *this* and *criticism*. The words that have been translated are recorded by the bit-string. The recording is as follows: the i^{th} bit is considered equal to (1) after the translation, otherwise it would be considered equal to (0). In the above mentioned case, only the (6^{th}) bit will be considered equal to (0) as it has not been translated. The value ($r = 5$) reveals that the final phrase in this sequence, $(4, 5, this\ criticism)$ will end at (5). Finally, α will be the score of the partial translation, calculated as:

$$\alpha = h(e(y)) + \sum_{k=1}^L g(p_k) + \sum_{k=1}^{L-1} \eta \times |t(p_k) + 1 - s(p_{k+1})| \quad (3.16)$$

Where $L = 3$, we have $e(y) = we\ must\ also\ take\ this\ criticism$ and $p_1 = (1, 3, we\ must\ also), p_2 = (7, 7, take), p_3 = (4, 5, this\ criticism)$.

Here we must keep in mind that the stage only records the last and the second last words in the derivation. This happens because *3-gram* language model carries sensitivity only to the last two words of the sequence. This is the reason that state records only the last two words.

We define the initial state as $q_0 = (*, *, 0^n, 0, 0)$ where (0^n) is bit-string of length n , with n zeroes. Here we have used the $(*)$ to introduce the special start symbol in the language model. The initial form of state has no translation for the words as all the bits were translated as (0). Even the value for r was (0) and the score α was (0).

⁵A problem is NP-hard if an algorithm that is used for solving it can be translated into one for solving any NP-problem which is also known as non-deterministic polynomial time. NP-hard therefore means at least as hard as any NP-problem, although it might be harder indeed.

⁶Recall that n is the length of the source language sentence.

Next we are going to define the function $ph(q)$ that adjusts the state q to the set of those phrases that can be appended to the q . In order to make the phrase p a member of $ph(q)$, where $q = (e_1, e_2, b, r, \alpha)$, we must satisfy the following conditions:

- p must never come in clash with the bit-string b^7 .
- We must never violate the distortion limit⁸.

In addition, for any state q , and for any phrase $p \in ph(q)$, $next(q, p)$ is defined to be the state formed by combining state q with phrase p . Formally, if $q = (e_1, e_2, b, r, \alpha)$, and $p = (s, t, \epsilon_1, \dots, \epsilon_M)$, then $next(q, p)$ is the state $q' = (e'_1, e'_2, b', r', \alpha')$ defined as follows:

- Define $\epsilon_{-1} = e_1$ and $\epsilon_0 = e_2$.
- Define $e'_1 = \epsilon_{M-1}$ and $e'_2 = \epsilon_M$.
- Define $b'_i = 1$ for $i \in \{s\dots t\}$, and $b'_i = b_i$ for $i \notin \{s\dots t\}$.
- Define $r' = t$.
- Define $\alpha' = \alpha + g(p) + \sum_{i=1}^M \log q(\epsilon_i | \epsilon_{i-2}, \epsilon_{i-1}) + \eta \times |r + 1 - s|$.

Hence e'_1 and e'_2 are used to record the last and the second last word in the translation which is basically formed by the appending phrase p to state q . b' is updated to be a bit-string, this modification is mainly done to record the fact that words $s\dots t$ have been translated. When we take a look at r' then it is simply set to t , i.e., the end point of the phrase p ; and α' is calculated by adding the phrase score $g(p)$, the language model scores for the words $\epsilon_1\dots\epsilon_M$, and the distortion term $\eta \times |r + 1 - s|$.

The final function that is required for the purpose of decoding algorithm is a simple function that tests the equality of both the states. This is the function: $eq(q, q')$, and the main idea of this function is to return true or false. Assuming $q = (e_1, e_2, b, r, \alpha)$, and $q' = (e'_1, e'_2, b', r', \alpha')$, $eq(q, q')$ is true if and only if $e_1 = e'_1$, $e_2 = e'_2$, $b = b'$, and $r = r'$.

3.3 Hierarchical Phrase-Based Translation Model

Hierarchical phrase-based translation model (Chiang, D., 2007) is considered to be one of the suitable translation models used for the purpose of improving SMT performance.

There is a difference between the Hierarchical model and the Classical model and that is differing terms of rule expressivity. These rules are allowed to carry one or more than one non-terminals. Each of these non-terminals must act as a variable that can be expanded into other expressions with the help of grammar. Here we must keep in mind that this expansion must take place in recursive manner. These grammars are basically organized by a Synchronous Context-Free Grammar (SCFG) which captures long-range dependencies such as syntactic information. This information plays an important role in generating the correct translation between target languages and the source.

The intuitive concept behind the hierarchical phrase-based translation model is to maintaining gaps in the phrases so that translation units used in the classical

⁷For instance, we must have $b_i = 0$ for $i \in \{s(p)\dots t(p)\}$.

⁸For instance, we must have $|r + 1 - s(p)| \leq d$ where d is considered to be the distortion limit.

phrase-based model could be generalized. Basic unit of translation in the classical model is a phrase while hierarchical model is famous for bringing the sub-phrases into being. These sub-phrases are then used to remove any sort of problems that are associated with the standard phrase-based translation model. This model is meant to capture the long-range dependencies as they are crucial to generate a correct translation.

3.3.1 Hierarchical Rules

The standard classical phrase-based models show a nice performance for the translations that are directly linked to the sub-strings plus they have been observed carefully from the start in the training dataset. It must also be kept in the mind that learning the phrases that carry more than three words hardly bring any improvement to the performance because infrequency might be caused in such phrases due to data sparsity. So instead of going this way, it is best to opt for the natural way. The natural way encourages us to learn some of the grammatical rules along with the small phrases and then blend them together to create a translation.

There are also some other phrase-based models that introduce the reordering as a distortion which is independent of their content. However, this would be similar to having a blindfolded duel with our opponent. The general rule is that every reordering must contain the use of context.

All these issues are resolved by the hierarchical phrase-based model. It is considered one step above the phrase-based translation model as it contains sub-phrases that allow for the natural movement of the sub-phrases and learning of grammar rules. The translation system uses the parallel data to learn these rules without any sort of syntactic annotation. The system uses the syntax-based translation machine to adopt the technology. However, it also presents a challenging problem in the shape of hierarchical phrases.

3.3.2 Grammars Definition

Hierarchical models are basically inspired from the SCFGs formalism that helps in generating the source and target sentences through rewriting the non-terminals in a successive way.

If we consider the Hierarchical phrase-based model then we will know that grammar G is considered to be a special case of SCFG is basically defined as a 4-tuple: $G = (T, N, R, R_g)$. At this point, T is considered to be set of terminals while N is considered to be a set of non-terminals in G . Commonly, there are two types of non-terminals used by the hierarchical model's grammar i.e. X and S . At this point, S stands for the special start symbol while R denotes the set of production rules of the form.

$$X \rightarrow \langle \gamma, \alpha, \sim \rangle, \gamma, \alpha \in \{X \cup T^+\} \quad (3.17)$$

In this formula, γ and α denote the sequences of the terminals and non-terminals in source and target sides. The \sim stands for the alignment of the non-terminals in the source and target sides in such a way that non-terminal pair that has been co-indexed is rewritten synchronously. All of these production rules are blended together in order to derive the top symbol S . This top symbol S is derived through the use of rules R_g . There are two types of glue rules used by the hierarchical model; one is $S \rightarrow \langle X_1, X_1 \rangle$ and the other is $S \rightarrow \langle S_1 X_2, S_1 X_2 \rangle$.

Again at this point, non-terminal indicators tell us that target non-terminals and synchronous rewriting of the source have the identical index. The second rule points out that by connecting the smaller spans, one can translate the longer ones.

3.3.3 Rules Extraction

Heuristic approach is used by the Hierarchical phrased-based models to extract the rules from the phrase-pairs. The training of the hierarchical models share the similar initial steps as shared by the Classical phrase-based models. They begin from the word alignments and move towards the generation of the aligned phrase pairs.

If a parallel text is provided then the training will first obtain source-target and target-source alignments with the help of an aligner, such as *GIZA++* or *fast-align*. Later on, the bidirectional alignments are symmetrized with the help of heuristic alignment strategies including intersection or union (Och F. J. and H. Ney, 2003). At the end, it extracts the aligned phrase-pairs by seeking help from alignment template approach (Och F. J. and H. Ney, 2004). It happens in such a way that extracted phrase-pairs become parallel to the word alignments.

In simple words, it can also be said that phrase pairs are coerced by the source-target alignments in such a way that all the alignment links belonging to the source (target) words connect to the target (source) words within the phrase.

For the sake of argument, we must consider the example of word-aligned sentence pair (s_1^J, t_1^I, A) . In this example, s_1^J and t_1^I stand for the source and target sentences of length while J and I are linked to the word alignment A . Source-target sequence pair $(s_i^j, t_{i'}^{j'})$ can be termed as a phrase-pair if the below mentioned alignment constraints are satisfied:

- $(k, k') \in A$ where $k \in [i, j]$ and $k' \in [i', j']$
- $(k, k') \notin A$ where $k \in [i, j]$ and $k' \notin [i', j']$
- $(k, k') \notin A$ where $k \notin [i, j]$ and $k' \in [i', j']$

Commonly, the hierarchical models restrict their extraction to the tighter phrase pairs. This is done so that any phrase-pair carrying an unaligned boundary word could be ignored. This is the strategy to control the number of extracted hierarchical model's rules. Otherwise, they would have been extremely higher. The tighter phrase-pairs constraint can be written as:

- $(k, k') \in A$ where $k = i$ and $k' = [i', j']$
- $(k, k') \in A$ where $k = j$ and $k' = [i', j']$
- $(k, k') \in A$ where $k = [i, j]$ and $k' = i'$
- $(k, k') \in A$ where $k = [i, j]$ and $k' = j'$

After extracting the initial phrase-pair, the heuristic algorithm used to extract the rules moves forward in the way mentioned as follows:

At first we must assume that for $x = \langle s_i^j, t_{i'}^{j'} \rangle$ to be an initial phrase-pair will be considered a rule $X \rightarrow \langle s_i^j, t_{i'}^{j'} \rangle$. Now we will assume that $x' = \langle s', t' \rangle$ will be a sub phrase of the previous rule for instance, $s_i^j = s_p s' s_s$ and $t_{i'}^{j'} = t_p t' t_s$. It further creates

a new rule where it introduces the non-terminal X in both source and target sides by covering the spans of the sub-phrase x' while applying the following rule:

$$X \rightarrow \langle s_p X_1 s_s, t_p X_1 t_s \rangle \quad (3.18)$$

Here we must keep in mind that, non-terminals on the right-side of the rule are co-indexed. Being co-indexed allows them to rewrite in a synchronous manner.

The hierarchical model reduces the decoding complexity and limits the grammar's size by imposing several constraints on the extracted rules. These extracted rules are filtered for the purpose of removing the rules that violate any of the constraints mentioned below:

- Initial phrase-pairs must not carry any sort of unaligned word in the source or target phrase boundaries.
- The bi-phrases can keep maximum ten words on each side (it must also be kept in mind that extracted rules are limited to five tokens on the source side).
- Rules can possess maximum of the two non-terminals.
- Adjacent non-terminals will not be allowed in the source side (this helps in avoiding the spurious ambiguities during the process of decoding which is basically characterized due to the same translation yield that carries identical values for the feature functions).
- The rule must be lexicalized with the help of at least one aligned source-target word pair (this will ensure that lexical evidence backs the translation rule).

In simple words, rule extraction procedure can be listed in two main steps:

1. Identify initial phrase pairs using the same criterion as most phrase-based systems, namely, there must be at least one word inside one phrase aligned to a word inside the other, but no word inside one phrase can be aligned to a word outside the other phrase.
2. In order to obtain rules from the phrases, they look for phrases that contain other phrases and replace the sub-phrases with non-terminal symbols.

3.3.4 Rule Parameters Learning

In order to use the extracted grammar for the purpose of decoding, we need to learn about the rule parameters for instance, conditional translation probabilities $P(t|s)$ and $P(s|t)$. Each sentence pair in the corpus could be obtained with the help of several derivations which are not usually kept in view. Due to the fact that maximum likelihood estimates of rule frequencies could not be determined, Chiang, D. (2007) uses heuristics to estimate a rule distribution.

It assumes a unit count for each phrase-pair which is then distributed equally according to the rules that are derived from the phrase-pair. Later on, training corpus is used in order to determine an aggregate of the rule counts $c(s, t)$ across all the phrase pairs. Afterwards, the conditional translation probabilities $P(t|s)$ and $P(s|t)$ are determined by relative frequency estimation (weight) of the counts.

Bod in 1998, presented an original proposal for the heuristic estimator to serve the purpose of data oriented parsing (DOP) so that it could gather the estimate of

probabilistic tree substitution grammars (PTSG) for parsing. Later on, this was also used for the classical phrase-based translation model (Och F. J. and H. Ney, 2004). It was also successfully used in several SMT models (Quirk et al., 2005; Galley et al., 2006b), including hierarchical phrase-based translation model.

3.3.5 Standard Features

If we follow the standard classical phrase-based translation model, it becomes clear that hierarchical method uses the log-linear model (Och and Ney, 2002) for the purpose of translation. Under this scenario, the probability of derivation can be written in terms of different feature functions ϕ as:

$$P(d) \propto \prod_{i=1}^k \phi_i^{w_i} \quad (3.19)$$

Where k represents the total number of features and w denote the weights of the feature functions. Hierarchical model uses the following standard feature functions:

- Conditional translation probabilities $P(t|s)$ and $P(s|t)$
- Conditional lexical weights $P_{lex}(t|s)$ and $P_{lex}(s|t)$
- Phrase penalty
- Word penalty
- Rule weight
- Language model

3.3.6 Hierarchical Model Decoding

Hierarchical phrase-based model uses a *CKY-style* algorithm (Cocke, 1969; Kasami, 1966; Younger, 1967) for the purpose of decoding. If provided with a source sentence s , the decoder finds the target side yield t_{best} of the best scoring derivation obtained by applying rules in the SCFG:

$$\hat{t} = t_{best} \left(\arg \max_{d \in D(s)} P(d) \right) \quad (3.20)$$

At this point, $D(s)$ is considered to be the set of derivations that is attained from the learned grammar for the source sentence s . The decoder parses the source sentence by seeking refuge in a modified version of CKY parser. During this entire procedure, the target side of the corresponding derivations simultaneously produces the candidate translations. Furthermore, the rule parameters and other features are used for the purpose of scoring the derivations along with the language model score of the target translation as shown in Equation (3.16).

The derivation initiates from the moment when leaf cells of the CKY chart correspond to the source side tokens and moves towards bottom-up. Decoder identifies the applicable rules and analogous to monolingual parsing in order to account for each cell in the CKY chart. While following these rules, the non-terminals should have corresponding entries in the respective antecedent cells. The target side of the production rules produces the translation that is used for the source span and the translations in the top-most cell. This translation corresponds to the entire sentence.

The log-linear model over derivations $P(d)$ can be factorized to separate the language model (LM) feature from other features. The LM feature scores the target yield as $P_{lm}(t)$ usually with an n -gram model trained separately. The model can be written by factorizing derivation d into its component rules R_d as below:

$$P(d) \propto \left(\prod_{i=1}^{k-1} \prod_{r \in R_d} \phi_i(r)^{w_i} \right) P_{lm}(t)^{w_{lm}} \quad (3.21)$$

Where w_i is the corresponding weight of the feature ϕ_i . The feature weights w_i are optimized by minimizing a loss (Aho and Ullman, 1969) or by comparing pairwise rankings (Hopkins and May, 2011).

3.4 Experimental Framework

In this section, we provide baseline translation systems for three pairs of languages. The three baseline systems are based on the Classical phrase-based translation model, and the other three baseline systems are based on the Hierarchical phrase-based translation model.

Even though, there various differences between Hierarchical and Classical phrase-based translation models, still the pipelines for training and testing are comparatively similar. This is the reason that we extend the *Moses* (based on Classical model) and *Cdec* (based on Hierarchical model) open-source translation engines, so that aforementioned popular translation models could be implemented. We use these open-source tool-kits due to following reasons:

- Offering support to the linguistically motivated factors.
- Getting relief from the confusing network decoding.
- Possessing efficient data formats for translation and language models.

In addition to the above mentioned reasons, the mentioned translation engines also provide us with the options of wide variety of tools that could be used for training, tuning and applying the system to various other translation tasks. (Ahmadnia and Serrano, 2015)

Due to the fact that it is an open-source phrase-based tool-kit, *Moses* (Koehn et al., 2007), can be used for the purpose of training statistical models of text translation from a source language to a target one. It allows new source text to be decoded with the help of these models so that automatic translations in the target language could be produced.

The basic requirement for the training is parallel corpus of passages in the two languages. These must be translated sentence pairs in a typical manual way. Translations probabilities could be calculated with the help of a sub-sample of occurrences of the concerned source phrase. Due to the fact that system accesses the target language corpus and word alignment data, phrase translations and their model parameters can be determined at run-time (Lopez, 2008).

Moses platform is mainly used because it offers us to automatically train translation models for the considered language pair. Once we get a trained model, an efficient search algorithm manages to find the highest probability translation among the exponential number of choices in the quickest time possible.

Cdec (Dyer et al., 2010), is commonly used for the SMT and similar structured prediction models. It is considered to be a decoder, aligner, and learning framework. The main feature of *Cdec* is that it uses a single unified internal representation for the purpose of translation forests. The decoder also separates model-specific translation logic from general re-scoring, pruning, and inference algorithms on strict basis. This unified representation teaches us that decoder can extract not only the *1-best* or *k-best* translations, but it can also assist with the alignments to a reference. Further it can extract the quantities necessary to drive discriminative training that make use of the gradient-based or gradient-free optimization techniques.

Cdec platform is used because of its mature and advanced nature. It is a mature software platform for research in development of translation models and algorithms. The architecture of this platform was created while keeping the machine learning and algorithmic research use-cases in mind. It has specifically been designed to work efficiently in both limited resource environments and large cluster environments.

We conducted experiments with Hierarchical translation models while using the *Cdec* translation engine. These experiments also included a range of corpora sizes. Later on, we compare the results with Classical phrase-based models while using *Moses* translation engine with the same corpora.

The evaluation metric that we choose for these experiments is BLEU (Papineni et al., 2002). We interpreted the higher scores through BLUE for the comparison of the translation systems.

3.4.1 Experimental Set-Up

We evaluated three language pairs for the purpose of this experiment i.e. *Spanish-English*, *English-Persian*, and *Persian-Spanish*. In all the cases backward translation direction has also been applied.

For the purpose of determining the best possible results, a statistical language model requires an extremely large amount of data in target language. It also needs to be trained in order to obtain proper probabilities. In this portion of experiments we apply *4-gram* language models.

For all sets of experiments we selected (500,000) parallel sentences for training step, (30,000) parallel sentences for tuning step, and (50,000) parallel sentences for testing step. All these parallel sentences were selected from *Open-Subtitles*⁹ parallel corpus (Tiedemann, 2012).

In addition, for the third mentioned language pair, we applied the test with an other parallel corpus called *Tanzil*¹⁰ (Tiedemann, 2012). For this experiment we selected (50,000) parallel sentences for the step of training, (5,000) parallel sentences for the step of tuning, and (10,000) parallel sentences for the step of testing.

Our *Moses* and *Cdec* translation tool-kits are trained in identical conditions. The *Open-Subtitles* collection of parallel corpora has been compiled from a large database of movie and TV subtitles. It includes a total of (1689) bilingual texts, spanning (2.6) billion sentences spread across (60) languages. This corpus also blends a number of enhancements in the preprocessing and alignment of the subtitles, such as:

- The automatic correction of OCR¹¹ errors.

⁹[http://opus.lingfil.uu.se/Open-Subtitles 2013.php](http://opus.lingfil.uu.se/Open-Subtitles%2013.php)

¹⁰<http://opus.lingfil.uu.se/Tanzil.php>

¹¹Optical Character Recognition is the process of converting scanned images of letters and words into a electronic versions.

- The use of meta-data to estimate the quality of each subtitle and score subtitle pairs.

As can be seen in the Table 3.1, we used the same amounts of *Open-Subtitles* parallel corpus for training both the translation model and the language model.

TABLE 3.1: Open-Subtitle corpus statistics.

| Directions | Spanish/English | English/Persian | Persian/Spanish |
|---------------|-----------------|-----------------|-----------------|
| Training step | 500,000 | 500,000 | 500,000 |
| English words | 3,245,701 | 3,262,827 | — |
| Spanish words | 3,018,988 | — | 3,020,347 |
| Persian words | — | 3,183,056 | 3,342,977 |
| Tuning step | 30,000 | 30,000 | 30,000 |
| English words | 185,017 | 184,002 | — |
| Spanish words | 177,514 | — | 176,820 |
| Persian words | — | 185,533 | 191,904 |
| Testing step | 50,000 | 50,000 | 50,000 |
| English words | 309,863 | 312,174 | — |
| Spanish words | 306,762 | — | 304,496 |
| Persian words | — | 308,033 | 332,352 |

Tanzil is a collection of *Quran* translations compiled by the Tanzil project as the other parallel corpus is used for another test between *Persian* and *Spanish* language pair. We also used the same amounts of this parallel corpus for training both the translation model and the language model.

TABLE 3.2: Tanzil corpus statistics.

| Steps | Training | Tuning | Testing |
|---------------|-----------|---------|---------|
| Sentences | 50,000 | 5,000 | 10,000 |
| Persian words | 1,624,002 | 110,327 | 254,816 |
| Spanish words | 1,060,829 | 116,249 | 218,844 |

3.4.2 Implementation

For the purpose of Moses-based experiments in both directions of every single translation task, we set the menu of *Moses* decoder as a suitable Classical phrase-based platform. We set the beam-size to (300), the distortion limit to (6), and the number of target phrases is limited to (10) for each source phrase, also the MERT iterations is set to (20). During our Cdec-based experiments we use the *Cdec* implementation of the Hierarchical phrase-based algorithm. Our maximum phrase length is adjusted to (10), and the Margin In-fused Relaxed Algorithm (MIRA) iterations is set to (20). However, the size of *k-best* list is determined at (300).

The language models that are used in all of the Moses-based and Cdec-based experiments are considered at *4-gram* models. They are smooth with a modified *Kneser-Ney* algorithm (Pickhardt et al., 2014) which is implemented in *KenLM* (Heafield,

2011), *SRILM* (Stolcke, 2002), and *IRSTLM* (Federico et al., 2008).

The issue of word alignment in the parallel corpus is quite critical that is why it needs attention. Sentence-aligned parallel corpora is considered to be useful for the application of machine learning to MT. However, due to some unfortunate events, parallel corpora does not originate in this form. It was also determined that there was a great shortage (comparatively) of bilingual text for *Persian-Spanish* translation. That is why; great measures were taken in order to ensure that the text that was available carried the best possible quality.

Several different methods were also used to perform alignment. Desirable characteristics of an efficient sentence alignment method included the speed, accuracy and no need for prior knowledge of the corpus or the languages in the pair. In our experiments we conducted use of the *fast-align* because it is a simple and fast alignment tool (Dyer et al., 2013) comparing with the other tool-kits.

3.4.3 Results Analysis and Evaluation

After implementing, we discuss the results which we achieved, and compare *Moses* and *Cdec* performances over our translation systems through different *4-gram* language models. We trained the translation machine on three pairs of languages in six directions. For the evaluation, we report the results using the BLEU evaluation metric. We start by comparing the translations yielding the best configuration generated by both *Cdec* and *Moses* translation engines.

In the first set of experiments, we applied *Mses* and *Cdec* translation engines for the *Spanish-English* translation direction and back translation under three kinds of *4-gram* language models. Table 3.3 shows the results.

TABLE 3.3: The performance comparison of Classical (*Moses*) and Hierarchical (*Cdec*) translation models for Spanish-English translation task according to different 4-gram language models on Open-Subtitles parallel corpus using BLEU scores.

| 4-gram language models | IRSTLM | SRILM | KenLM |
|------------------------|--------|-------|-------|
| Moses–Classical | 25.65 | 25.96 | 26.19 |
| Cdec–Hierarchical | 27.12 | 27.51 | 27.68 |

As seen in Table 3.3, with *KenLM*, the BLEU score for *Cdec* is (27.68), and for *Moses* is (26.19), while with *SRILM* and *IRSTLM*, *Cdec* allocated (27.51), and (27.12) BLEU points, and *Moses* reached at (25.96), and (25.65) BLEU points, respectively, in the mentioned translation direction. In all cases, the BLEU scores for *Cdec* show better results in comparison to *Moses* with all applied language models for *Spanish* to *English* translation task.

In the second set of experiments, for the back translation task (*English-Spanish*), we applied *Moses* and *Cdec* translation platforms as well. Table 3.4 illustrates that using *KenLM*, the BLEU score for *Moses* is (27.91), for *Cdec* is (26.45). Using *SRILM* and *IRSTLM*, *Moses* allocated (27.73), and (27.44) BLEU points, *Cdec* reached at (26.14), and (25.92) BLEU points, respectively. The BLEU scores for *Moses* show better performance in comparison to *Cdec* through all applied language models for *English* to *Spanish* translation task.

TABLE 3.4: The performance comparison of Classical (Moses) and Hierarchical (Cdec) translation models for English-Spanish translation task according to different 4-gram language models on Open-Subtitles parallel corpus using BLEU scores.

| 4-gram language models | IRSTLM | SRILM | KenLM |
|------------------------|--------|-------|-------|
| Moses-Classical | 27.44 | 27.73 | 27.91 |
| Cdec-Hierarchical | 25.92 | 26.14 | 26.45 |

Figure 3.3 illustrates the learning curve changes of *Moses* and *Cdec* translation systems using 4-gram language model of *Ken*, *SRI*, and *IRST* tool-kits for both *Spanish-English* and *English-Spanish* translation directions according to BLEU scores.

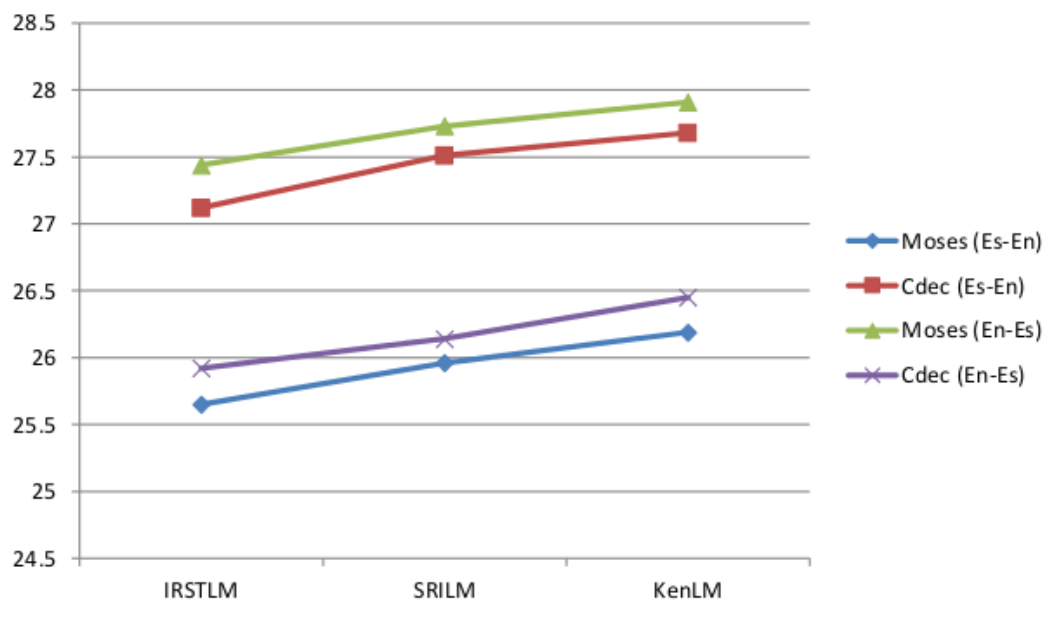


FIGURE 3.3: The learning curve of Classical (Moses) and Hierarchical (Cdec) phrase-based translation models for Spanish-English translation and vice versa using 4-gram language models on Open-subtitles parallel corpus according to BLEU scores.

According to this figure, two curves (the blue one and the red one) are related to the forward translation task (*Spanish-English*), and two curves (the green one and the purple one) are related to the backward translation task (*English-Spanish*).

The results shown in Tables 3.3 and 3.4, and also the changes of learning curve in Figure 3.3 prove that for *Spanish-English* translation direction Hierarchical phrase-based translation (*Cdec*) outperforms the Classical model, while for back translation task from *English* to *Spanish*, the Classical phrase-based translation model outperforms the Hierarchical one under applying a 4-gram of *Ken*, *SRI*, and *IRST* language models.

In the third and fourth sets of experiments, *English-Persian* and back translation tasks are applied under the same conditions between *Moses* and *Cdec*. Table 3.5 shows that the suitable translation model for *English-Persian* task is the Hierarchical mode, i.e. *Cdec* system achieved (24.06), (23.78), and (24.52) BLEU points applying *SRILM*, *IRSTLM*, and *KenLM*, respectively, while *Moses* scores (22.31) at

KenLM, (21.97) at *IRSTLM*, and (22.12) at *SRILM*. Table 3.6 shows that the Classical phrase-based translation model has better performance than the Hierarchical one. The BLEU scores of *Moses* for *IRST*, *SRI*, and *Ken* language models are (25.18), (25.35), and (25.61), respectively. On the other hand, the BLEU score for *Cdec* are (23.54), (23.75), and (23.97), for the same language models.

TABLE 3.5: The performance comparison of Classical (Moses) and Hierarchical (Cdec) translation models for English-Persian translation task according to different 4-gram language models on Open-Subtitles parallel corpus using BLEU scores.

| 4-gram language models | IRSTLM | SRILM | KenLM |
|------------------------|--------|-------|-------|
| Moses-Classical | 21.97 | 22.12 | 22.31 |
| Cdec-Hierarchical | 23.78 | 24.06 | 24.52 |

TABLE 3.6: The performance comparison of Classical (Moses) and Hierarchical (Cdec) translation models for Persian-English translation task according to different 4-gram language models on Open-Subtitles parallel corpus using BLEU scores.

| 4-gram language models | IRSTLM | SRILM | KenLM |
|------------------------|--------|-------|-------|
| Moses-Classical | 25.18 | 25.35 | 25.61 |
| Cdec-Hierarchical | 23.54 | 23.75 | 23.97 |

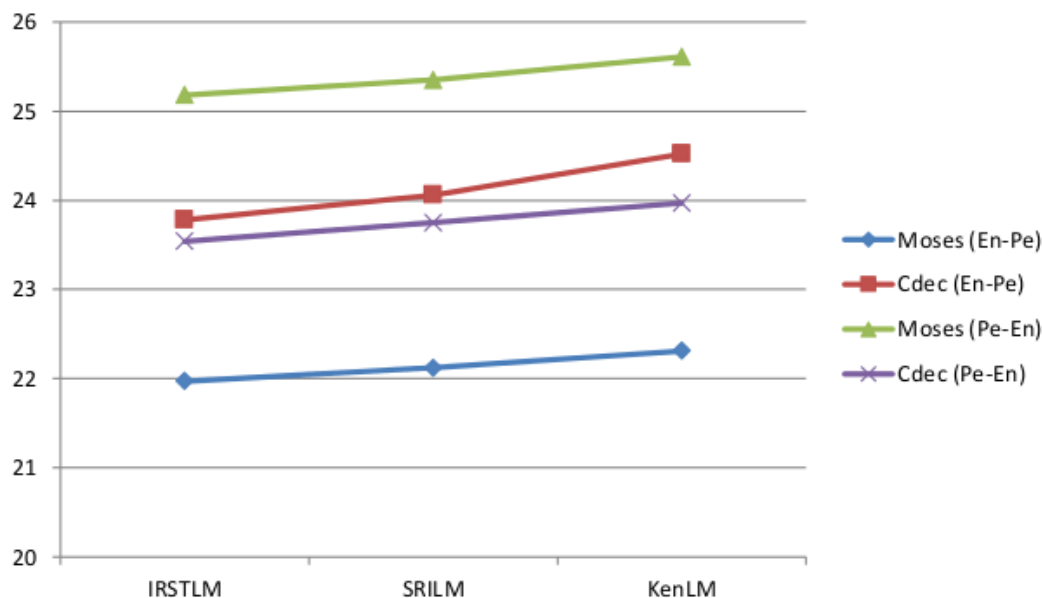


FIGURE 3.4: The learning curve of Classical (Moses) and Hierarchical (Cdec) phrase-based translation models for English-Persian translation and vice versa using 4-gram language models on Open-subtitles parallel corpus according to BLEU scores.

Figure 3.4 illustrates the learning curve changes of *Moses* and *Cdec* translation systems using 4-gram language model of *Ken*, *SRI*, and *IRST* tool-kits for both *English-Persian* and *Persian-English* translation directions according to BLEU scores. According to Figure 3.4, two curves (the blue one and the red one) are related to the forward translation task (*English-Persian*), and two curves (the green one and the purple one) are related to the backward translation task (*Persian-English*).

The results shown in Tables 3.5 and 3.6, and also the changes of learning curve in Figure 3.4 demonstrate that for *English-Persian* translation direction Hierarchical phrase-based translation (*Cdec*) outperforms the Classical model, while for back translation task from *Persian* to *English*, the Classical phrase-based translation model outperforms the Hierarchical one under applying a 4-gram of *Ken*, *SRI*, and *IRST* language models.

In the fifth experiment set, *Cdec* and *Moses* are applied for the *Persian-Spanish* translation task. Table 3.7 shows that by using the *KenLM*, the BLEU score for *Moses* is (23.48), while for *Cdec* the BLEU score is (22.77). When we apply the *SRILM*, the *Moses*-based system achieves (23.15) BLEU points, and the *Cdec*-based system achieves (22.50) points of BLEU. On the other hand, after applying the *IRSTLM*, the BLEU score for *Moses* is (22.89), while for *Cdec* is (22.16). The scores show that *Moses* as a Classical phrase-based system has better performance than *Cdec* as a Hierarchical one in *Persian-Spanish* translation task.

TABLE 3.7: The performance comparison of Classical (*Moses*) and Hierarchical (*Cdec*) translation models for Persian-Spanish translation task according to different 4-gram language models on Open-Subtitles parallel corpus using BLEU scores.

| 4-gram language models | IRSTLM | SRILM | KenLM |
|---------------------------|--------|-------|-------|
| <i>Moses</i> -Classical | 22.89 | 23.15 | 23.48 |
| <i>Cdec</i> -Hierarchical | 22.16 | 22.50 | 22.77 |

The sixth set of experiments is related to *Spanish-Persian* translation direction. In this experiment we apply Classical phrase-based translation system (*Moses*) as well as Hierarchical phrase-based platform (*Cdec*). As seen in Table 3.8, the BLEU scores for *Cdec* are (21.88), (21.65), and (21.39) applying *KenLM*, *SRILM*, and *IRSTLM* respectively. While for the mentioned language models, *Moses* allocated (21.02), (20.78), and (20.56) BLEU points respectively.

TABLE 3.8: The performance comparison of Classical (*Moses*) and Hierarchical (*Cdec*) translation models for Spanish-Persian translation task according to different 4-gram language models on Open-Subtitles parallel corpus using BLEU scores.

| 4-gram language models | IRSTLM | SRILM | KenLM |
|---------------------------|--------|-------|-------|
| <i>Moses</i> -Classical | 20.56 | 20.78 | 21.02 |
| <i>Cdec</i> -Hierarchical | 21.39 | 21.65 | 21.88 |

The scores for *Cdec* as a Hierarchical phrase-based translation engine show high-quality performances in comparison to *Moses* as a Classical one using three kinds of 4-gram language models.

Figure 3.5 shows the learning curve changes of *Moses* and *Cdec* translation systems using 4-gram language model of *Ken*, *SRI*, and *IRST* tool-kits for *Persian-Spanish* translation task as well as *Spanish-Persian* translation task according to BLEU scores. In this figure four curves are observed; two of them (the blue one and the red one) are related to the forward translation direction (*Persian-Spanish*), and two of them (the green one and the purple one) are related to the backward translation direction (*Spanish-Persian*).

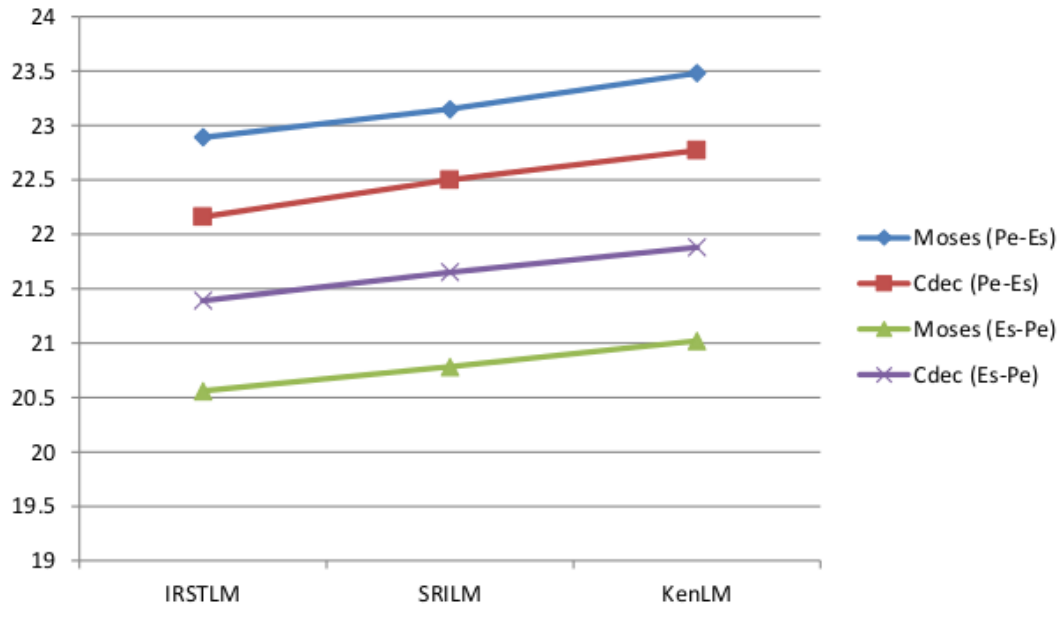


FIGURE 3.5: The learning curve of Classical (Moses) and Hierarchical (Cdec) phrase-based translation models for Persian-Spanish translation and vice versa using 4-gram language models on Open-subtitles parallel corpus according to BLEU scores.

In the separate sets of experiments, we applied the 4-gram language model through three kinds of tool-kits using both Classical and Hierarchical translation systems with different domain of parallel corpus (*Tanzil*) just in order to evaluate the *Persian-Spanish* translation task as well as the back translation.

Table 3.9 and 3.10 illustrates similar results as Tables 3.7 and 3.8.

TABLE 3.9: The performance comparison of Classical (Moses) and Hierarchical (Cdec) translation models for Persian-Spanish translation task according to different 4-gram language models on *Tanzil* parallel corpus using BLEU scores.

| 4-gram language models | IRSTLM | SRILM | KenLM |
|------------------------|--------|-------|-------|
| Moses-Classical | 21.02 | 21.28 | 21.79 |
| Cdec-Hierarchical | 19.96 | 20.21 | 20.56 |

The BLEU scores for the forward translation direction in *Moses* as the Classical phrase-based translation system, applying the *KenLM*, *SIRLM*, and *IRSTLM* tool-kits, are (21.79), (21.28), and (21.02), respectively, while these scores for the same

translation engine are (20.34), (20.11), and (19.93) in the backward translation direction. However, in the forward translation task after applying the *KenLM*, *SRILM*, and *IRSTLM* tool-kits separately on the systems, the BLEU score (20.56), (20.21), and (19.96) in *Cdec* as the Hierarchical phrase-based translation system, while this metric show the scores (22.32), (22.08), and (21.77) for the same platform in the back translation respectively.

TABLE 3.10: The performance comparison of Classical (Moses) and Hierarchical (Cdec) translation models for Spanish-Persian translation task according to different 4-gram language models on Tanzil parallel corpus using BLEU scores.

| 4-gram language models | IRSTLM | SRILM | KenLM |
|------------------------|--------|-------|-------|
| Moses-Classical | 19.93 | 20.11 | 20.34 |
| Cdec-Hierarchical | 21.77 | 22.08 | 22.32 |

According to the results shown by Tables 3.7, 3.8, 3.9, and 3.10, for *Persian to Spanish* translation task *Moses* as a Classical phrase-based system outperforms the Hierarchical one. However, for *Spanish to Persian* translation direction, *Cdec* as a Hierarchical phrase-based translation system still has a better performance than the *Moses* as the Classical one.

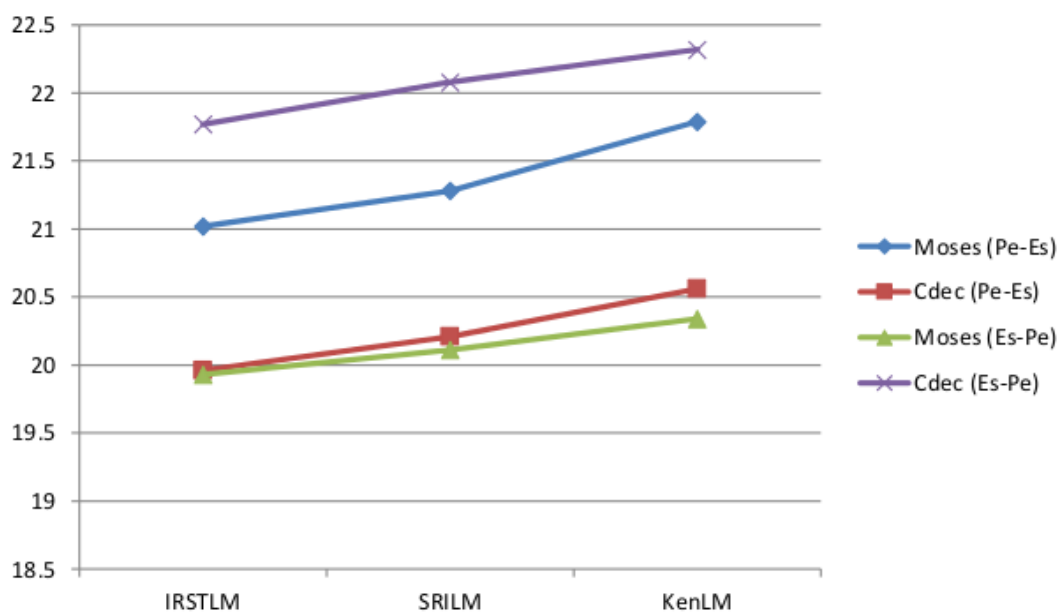


FIGURE 3.6: The learning curve of Classical and Hierarchical phrase-based translation models for Spanish-English translation and vice versa using 4-gram language models on Tanzil parallel corpus according to BLEU scores.

Figure 3.6 provides the learning curve changes of *Moses* translation system as well as *Cdec* one using 4-gram language model of *Ken*, *SRI*, and *IRST* tool-kits for both *Persian-Spanish* and *Spanish-Persian* translation directions according to BLEU scores. In this figure the blue curve and the red one are related to *Persian-Spanish* translation task, while the green curve and the purple one are related to *Spanish-Persian* translation task.

3.5 Comparative Performance of Phrase-Based Models

In order to set an appropriate aim of SMT research for our considered translation tasks, the comparative performance of suitability and potential between the Classical phrase-based translation model and the Hierarchical one becomes an initial problem.

In this section, based on standard settings, we employ the different surrounding words i.e. *3-gram* and *5-gram* language models through all *Ken*, *SRI*, and *IRST* toolkits on our considered translations, and with investigation and comparison of the impact of different statistical language models, we can understand that "how they could incur affects on a translation result?".

KenLM estimates unpruned language models with modified *Kneser-Ney* smoothing. The builder is disk-based; specify the amount of RAM to use and it performs disk-based merge sort when necessary. It is faster than *SRILM* and *IRSTLM* and scales to much larger models. *IRSTLM* can scale but, to do so, it approximates modified *Kneser-Ney* smoothing. The latest version of *BerkeleyLM* (Pauls and Klein, 2011) does not implement interpolated modified *Kneser-Ney* smoothing, but rather implements absolute discounting without support for interpolation or modified discounting. *SRILM* uses memory to the point that building large language models is infeasible.

We gain results that in *Spanish-English* translation task *5-gram* Hierarchical model is preferable, while in the task of *English-Spanish* translation *5-gram* Classical model is preferable.

In *English-Persian* translation direction *5-gram* Hierarchical system works well, while in *Persian-English* *5-gram* Classical system has better performance than the rests.

In the direction of *Persian* to *Spanish* translation task, *5-gram* Classical model outperforms the rests, while in the back direction from *Spanish* to *Persian* translation task, *5-gram* Hierarchical model gains a better BLEU point over the others, independent on the kind of selected language models.

The results from these performances comparison will be used in order to set as state-of-the-art so further researches could be conducted on SMT.

3.5.1 Experiments Setting

For the *Spanish↔English* experiments, a collection of *Europarl* (Koehn, P., 2005) corpus, and for the *English↔Persian* experiments a subset of *TEP* corpus (Pilevar et al., 2011) have been used. The former collection consists of (*500K*) parallel sentences for training step, (*50K*) parallel sentences for tuning step, and (*100K*) parallel sentences for the testing one, while the latter collection consists of (*400K*) parallel sentences as the training data, (*30K*) parallel sentences as the developing set, and (*70K*) parallel sentences as the testing set.

To experiment the *Persian-Spanish* translation and vice versa, we collected a parallel corpus gathered from both *Open-Subtitle* and *Tanzil* corpora (Tiedemann, 2012). The former and latter consist of (*100K*) and (*50K*) *Persian-Spanish* sentence pairs, respectively. In total we gain a parallel corpus between *Persian* and *Spanish* with approximately (*150K*) sentence pairs. We manually selected (*5K*) sentence pairs as a

development set and randomly select (10K) sentence pairs as a test set. The (135K) remaining sentence pairs were applied as a training set.

It is our objective to compare the quality of forward and backward translation performance in each translation task while we are using both the Classical and Hierarchical phrase-based systems. In order to generate the 3-gram and 5-gram language models of *Spanish*, *English*, and *Persian*, we adopt the strategy of exploiting the *Ken*, *SRI*, and *IRST* language model tool-kits.

The selection of *Moses* is done as well as *Cdec* so they can perform their function on the phrase/rule extraction, phrase/rule-table, generation, and decoding. On the other hand, the MERT function is done as well as MIRA function so that feature weights of both the models could be tuned. The outcome of *Moses* and *Cdec* for the Classical and Hierarchical phrase-based models emerged in the form of phrase-table and rule-table, respectively.

The difference between both of the tables is that Hierarchical rule-table includes translations of both terminal and non-terminal nodes so that hierarchy could be clarified whereas, Classical phrase-table provides information about translation pairs of phrase including the word order.

3.5.2 Results

We have conducted the evaluation using BLEU metric (Papineni et al., 2002) for the systems including both of our considered translation directions. The evaluation involves 3-gram Classical model (3-gram C), and 5-gram Classical model (5-gram C), as well as 3-gram Hierarchical model (3-gram H), and 5-gram Hierarchical model (5-gram H). In our experiments all the results of the Classical models are based on *Moses* platform, while the output results of the Hierarchical models are based on *Cdec* platform.

Table 3.11 shows the *Spanish-English* translation results according to BLEU scores for comparing the performance of 3-grams and 5-grams language models using *KenLM*, *SRILM*, and *IRSTLM* through *Moses* as Classical-based, and *Cdec* as Hierarchical-based systems.

TABLE 3.11: The performance of applying 3-gram and 5-gram language models through Ken, SRI, and IRST tool-kits on Spanish-English translation task using BLEU metric.

| Language models | 3-gram C | 5-gram C | 3-gram H | 5-gram H |
|-----------------|----------|----------|----------|----------|
| KenLM | 26.14 | 26.24 | 27.63 | 27.71 |
| SRILM | 25.92 | 26.06 | 27.40 | 127.57 |
| IRSTLM | 25.61 | 25.72 | 27.08 | 27.14 |

According to the results shown in Table 3.11, the Hierarchical phrase-based translation model is preferable for the translation process from *Spanish* language to *English* language under the SMT platform. In all cases, independent of the kind of

n -grams language models, *Cdec* as a Hierarchical translation system has better performance than *Moses* as a Classical translation system.

Figure 3.7 demonstrates the performance chart of both Classical (*Moses*) and Hierarchical (*Cdec*) phrase-based translation systems in *Spanish-English* translation task. The chart shows that the 5-gram and 3-gram Hierarchical phrase-based models (the Purple and Green bars respectively) outperform the Classical models under applying all language modelling tool-kits.

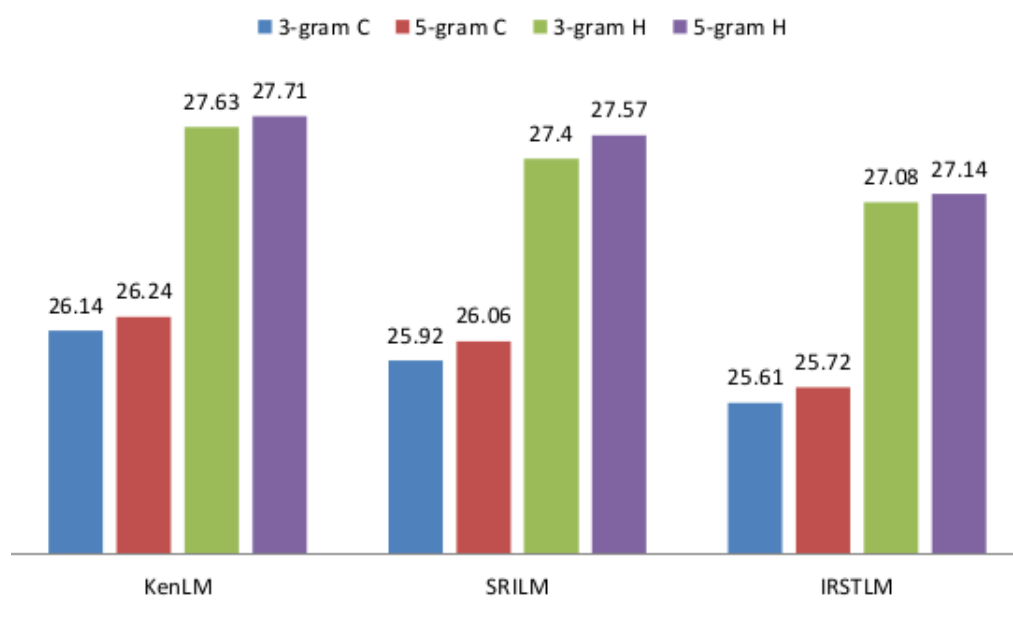


FIGURE 3.7: The performance chart of Spanish-English translation task in different kinds of language models according to BLEU.

The results of *English-Spanish* translation have been shown in Table 3.12. These results are also based on BLEU scores and the translation systems are under 3-grams and 5-grams language models applying *KenLM*, *SRILM*, and *IRSTLM*.

TABLE 3.12: The performance of applying 3-gram and 5-gram language models through Ken, SRI, and IRST tool-kits on English-Spanish translation task using BLEU metric.

| Language models | 3-gram C | 5-gram C | 3-gram H | 5-gram H |
|-----------------|----------|----------|----------|----------|
| KenLM | 27.84 | 27.97 | 26.42 | 26.51 |
| SRILM | 27.68 | 27.45 | 26.07 | 26.19 |
| IRSTLM | 27.39 | 27.49 | 25.88 | 25.94 |

According to the results shown in Table 3.12, for implementing the translation process from *English* language to *Spanish* language, the Classical phrase-based translation model outperforms the Hierarchical phrase-based translation model under the SMT platform. In all cases, independent of the kind of n -grams language models, *Moses* as a Classical phrase-based translation system has better performance than *Cdec* as a Hierarchical one.

Figure 3.8 illustrates the performance chart of both Classical (Moses) and Hierarchical (Cdec) phrase-based translation systems in *English-Spanish* translation task. The chart shows that the 5-gram and 3-gram Classical phrase-based models (the Red and Blue bars respectively) outperform the Hierarchical models under applying all language modelling tool-kits.

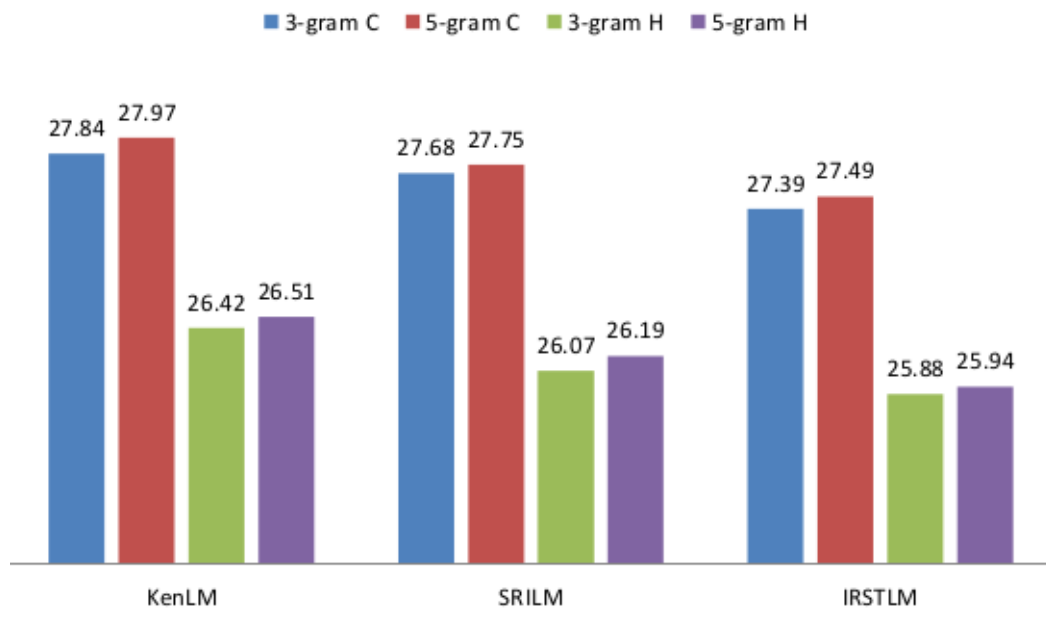


FIGURE 3.8: The performance chart of English-Spanish translation task in different kinds of language models according to BLEU.

Table 3.13 shows the results based on BLEU scores of *English-Persian* translation task. In this set of experiments the translation systems are under 3-grams and 5-grams language models applying *KenLM*, *SRILM*, and *IRSTLM*, under *Moses* and *Cdec* translation engines.

TABLE 3.13: The performance of applying 3-gram and 5-gram language models through Ken, SRI, and IRST tool-kits on English-Persian translation task using BLEU metric.

| Language models | 3-gram C | 5-gram C | 3-gram H | 5-gram H |
|-----------------|----------|----------|----------|----------|
| KenLM | 22.29 | 22.34 | 24.48 | 24.59 |
| SRILM | 22.10 | 22.19 | 24.01 | 24.11 |
| IRSTLM | 21.95 | 22.02 | 23.75 | 23.84 |

According to these results, in the translation process from *English* language to *Persian* one, the Hierarchical phrase-based translation model outperforms the Classical phrase-based translation model under the SMT platform. In all cases, independent of the kind of *n-grams* language models, *Cdec* as a Hierarchical system has better performance than *Moses* as a Classical system.

Figure 3.9 shows the performance chart of both Classical (Moses) and Hierarchical (Cdec) phrase-based translation systems in *English-Persian* translation task. The

chart shows that the 5-gram and 3-gram Hierarchical phrase-based models (the Purple and Green bars respectively) have better performance than the Classical models under applying all language modelling tool-kits.

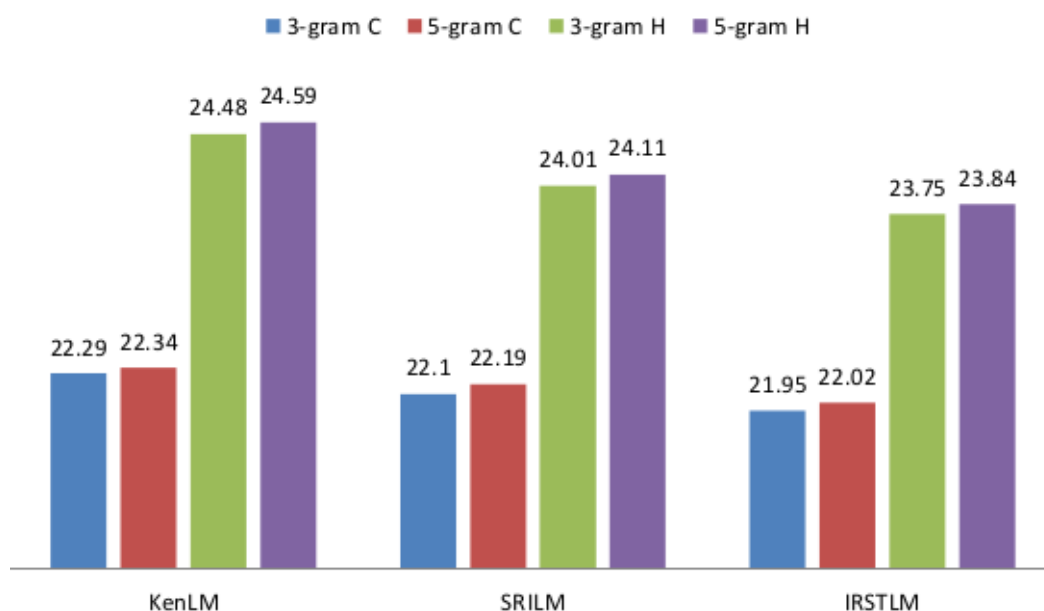


FIGURE 3.9: The performance chart of English-Spanish translation task in different kinds of language models according to BLEU.

The results of translation process from *Persian* to *English* based on the scores of BLEU metric have been shown in Table 3.14. Our translation systems; *Moses* and *Cdec* are under 3-grams and 5-grams language models applying *KenLM*, *SRILM*, and *IRSTLM*.

TABLE 3.14: The performance of applying 3-gram and 5-gram language models through Ken, SRI, and IRST tool-kits on Persian-English translation task using BLEU metric.

| Language models | 3-gram C | 5-gram C | 3-gram H | 5-gram H |
|-----------------|----------|----------|----------|----------|
| KenLM | 25.54 | 25.66 | 23.95 | 24.02 |
| SRILM | 25.29 | 25.42 | 23.74 | 23.78 |
| IRSTLM | 25.10 | 25.25 | 23.51 | 23.58 |

According to the results, in the *Persian-English* translation, the Classical phrase-based translation models work better than the Hierarchical phrase-based translation models under the SMT platform. In all cases, independent of the kind of n -grams language models, the Classical-based system (*Moses*) has better performance than the Hierarchical-based system (*Cdec*).

The performance chart of both Classical (*Moses*) and Hierarchical (*Cdec*) phrase-based translation systems in *Persian-English* translation direction has been shown in Figure 3.10. This chart shows that the Classical 5-gram and 3-gram models (the Red and Blue bars respectively) have better performance than the Hierarchical models under applying all language modelling tool-kits.

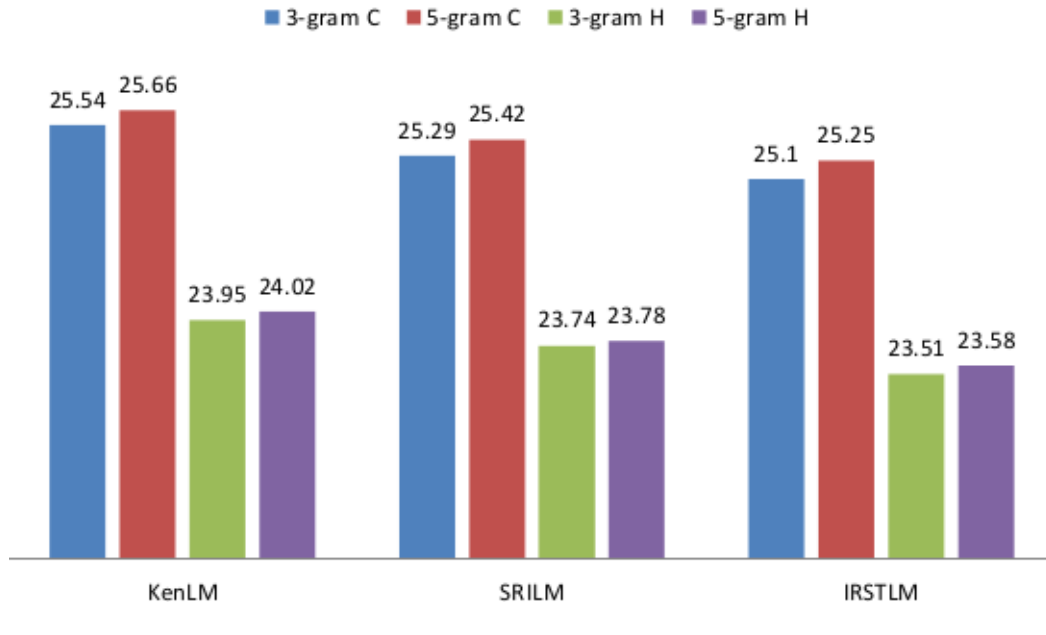


FIGURE 3.10: The performance chart of English-Spanish translation task in different kinds of language models according to BLEU.

Now, we focus on the experimental framework of *Persian*↔*Spanish* translation tasks, and investigate the impact of different statistical language models on the translation systems' performance.

The experimental results showed in the Tables 3.15 demonstrate the accuracy of translation in accordance with the term of BLEU scores for *Ken*, *SRI*, and *IRST*, under the impacts of *3-gram* and *5-gram* language models applying *Moses* and *Cdec* translation engines as Classical and Hierarchical decoders, respectively, in the translation process from *Persian* language to *Spanish* one.

TABLE 3.15: The performance of applying 3-gram and 5-gram language models through Ken, SRI, and IRST tool-kits on Persian-Spanish translation task using BLEU metric.

| Language models | 3-gram C | 5-gram C | 3-gram H | 5-gram H |
|-----------------|----------|----------|----------|----------|
| KenLM | 22.58 | 22.66 | 21.65 | 21.69 |
| SRILM | 22.19 | 22.25 | 21.29 | 21.41 |
| IRSTLM | 21.94 | 22.03 | 20.99 | 21.11 |

According to the results of translation process from *Persian* to *Spanish* based on the scores of BLEU in Table 3.15, under all conditions, independent of the kind of *n-grams* language models, the Classical-based system through *Moses* has better performance than the Hierarchical-based system through *Cdec*.

Figure 3.11 illustrates the performance chart of both Classical (*Moses*) and Hierarchical (*Cdec*) translation systems in *Persian-Spanish* translation task. The chart shows that the *5-gram* and *3-gram* Classical phrase-based translation models (the

Red and Blue bars) outperform the Hierarchical phrase-based translation models applying all language modelling tool-kits.

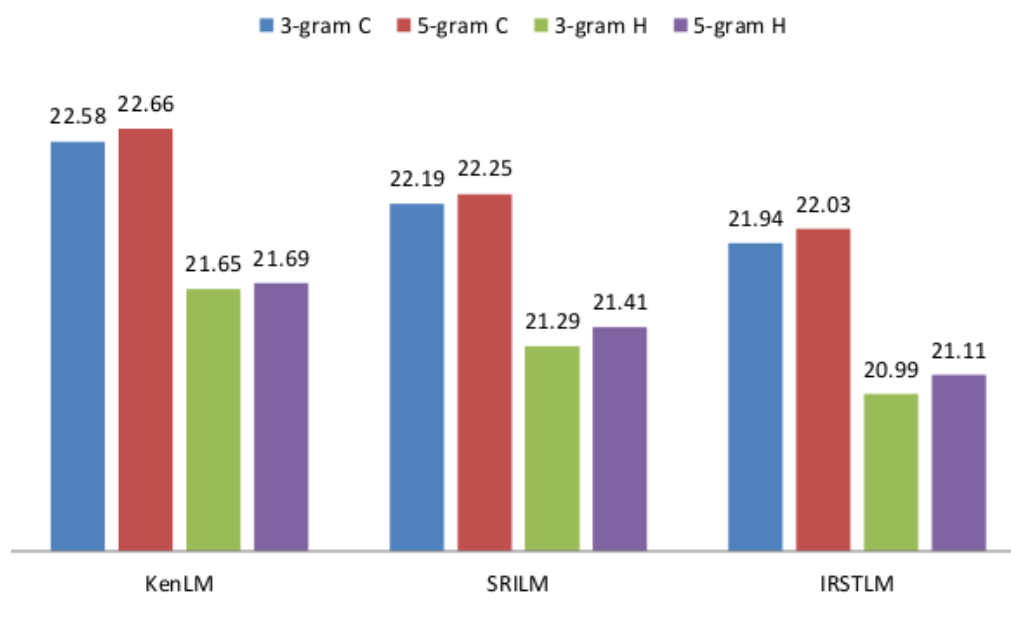


FIGURE 3.11: The performance chart of Persian-Spanish translation task in different kinds of language models according to BLEU.

On the other hand, the output results in the Tables 3.16 show the accuracy of translation in accordance with the term for all three kinds of *3-gram* and *5-gram* statistical language models using *Moses* and *Cdec* translation engines as Classical and Hierarchical decoders, respectively, according to BLEU scores.

TABLE 3.16: The performance of applying 3-gram and 5-gram language models through Ken, SRI, and IRST tool-kits on Spanish-Persian translation task using BLEU metric.

| Language models | 3-gram C | 5-gram C | 3-gram H | 5-gram H |
|-----------------|----------|----------|----------|----------|
| KenLM | 20.62 | 20.73 | 22.05 | 22.16 |
| SRILM | 20.38 | 20.50 | 21.80 | 21.93 |
| IRSTLM | 20.17 | 20.28 | 21.49 | 21.67 |

According to the results shown in Table 3.16, for implementing the translation process from *Spanish* language to *Persian* language, the Hierarchical phrase-based translation model outperforms the Classical phrase-based translation model under the SMT platform. In all cases (*3-gram* and *5-gram* of the Classical and Hierarchical phrase-based translation models), independent of the kind of *n-grams* language models, *Cdec* as a Hierarchical phrase-based translation system has better performance than *Moses* as a Classical phrase-based system.

Figures 3.12 demonstrates the performance chart of both Classical (*Moses*) and Hierarchical (*Cdec*) translation systems in *Spanish-Persian*. The charts shows that the *5-gram* and *3-gram* Hierarchical models (the Purple and Green bars) through *Cdec* have better performance than the other models in all cases of language models through *Moses*, respectively.

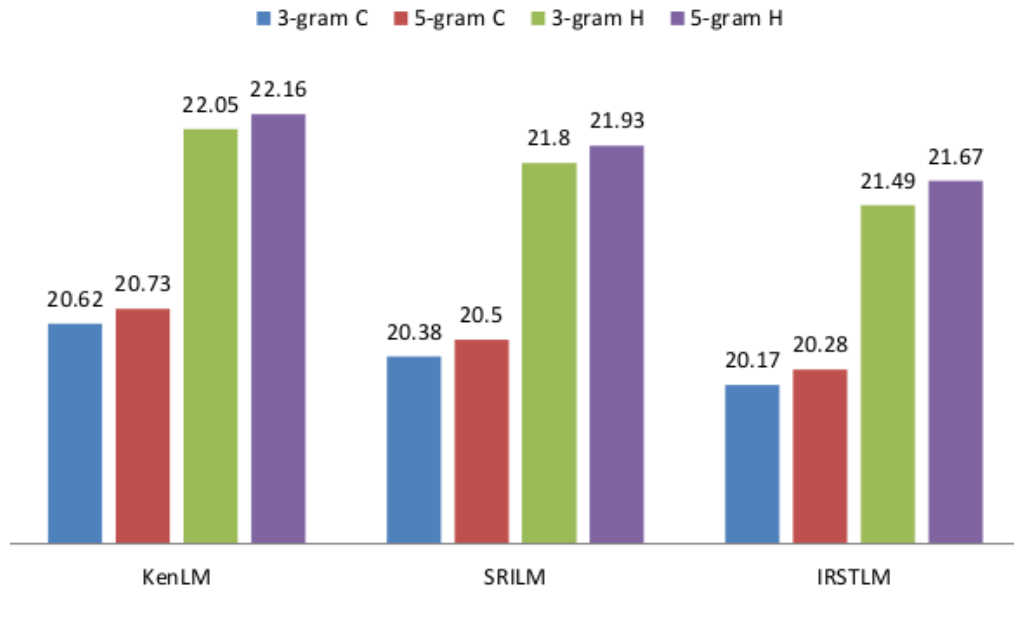


FIGURE 3.12: The performance chart of Spanish-Persian translation task in different kinds of language models according to BLEU.

Due to the fact that results acquired after doing the translation process for *Spanish* and *Persian* language pair (applying different kinds of n -gram language models) which comparatively differ from each other, it is best to focus all the energies on making the 5-gram Classical model for *Persian-Spanish* as well as 5-gram Hierarchical model for *Spanish-Persian* work. With less n -gram, the rules that have been generated emerge in smaller form. As a result of which, the size of corpus is not required to create a cover over the sparseness of the surrounding words.

3.6 Discussion

Generally, improving the statistical language model is one of the most reliable ways to improve the performance of SMT systems. This applies to both Classical and Hierarchical phrase-based systems. While the exact BLEU improvement might not be identical for both systems, it is unlikely that a particular n -gram LM would flip a result between systems if there is a significant gap. If the systems are very close to each other, it is possible that a better LM would help one more than the other and change the order. In machine learning, we can observe trends that hold most of the time, but we may always get a surprising result for an unusual data set.

Pairs of corresponding source and target language phrases were learned from the training data due to the fact that Hierarchical phrase-based translation was based on SCFG. The difference was primarily in the Hierarchical models i.e. phrases may contained gaps, and were represented by non-terminal symbols of the SCFG.

If a source phrase contained a non-terminal, then the target phrase will also contain that non-terminal. In such an instance, the decoder can easily replace the non-terminal with the help of any source phrase and its translation.

One of the major differences reported between *English* and *Spanish* with *Persian* was that of the word order. *Persian* as the target language possessed some features

that incurred negative effects on the SMT performance. *Persian* is considered to be much more richer in morphology than *English* and *Spanish*. For instance, if *Persian* language is characteristically analysed then it comes out as a morphologically rich language. The main feature of this language is that there is no need to distinct between the upper-case and lower-case letters. This language does not require the use of abbreviations or indefinite articles such as "a" or "an". There are certain other notable differences in *Persian* such as the sentence structure is totally distinct as the parts of speech are placed in unexpected places. While dealing with *Persian*, translators tend to invent new words because there are unlimited versions of spelling for certain *Persian* words. This act can trigger Out-Of-Vocabulary (OOV) output.

Another distinct feature of *Persian* is that it creates greater noise in training data that result in harder sparse-data problems. This issue occurs due to vocabulary that combines words from various sources. *Persian*, being rich in morphology on the target side means that besides selecting a lexically correct *Persian* equivalent of an *English* or *Spanish* word the SMT system must also correctly guess grammatical features. This means that translation would require us to perform significant reordering.

Our experiments also reveal that the Hierarchical model in *Cdec* achieved the quality that was similar to the phrase-based model in *Moses*, despite the fact that their implementation was less mature.

Generally speaking, phrase-based methods help in identifying the contiguous bilingual phrase pairs based on automatically generated word alignments. Phrase pairs are extracted up to a maximum length that is fixed because if the phrases are extremely long then they would rarely have a tangible impact during translation.

Extracted phrase pairs are reordered so that the fluent target output could be generated during the decoding process. Later on, the reordered translation output is evaluated under a distortion model. Furthermore, it is corroborated by one or more *n-gram* language models. At this point, there is certain confusion as these models do not have an explicit representation of how to reorder phrases.

In order to avoid any type of explosion with respect to search space, most of the systems place a limit on the distance so that the source segments could be moved within the source sentence. This limit, along with the phrase length limit, determines the scope of reordering represented in a phrase-based system.

3.7 Summary

In this chapter, a detailed comparison between Classical and Hierarchical phrase-based translation models provided. According to our in-depth comparison, the rules related to the Hierarchical model and the phrases for Classical model are distributed into separate tables in the parallel corpus. After the completion of this distribution, the data in a parallel corpus is used for generating a language model in training.

If we view in accordance to the summary then we will come to know that three mandatory outputs are returned for the testing process during the training process. These mandatory outputs are considered to be rule-table for the Hierarchical model, phrase-table for the Classical model, and language model for both. Furthermore, we need to get input sentence for translation in the testing process. Due to the fact that system can only accept one input per sentence, the input is designed in accordance

to one sentence per line. If the need arises to do translation according to the Hierarchical and Classical models then each decoder is executed in a separate way. Due to which it returns a separate translation result.

In the experiments we showed that the Classical model outperforms the Hierarchical model in *English-Spanish* translation as well as *Persian-English* and *Persian-Spanish* translations, while the Hierarchical model has better performance than the Classical one in *Spanish-English* task as well as *Spanish-Persian* and *English-Persian* tasks.

On the other hand, we investigated the effects of applying different *n-gram* language models on our considered case-study language pairs, and we concluded that, if the Classical model is preferable in one case, all the *n-grams* language models of this model are preferable as well. In the opposite side, we concluded the same for the other model. It means that if a Hierarchical model outperforms in one case, all the related *n-grams* language models of this model have better performance than the other model(s).

Chapter 4

Direct-Bridge Combination for Minimal Parallel-Resource SMT

Since state-of-the-art Statistical Machine Translation (SMT) has shown that, high-quality translation output is dependent on the availability of massive amounts of parallel texts in the source and target languages, the biggest issue is that high-quality parallel corpus is not always available. This is one of the reasons that SMT is to introduce a third language, named bridge (pivot) for the purpose of resolving the training data scarcity. This third language will act as an intermediary language for which there exist high-quality source-bridge and bridge-target bilingual corpora.

This chapter investigates the idea of making effective use of bridge-language technique to respond to minimal parallel-resource training text bottleneck reality, and also provides our proposed method to improve the translation quality, in the case of *Persian-Spanish* minimal parallel-resource language pair using a well-resource language such as *English* as the bridge one.

In this chapter, first, the sentence-level bridging (transfer method) and the phrase-level in turn (triangulation method) are introduced, then a performance comparison between the phrase bridging strategy and the sentence bridging one is demonstrated as well. Later, an interpolation as a combination model between direct and bridge-based translation systems is investigated to enhance the translation quality for minimal parallel-resource language pairs. After that, we investigate the proposed improvement in bridge-language technique. Finally, we propose an optimized direct-bridge combination (ODBC) method to enhance the translation performance, and also we analyse the effects of this proposed method on our considered case-study minimal parallel-resource SMT system.

4.1 Introduction

The primary goal of SMT is to conduct the translation of the source-language sequences into a target-language. This must be achieved after plausibility of the source has been assessed along with the target sequences. At this point only those target sequences must be analysed that have a specific relation to the existing bodies of translation between the two languages (Ahmadnia et al., 2017).

Special effects are incurred by the sizeable bodies of aligned parallel corpora on the functions and performance of SMT systems. However, gathering parallel data becomes quite an issue if it has to be done in practice because of two reasons i.e. high-costs, and limitations in scope. Both of these reasons must intense pressure on the concerned research and the application of that research. This is the reason that scarce nature of the parallel data with respect to different languages is considered to

be one of the main issues in SMT (Babych et al., 2007). These types of Corpora are not easily found, especially in the case where minimal-resource language pairs are involved. Even if we analyse the cases involving the well-resource languages, such as *Europarl*¹ (Koehn, P., 2005), the SMT performance adopts a downward trend in significant way if it is applied to a slightly different domain.

This is the reason that the efficiency of the performance decreases as the change occurs in the domain. In order to tackle with the lack of parallel data, bridge (pivot) language technique, as a common solution is used. If the languages with inefficient Natural Language Processing (NLP) resources are to be involved then this issue becomes significant in relation to an SMT system. However, the most encouraging point is the sufficient availability of the resources between them and the other languages. This issue becomes important if the languages are involved. Even though it has been determined that improvement in general case does not occur as a result of intermediary languages, still this particular idea can be employed in the form of a simple method. This idea is adopted as a simple method so that translation performance for the existing systems could be enriched (Matusov et al., 2008).

Recently some efforts have been made so that the quality and recall of the bridge-based SMT could be enhanced. During one of the experiments, Kumar et al. (2007), sought help from the bridge language so that word alignment system along with the procedure for combining word alignment systems could be created. They did this experiment so that these systems could be created from multiple bridge languages. When it comes to the stage of obtaining the final translation then it is conducted through consensus decoding. The entire process of consensus decoding combines hypotheses that are gained after all the bridge language word alignments are obtained.

Later on, the effect of the bridge language during the final translation system was examined by Paul et al. (2009). They revealed through their experimentation that if the size of training data is small in any case then the bridge language should be same as the source one but if training data is large then in that case, the bridge language looks similar to the target one. Whatever the case is, it will be preferable to use bridge language with a structure similar to source and target languages.

Recently an experiment was conducted by Zhu et al. (2014), during which the researchers focused on resolving the issue through the help of source -target translations. However, the interesting fact is that these source-target translations were not generated because the source phrase and target phrase that correspond with these translations connect to different bridge phrases. In order to decrease the intensity of the problem, the researchers connected the concerned translation phrases between source and target languages by utilizing the *Markov Random Walks*.

One of the basic ways through bridging idea can be demonstrated is by the large-size of the newly created bridge phrase-table. Recently some effort has been made so that precision on bridging could be improved. According the studies conducted by Saralegi et al. (2011), it has been confirmed that transitive property between three languages does not exist. So it can easily be said that most of the translations that were produced within the final phrase-table could not be right. This is the reason that phrase-table two methods are used so that wrong and weak phrases could be removed.

¹The Europarl corpus is a set of documents that consists of the proceedings of the European Parliament from 1996 to the present. The data that makes up the corpus was extracted from the website of the European Parliament and then prepared for linguistic research.

One of the methods has been derived from the structure of source dictionaries while the other method has been derived from the distributional similarity. Recently a strategy has been introduced that uses context vectors so that pruning method could be created for the purpose of removing the phrase pairs. At this point, only those phrase pairs are removed that link to each other either through weak translations of through polysemous bridge phrase (Tofighi Zahabi et al., 2013).

4.2 Bridge Language Theory

High-quality data set is not always available for training the SMT systems. One of the possible ways to solve this impasse is to using a third language as a bridge (pivot) one for which there exist high-quality source-bridge and bridge-target bilingual resources.

A bridge language is an artificial or natural language used as an intermediary language for translation between many different languages. The bridge language technique is an idea to generate a systematic SMT when a proper bilingual corpus is lacking or the existing ones are weak.

The major drawback and concern of generated translations through bridging is the translation quality, as it is possible to produce erroneous translations by transferring errors or ambiguities from a language pair to another through the pivot language. However, when language resources in specific language pairs do not exist or are scarce, the use of pivot languages as data bridges can prove to be a convenient linguistic short-cut for offering language services or building and enhancing language resources.

The first and foremost requirement is to have a high-quality parallel corpus while dealing with the SMT. However, such situation is not possible in the case of minimal parallel-resource languages. Thus, the most advanced research on multilingual SMT has focused on the use of bridge language for the translation of such resource disadvantaged languages.

English, due to its richness in language resources, comes first as possible bridge language. Generally, the choice of bridge language is done according to two criteria:

1. Availability of language resources.
2. Relatedness between source and bridge languages.

However, the issue is that preceding criteria might not be reliable enough to help in choosing the best bridge language. According to the recent researches, use of the *non-English* language as bridge creates an improvement in the system performance.

4.3 Bridging Approaches

There are methods by which the resources of bridge language can be utilized as explained in (Wu and Wang, 2007), namely;

- Sentence translation or transfer approach.
- Synthetic corpus approach.
- Phrase-table construction or triangulation approach.

4.3.1 Transfer Approach

This method is also recognized as cascade, or sentence translation bridge strategy. The transfer method first converts the source-language into bridge (pivot) one by translating it with the help of source-bridge translation system. After then it converts from bridge-language to target one through the bridge-target translation system.

Given a source sentence s , we can also translate it into n bridge-language sentences $(b_1, b_2, b_3, \dots, b_n)$, using a source-bridge translation system. Each of these n sentences, b_i , can then be translated into m target-language sentences $(t_{i1}, t_{i2}, t_{i3}, \dots, t_{im})$, using bridge-target translation system. Thus, in total we will have $(m \times n)$ target-language sentences. These sentences can then be re-scored with the help of source-bridge and bridge-target translation system scores.

If we denote source-bridge system features as γ^{sb} and bridge-target system features as γ^{bt} , the best scoring translation is calculated using Equation (4-1):

$$\hat{t} = \arg \max_t \sum_{k=1}^L \left(\lambda_k^{sb} \gamma_k^{sb}(s, b) + \lambda_k^{bt} \gamma_k^{bt}(b, t) \right) \quad (4.1)$$

Where L is the number of features used in SMT systems, λ^{sb} , and λ^{bt} are feature weights.

In other words, in transfer approach, first the source sentences are translated into the bridge ones, followed by translation of these bridge sentences into the target ones separately. We choose the highest scoring sentence amongst the target sentences. In this approach for assigning the best target candidate sentence t to the input source sentence s , we maximize the probability $P(t|s)$ by defining hidden variable b , which stands for the bridge language sentences, we gain:

$$\arg \max_s P(t|s) = \arg \max_s \sum_b P(t, b|s) = \arg \max_s \sum_b P(t|b, s) P(b|s) \quad (4.2)$$

Assuming that, s and t are independent given b :

$$\arg \max_s P(t|s) \approx \arg \max_s \sum_b P(t|b) P(b|s) \quad (4.3)$$

In Equation (4.3) summation on all b sentences is difficult, so we replace it by maximization, and Equation (4.4) is an estimate of Equation (4.3):

$$\arg \max_s P(t|s) \approx \arg \max_s \max_b P(t|b) P(b|s) \quad (4.4)$$

Instead of searching all the space of b sentences, we can just search a subspace of it. For simplicity we limit the search space in Equation (4.5). A good choice is b subspace produced by the k -best list output of the first SMT system (source-bridge):

$$\arg \max_s P(t|s) \approx \arg \max_s \max_{b \in k\text{-best}(t)} P(t|b) P(b|s) \quad (4.5)$$

In fact each sentence s of the source test set is mapped to a subspace of total b space and search is done in this subspace for the best candidate sentence t of the second SMT system (bridge-target).

4.3.2 Synthetic Corpus Approach

This method attempts to develop a synthetic source-target corpus by translating the bridge part in the source-bridge corpus, into the target language by means of a bridge-target model, and translating the bridge part in the target-bridge corpus into the source language with a bridge-source model.

Eventually, it combines the source sentences with the translated target sentences or combines the target sentences with the translated source sentences. The source-target corpora that is created using the above two methods can be blended together so a final synthetic corpus could be produced. However, it is complicated to create a high-quality translation system with a corpus compiled merely by an SMT system.

4.3.3 Triangulation Approach

This method is known as phrase-table multiplication, or phrase translation pivot strategy. In this approach, phrase s in the source-bridge phrase-table is connected to b , and this phrase b is associated with phrase t in the bridge-target phrase-table. We link the phrases s and t in the new phrase-table for the source-target language pair. For scoring the pair phrases of the new phrase-table, assuming $P(s|b)$ as the score of the source-bridge phrases and $P(b|t)$ as the score of the bridge-target phrases, then the score of the new pair phrases s and t , $P(s|t)$, in source-target phrase-table is counted.

In this method, the training of source-bridge and bridge-target translation models are conducted with the help of source-bridge and bridge-target corpora respectively. Using these two models, we have so far induced a source-target model. The two important components to be induced are:

1. Phrase translation probability.
2. Lexical reordering weight.

Phrase translation probability is mainly induced due to the assumption that source and target phrases are conditionally independent, but this is considered when they are conditioned on bridge phrases. It can be given as:

$$P(s|t) = \sum_b P(s|b) P(b|t) \quad (4.6)$$

Where s , b , and t are phrases in the source, bridge and target languages, respectively.

According to the research conducted by Koehn et al. (2003), the lexical reordering weight depends on word alignment information in a phrase pair (s, t) , and lexical translation probability $w(s|t)$. The induction of word alignment is done from the source-bridge and bridge-target alignment in order to calculate the lexical reordering weight. Furthermore, the estimate of lexical translation probabilities and the induced alignment are used to calculate the lexical reordering weights. Here we must keep in mind that both lexical reordering weight and phrase translation probability are language dependent.

The triangulation approach will be introduced here that mainly performs the phrase-based SMT for a source-target language pair. This phrase-based SMT is performed through the help of two bilingual corpora of source-bridge and bridge-target languages. Two translation models are then trained so they can deal with source-bridge and bridge-target. Next these models are made a base for the construction of

bridge translation model, which is mainly created for the source-language, containing b as a bridge-language.

Algorithm 4 shows the triangulation mechanism.

Algorithm 4 Triangulation technique

Input: Phrase-table between source and bridge (P_{s-b}), Phrase-table between bridge and target (P_{b-t}), Selecting top-n phrase pairs (N).

Output: $P_{triangulation}$. // Which is initially empty.

- 1: **for** all (source-bridge) in top-n P_{s-b} **do** // Searching to find all source-bridge phrase pairs in N .
 - 2: **if** bridge phrase in P_{b-t} , then
 - 3: **for** all (bridge-target) pairs in P_{b-t} **do**
 - 4: Compute feature values for (source-target)
 - 5: **end for**
 - 6: Select top-n (source-target) pair, and add to $P_{triangulation}$
 - 7: **end if**
 - 8: **end for**
-

4.3.3.1 Phrase Translation Probabilities

With the help of source-bridge and bridge-target bilingual corpora, we can train two phrase translation probabilities $P(s_i|b_i)$ and $P(b_i|t_i)$. At this point, b_i is the phrase in bridge language b . We derive the phrase translation probability $P(s_i|t_i)$ according to the following model:

$$P(s_i|t_i) = \sum_{b_i} P(s_i|b_i, t_i) P(b_i|t_i) \quad (4.7)$$

The phrase translation probability $P(s_i|b_i, t_i)$ does not depend on the phrase t_i in the target language as it comes from the source-bridge bilingual corpus. Thus, Equation (4.7) can be rewritten as:

$$P(s_i|t_i) = \sum_{b_i} P(s_i|b_i) P(b_i|t_i) \quad (4.8)$$

In order to check the correction of probability calculations, the formulation of phrase translation probability $P(s_i|t_i)$ is marginalised:

$$P(s_i|t_i) = \sum_{b_i} P(s_i, b_i|t_i) \quad (4.9)$$

Now the chain rule is used:

$$P(s_i|t_i) = \sum_{b_i} P(s_i|b_i, t_i) P(b_i|t_i) \quad (4.10)$$

Since, we have source-bridge corpus available, the calculation of first term in the above equation will not depend on b i.e. $P(s_i|b_i, t_i)$. It will now reduce to $P(s_i|b_i)$. Thus, the final equation will be:

$$P(s_i|t_i) = \sum_{b_i} P(s_i|b_i) P(b_i|t_i) \quad (4.11)$$

4.3.3.2 Lexical Reordering Weights

According to Koehn et al. (2003), lexical weight can be calculated through the help of the following model:

$$P_w(s|t, \alpha) = \prod_{i=1}^n \frac{1}{|j|(i, j) \in \alpha|} \sum_{\forall (i, j) \in \alpha} w(s_i|t_j) \quad (4.12)$$

For the purpose of calculating the lexical weight, we first need to obtain the alignment information α between two phrases s and t . We will be able to calculate the lexical translation probability $w(s|t)$ according to the alignment information after we have found the desired information.

The alignment information for the phrase pair $(s|t)$ can be derived with the help of two phrase pair $(s|b)$ and $(b|t)$. Let α_1 and α_2 be the word alignment information inside phrase pairs $(s|b)$ and $(b|t)$ respectively.

$$\alpha = \{(s, t) | \exists P : (s, b) \in \alpha_1 \& (b, t) \in \alpha_2\} \quad (4.13)$$

With this induced alignment information, there exists an approach to estimate the probability directly from the induced phrase pairs named phrase method. If we use K to denote the number of induced phrase pairs, co-occurring frequency of the word pair (s, t) can be estimated according to the following model²:

$$count(s, t) = \sum_{k=1}^K P_k(s, t) \sum_{i=1}^n \delta(s, s_i) \delta(t, t_{\alpha_i}) \quad (4.14)$$

Where, for phrase pair k , $P_k(s|t)$ is phrase translation probability.

Thus, lexical translation probability can be estimated as:

$$w(s, t) = \frac{count(s, t)}{\sum_{s'} count(s', t)} \quad (4.15)$$

$w(s|t)$ can also be calculated using word method as:

$$w(s, t) = \sum_b w(s, b) w(b, t) \quad (4.16)$$

Where, $w(s|b)$ and $w(b|t)$ are two lexical probabilities.

4.3.4 Interpolated Model

Assuming that a small source-target parallel corpus is available to us. In such a case, if we train the translation model just on this corpus then it will result in poorly performing system. The only reason that will result in such poor performance will be sparse data.

We can seek refuge in additional source-bridge and bridge-target parallel corpora for the purpose of improving this performance. Apart from that, we can also make use of one or more bridge language in order to improve the translation performance. There are certain chances that different bridge language will get exposed to

² $\delta(x, y) = 1$ if $x = y$; otherwise 0.

different language phenomenon due to which it will improve the translation quality by including the quality source-target phrase pairs.

If we include n bridge languages then the estimation of n bridge models can be done according to the description mentioned above. We will use the linear interpolation so we could combine all these models with the standard model trained with the source-target corpus. The phrase translation probability and the lexical reordering weights are estimated as shown in Equations (4.17) and (4.18), respectively:

$$P(s|t) = \sum_{i=0}^n \alpha_i P_i(s|t) \quad (4.17)$$

$$P_w(s|t, \alpha) = \sum_{i=0}^n \beta_i P_{w,i}(s|t, \alpha) \quad (4.18)$$

Where, $\sum_{i=0}^n \alpha_i = 1$, and $\sum_{i=0}^n \beta_i = 1$.

$P_0(s|t)$ and $P_{w,0}(s|t, \alpha)$, denote the phrase translation probability and the lexical weight respectively, trained with the source-target corpus.

$P_i(s|t)$ and $P_{w,i}(s|t, \alpha)$, ($i = 1, 2, \dots, n$), are the phrase translation probability and the lexical reordering weights estimated by using bridge languages. α_i and β_i are interpolation coefficients. Figure 4.1 shows a general view of interpolated model.

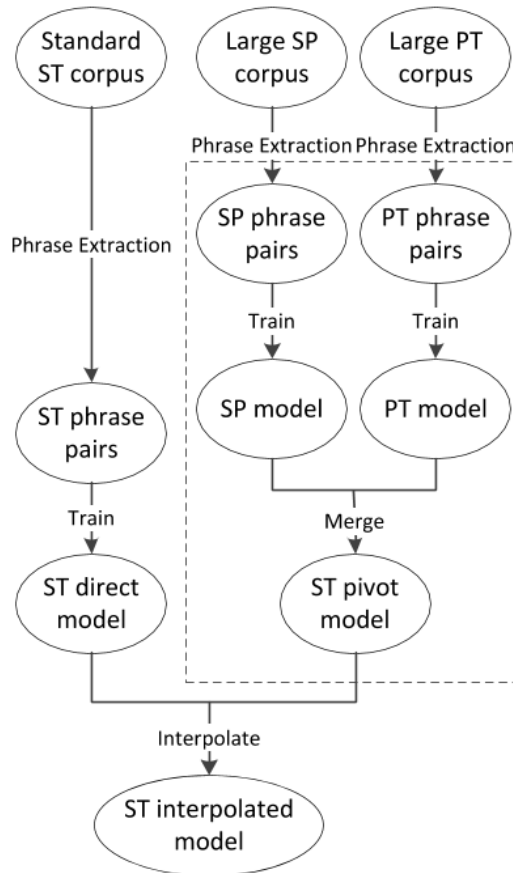


FIGURE 4.1: The interpolation method schematic between source, pivot, and target models.

4.4 Proposed Improvements in Bridge Language Technique

The recent developments in SMT made it possible to build a prototype system for any language pair just within a few hours. Nakov and Ng (2012) proposed a language-independent approach for improving SMT for scarce-resource languages, exploiting their similarity to well-resource ones. In simple words it can also be said, that we have a low-resource language (*source*) which is directly linked to the well-resource language (*target*). Source and target may have similarities in word order, vocabulary, spelling, syntax, etc.

We improve translation from scarce-resource language (*source*) by converting it into a well-resource language (*bridge*) with the help of bilingual text containing a limited number of parallel sentences for *source-bridge* and large bilingual text for *bridge-target*. In order to use bilingual text of one language to improve SMT for some related languages, two general strategies are used:

1. Bilingual text interpolation, where possible repetitions of original bilingual text are required for balance.
2. Phrase-table combination, where each bilingual text is used to build separate phrase-table, and then two phrase-tables are combined.

4.4.1 Interpolating Bilingual Texts

This approach simply requires us to interpolate the bilingual texts for *source-bridge* and *bridge-target* into a large bilingual text. It can help in improving the alignments obtained from *source-bridge* bilingual text. This is because, additional sentences can provide context for rare words in that bilingual text.

Interpolation also holds the capacity to provide us with more source-side translation options. Due to which it is the main source which increases lexical coverage and reduces the number of Out-Of-Vocabulary (OOV) words. It can also introduce new non-compositional phrases on source-side so that fluency could be improved. Furthermore, it offers new target language phrases.

Here we must keep in mind that the inappropriate phrases from *target* that do not exist in *source* will fail to match the test time input. However, this approach of simple interpolation can cause certain problems.

Since the size of *bridge-target* bilingual text is much higher than *source-bridge* bilingual text, the former will prove more dominating during the phase of word alignment and phrase extraction. This can affect the lexical and phrase translation probabilities in negative way, as it will result in poor performance.

This imbalance of bilingual texts can be corrected by repeating smaller *source-bridge* bilingual text several times so that large one does not dominate. Additional and original training bilingual texts are combined in following ways:

- **1×int:** A simple interpolation of additional and original bilingual text to generate a new training bilingual text, which is basically used to train a new phrase-based SMT system.
- **n×int:** An interpolation of n copies of original bilingual text and a copy of additional bilingual text in order to create a new training bilingual text³.

³The value of n is selected so that original bilingual text could approximately match the size of additional bilingual text.

- **n(aligned)×int:** We interpolate n copies of original bilingual text and one copy of additional bilingual text to form a new training bilingual text. At first the word alignments are generated from this new bilingual text, and then all sentence pairs and word alignments are discarded except for one copy of original bilingual text⁴.

4.4.2 Combining Phrase-tables

There is also an alternate way to use additional training bilingual text and that is to build separate phrase-tables. These phrase-tables can be used together by merging them, or interpolating. Phrase-table construction method possesses various advantages such as:

- The phrase pairs extracted from *source-bridge* bilingual text are clearly different from the riskier ones from *bridge-target* bilingual text.
- The lexical and phrase translation probabilities are blended together in proper way.

If seen in the negative light, word alignments for sentences in *source-bridge* bilingual text are not as strong as they were before this. Following are the three phrase-table construction strategies:

- **Two-tables:** Two separate phrase-tables are constructed from two bilingual texts that could be used as alternative decoding paths.
- **Concatenation:** Two separate phrase-tables are built from original and additional bilingual texts. In order to combine corresponding conditional probabilities the linear interpolation is used.

$$P(t|s) = \alpha P_{original}(t|s) + (1 - \alpha) P_{additional}(t|s) \quad (4.19)$$

The value of α is optimized over a development dataset.

- **Merge:** In this case, two separate phrase-tables are built from original and additional bilingual texts. We keep all *source-target* phrases from $T_{original}$ and then include those *source-target* phrase pairs from $T_{additional}$ that were not present in $T_{original}$. The associated lexical and phrase translation probabilities are retained for each added phrase pair.

4.5 Experiments

Our case study is *Persian-Spanish* minimal parallel-resource language pair, and we employ bridge language technique to improve the translation quality in both forward and backward translation directions. In this case, *English* is used as the bridge language, and the source-bridge SMT is combined with the bridge-target one, where the relatively large corpora of each may be used in support of the source-target pairing.

⁴It must be kept in mind that only the word alignments from original bilingual text are induced through additional statistical information from additional bilingual text. Later on, these alignments are used to build a phrase-table.

The data is gathered from in-domain Tanzil parallel corpus⁵ (Tiedemann, 2012). In this corpus, the *Persian-Spanish* part encompasses more than (68K) parallel sentences, nearly (2.06M) words in the *Persian* side, and more than (1.45M) words in the *Spanish* side. Besides, the *Persian-English* part includes more than (138K) parallel sentences, around (377K) *Persian* words, and more than (830K) *English* words. The *English-Spanish* part contains more than (1M) parallel sentences, approximately (31M) words in the *English* side, and over (26M) words in the *Spanish* side. Table below presents the corpus statistics, which have been used in our experiments, including the source and the target languages information in each direction.

TABLE 4.1: Corpus statistics including the source and target languages information.

| Directions | Pesian↔ English | English↔ Spanish | Persian↔ Spanish |
|--------------|-----------------|------------------|------------------|
| Sentences | 138,822 | 1,028,996 | 68,601 |
| Source words | 376,933 | 30,872,937 | 2,058,231 |
| Target words | 832,696 | 26,143,026 | 1,454,778 |

The training part consist of (60k) parallel sentences. In order to conduct the tuning and testing steps, we gathered parallel texts from *Tanzil* corpus; (3K) sentences for the tuning step, and (5K) sentences for the testing step were extracted. The *tokenize.perl* script has been employed for tokenizing all data sets.

Moses package⁶ (Koehn et al., 2007), is employed for training our SMT systems. Through employing this decoder, *fast-align* approach (Dyer et al., 2013), is applied for word alignment. We employ *5-grams* language model for all SMT systems and they are developed by means of the *KenLM* tool-kit (Heafield, 2011).

In addition, for evaluating the systems performance, we use both the *BLEU* metric. We set the beam-size to (100), and the distortion limit to (6). We restrain the maximum target phrases to (6) that are loaded for each source phrase, and we draw on the same other default features of *Moses* translation engine.

4.6 Results and Evaluation

For the translation systems we conduct two sets of experiments in each translation direction. In the first set of experiments, three portions are investigated as follows:

1. For conducting the first portion of the first phase of our experiments, *English* was utilised by the transfer bridging system as an interface between two separate phrase-based SMT systems, specifically a *Persian-English*, and an *English-Spanish* direct translation systems. Besides, while translating *Persian* to *Spanish*, the *English top-1* output of the *Persian-English* system was forwarded as input to the *English-Spanish* system. The *English* language model which was used to train the *Persian-English* system is developed from the counterpart of the *Spanish* data used to build the *Spanish* language model in our considered parallel corpus.

⁵<http://opus.lingfil.uu.se/Tanzil.php>

⁶<http://www.statmt.org/moses>

2. For applying the triangulation method during the second portion of the first experiments phase, we required to create a phrase-table to train the phrase-based SMT system. Therefore, a *Persian-English*, and an *English-Spanish* phrase-tables were needed. Based on these phrase-tables, we formed a *Persian-Spanish* phrase-table. Furthermore, a matching algorithm that identifies parallel sentence pairs among the phrase-tables were utilized. After identifying candidate sentence pairs, we finally use a classifier to determine if the sentences in each pair are a good translation for each other and update our *Persian-Spanish* phrase-table with the selected pairs.
3. For the last portion of the first phase of the experiments, we examine a combination approach (interpolated method) so as to achieve a higher coverage and a better translation quality, aiming at efficiently merging both the transfer, and the triangulation interpolated models with a direct translation model developed from a given parallel corpora. In particular, this approach is an attempt to combine the direct and bridge-based models in order to rise the amount of the gained information. In order to achieve this aim, several combination models are approachable and practical. For interpolation of direct and transfer models, after phrase extraction of source-bridge and bridge-target phrase pairs, we train and merge the source-bridge and bridge-target models respectively, and finally interpolate the source-target direct translation model with the generated source-bridge-target translation model. For interpolate the direct model and the multiplication (triangulation) model, we employ a combination model where the translation options are gathered from one table, and additional options are collected from other tables. Reaching similar translation options in multiple tables, we form separate translation options for each occurrence with different scores.

Table 4.2 illustrates the results of both *Persian-Spanish* and *Spanish-Persian* standard direct, and bridge-based (transfer, triangulation, and interpolated models) translation systems through *English* as the intermediary language. The results indicate that, the bridge-based translation method is suitable for the scenario that there exist large amounts of source-bridge and bridge-target bilingual corpora and only a little source-target bilingual data. Thus we selected (60K) sentence pairs from the source-target bilingual corpora to simulate the lack of source-target bilingual data.

TABLE 4.2: The BLEU scores comparing the performance of direct translation with bridge-based translation for *Persian-Spanish* SMT system and back translation through *English* as bridge language.

| Translation systems | Pe-(En)-Es | Es-(En)-Pe |
|---------------------------------------|------------|------------|
| Direct | 19.39 | 19.07 |
| Transfer | 20.78 | 20.33 |
| Triangulation | 21.55 | 21.02 |
| Interpolated 1 (Direct+Transfer) | 20.18 | 19.80 |
| Interpolated 2 (Direct+Triangulation) | 20.57 | 20.14 |

As seen in Table 4.2 in triangulation, as the best bridging technique, the performance of *Persian-Spanish* and back translation systems, through *English*, relative increase from direct systems are approximately (11.11%), and (11.02%) respectively.

This suggests that, we are making better use of the available resources. The differences between bridge language method and direct translation approach are statistically significant confidence level.

Figure 4.2 shows the chart of *Persian-Spanish* and back translation performance between the direct and bridge translation systems according to their BLEU scores.

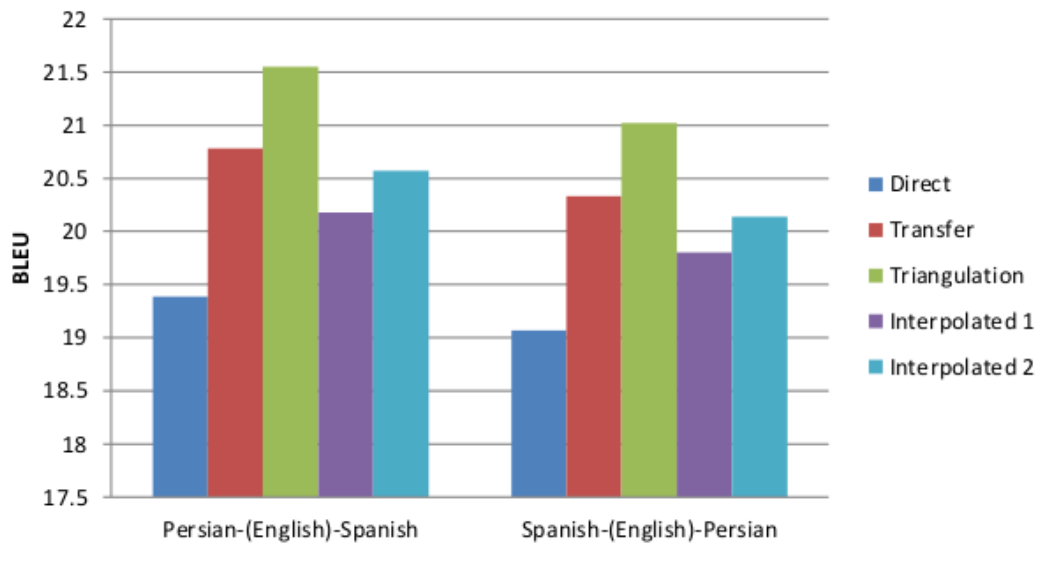


FIGURE 4.2: Performance comparison of the direct and bridge-based translation systems for both *Persian-Spanish* and *Spanish-Persian* tasks.

In the second set of experiments we investigate the effects of applying the proposed improvements to overcome data scarcity bottleneck and improve the performance of minimal-resource SMT systems through bridge language idea.

For this set of experiments we use the same language pair (*Persian* and *Spanish*) extracted from *Tanzil* parallel texts like previous set of experiments, and also *Moses* package is used as our translation engine with the same features as the previous experiments set.

Two general strategies are mainly implemented when the *Persian-Spanish* SMT and back translation is being carried out for this set of experiments:

1. **Sequential strategy:** This strategy basically focuses on creating a sequential link between the two SMT systems. One link is created between *Persian* and *English*, while the other link is created between *English* and *Spanish*. Due to the fact that errors from one system contribute towards propagating to the input of the other system, this entire procedure is basically known as error additive approach.
2. **Direct strategy:** This strategy uses the *English-Persian* SMT system so that the entire *English* side of the *Spanish-English* corpus could be translated into the *Persian*. The *English-Persian* SMT system used in this procedure belongs to a general domain. Later on, this automatically translated *Persian* text helps in the training procedure of *Spanish-Persian*. At this point, it can be easily be realized that errors arising in the *English-Persian* system might get very low probabilities during the training of *SSpanish-Persian* translation system as there is not a single chance that they will correlate with *Spanish* test.

In this portion, a number of experiments were done in order to test the similarity between the original (*Persian* and *Spanish*) and the intermediary language (*English*).

Persian-Spanish SMT and back translation is improved using *English* as bridge language. Various conclusions are drawn according to results of the experiments. It is clear that relative languages can help to improve SMT.

Method of simple interpolation is helpful, but it can be problematic when additional sentences are way more than original. Interpolation works well if original bilingual text is repeated enough number of times to match to the size of additional bilingual text. To give additional weighting to original phrases in merging method is a good strategy. Improvement in system is due to improvement in word alignment as well as due to increased lexical coverage.

Table 4.3 provides the results of *Persian-Spanish* minimal parallel-resource SMT system and back translation as well via *English* as auxiliary language through the interpolating bilingual texts improvements.

TABLE 4.3: The BLEU scores comparing the performance of different interpolating bilingual texts improvements for *Persian-Spanish* and *Spanish-Persian* SMT systems through *English* as bridge language.

| Interpolation bilingual texts | Pe-(En)-Es | Es-(En)-Pe |
|-------------------------------------|------------|------------|
| $1 \times \text{int}$ | 21.62 | 21.48 |
| $n \times \text{int}$ | 21.71 | 21.53 |
| $n(\text{align}) \times \text{int}$ | 21.78 | 21.65 |

Figure 4.3 shows the performance chart of the systems based on interpolation bilingual texts improvements according to BLEU scores.

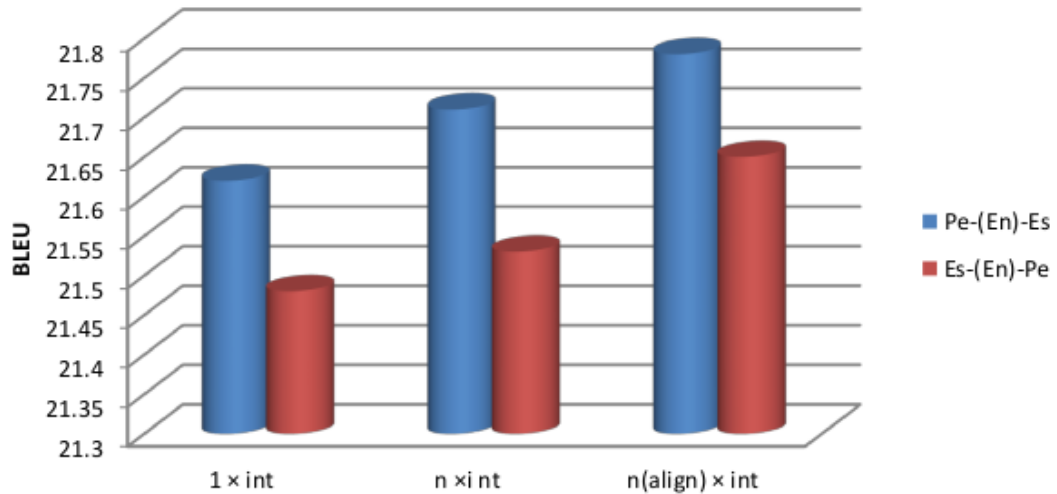


FIGURE 4.3: Comparison of interpolation bilingual texts methods performance for both *Persian-Spanish* and *Spanish-Persian* SMT tasks according to BLEU score.

Considering the interpolation bilingual texts scenario, the performances of $1 \times \text{int}$, $n \times \text{int}$, and $n(\text{align}) \times \text{int}$ systems relative increase from direct *Persian-Spanish* translation system are approximately (11.15%), (11.19%), and (11.23%) respectively. While in the *Spanish-Persian* translation task the performance of mentioned methods outperform the direct translation system by relative increase approximately (11.26%) with $1 \times \text{int}$, (11.28%) with $n \times \text{int}$, and (11.35%) with $n(\text{align}) \times \text{int}$.

On the other hand, these approaches also outperforms the triangulation technique which has the best performance in both forward and backward directions for *Persian* and *Spanish* translation task. The relative increase from the forward direct system is approximately (10.03%) by $1 \times int$, (10.07%) by $n \times int$, and (10.10%) by $n(align) \times int$. The performance of these mentioned systems relative increase from the backward direct system are approximately 10.36%, 10.24%, and 10.29% respectively.

In Table 4.4 we provide combining phrase-tables improvements results for both *Persian-Spanish* minimal-resource SMT system and vice versa via *English* as pivot language.

TABLE 4.4: The BLEU scores comparing the performance of different interpolating bilingual texts improvements for *Persian-Spanish* and *Spanish-Persian* SMT systems through *English* as bridge language.

| Interpolation bilingual texts | Pe-(En)-Es | Es-(En)-Pe |
|-------------------------------|------------|------------|
| Two-tables | 21.91 | 21.54 |
| Concatenation | 22.07 | 21.68 |
| Merge | 22.21 | 21.81 |

Figure 4.4 illustrates the performance chart of the systems based on combination phrase-tables improvements according to BLEU scores.

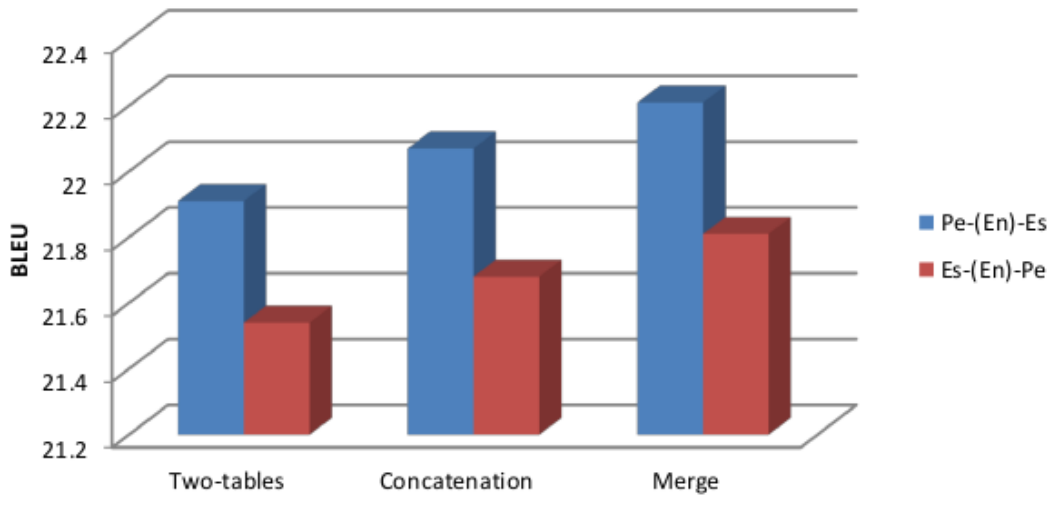


FIGURE 4.4: Comparison of combination phrase-tables approaches performance for both *Persian-Spanish* and *Spanish-Persian* SMT tasks according to BLEU score.

Considering the combination phrase-tables hypothesis, the performance of *Two-tables*, *Concatenation*, and *n(Merge)* approaches relative increase from direct *Persian-Spanish* translation system are approximately (11.29%), (11.38%), and (11.45%) respectively. While in the *Spanish-Persian* translation task the performance of mentioned methods outperform the direct translation system by relative increase approximately (11.29%) with *Two-tables*, (11.36%) with *Concatenation*, and (11.43%) with *Merge*.

On the other hand, these approaches also outperforms the triangulation technique which has the best performance in both *Persian-Spanish* and *Spanish-Persian* translation tasks. The relative increase from the forward direct system is approximately (10.16%) by *Two-tables*, (10.24%) by *Concatenation*, and (10.30%) by *Merge*. The performance of these mentioned systems relative increase from the *Spanish-Persian* direct system are approximately 10.24%, 10.31%, and 10.37% respectively.

As seen in Tables 4.3 and 4.4, and Figures 4.2 and 4.3, in all cases, the combining phrase-tables approach outperforms the interpolating bilingual texts approach, and both of these proposed improvements outperform the triangulation model (as the best standard bridging method) in both forward and backward translation directions between *Persian* and *Spanish* language pairs through *English* as bridge language.

Our experiments results prove that it is possible to generate a large-scale SMT system between *Persian* and any other language as long as there many parallel corpus available between that language and *English* (such as *Spanish*).

4.7 Proposed Method

During this section we will be proposing the *Optimized Direct-Bridge Combination* (ODBC) method that basically deals with bridge language technique so that the performance of minimal-resource SMT systems could be enhanced.

4.7.1 Optimized Direct-Bridge Combination Method

As it has previously been mentioned during this chapter, to alleviate the parallel data scarceness, a conventional solution introduces a bridge (pivot) language so that source and target languages could be connected and the scarceness of the parallel data could be alleviated as well. This strategy is usually applied in the situations where large amounts of source-bridge and bridge-target parallel corpora are available.

If we seek for the best performing approach of the bridge language technique then it is called triangulation which helps in the construction of an induced new phrase-table so that source and target languages could be linked. The biggest issue encountered during the application of this approach is that the size of the bridge phrase-table is very large.

If we are indulged in a scenario where we have to deal with the parallel corpus between the source and target languages, we must try to improve the overall translation quality and coverage. However, this translation quality and coverage could only be improved if the direct model based on this parallel corpus is combined with a bridge model. So, increasing the information gain is a reason to propose the direct-bridge combination method.

In this section, a combination method of direct and bridge SMT models will be proposed by us. The basic reason for this proposal is to prevent the relevant portions of the bridge SMT model from interfering with the direct SMT model. We show positive results for our case-study, *Persian-Spanish* SMT on different direct training data sizes.

The approach proposed by us is similar to Domain Adaptation methods. These methods enable us to combine the training data from various sources and build a

single translation model. This single translation model is then used for the purpose of translating sentences into the new domain.

Various methods have been used to explore the domain adaptation within the field. Some of these methods focus on using the Information Retrieval (IR) techniques so that sentence pairs related to the target domain from a training corpus could be retrieved (Eck et al., 2004; Hildebrand et al., 2005). Other domain adaptation methods focus on creating a distinction between the examples of general and specific domain (Daume III, H. and D. Marcu, 2006). Schroeder (2007), during the similar scenario, used the multiple alternative decoding paths so that various translation models could be combined. They also made sure that the weights of these translation models are set using help from the Minimum Error Rate Training (MERT) (Och, F. J., 2003).

In our proposed scenario we generate a new source-target translation model which is in contrast to domain adaptation. Our method contains the phrase bridging (triangulation) technique from two models. But we also use the domain adaptation approach so that relevant portions of the bridge phrase-table could easily be selected. Furthermore, we improved the translation quality by combining these portions with the direct translation model. We also explore how to merge bridge and a direct model built from a given parallel corpora into an effective combination by using the optimized direct-bridge combination method. This combination will help us in enhancing the coverage and bringing an improvement to the translation quality.

We take the information that is gained through the relevant portions of the bridge model and then try to maximize it. The information use by us do not interfere with the trusted direct model. So in order to achieve our purpose, we further ponder over the notion of categorizing the bridge phrase pairs. Later on, we divide these bridge phrase pairs into five different categories in accordance with their relation to the existence of source or target phrases in the direct model.

The phrase pairs included in the first category (*cat-1*) present a combination of the source and target phrases in the direct system. The second category (*cat-2*) is a bit different from the first category. The only similarity between both of the categories is that both of them contain the source and target phrases. However, the source and target phrases in the second category are not merged as a phrase pair in the direct system. The third (*cat-3*), fourth (*cat-4*) and fifth (*cat-5*) categories represent the presence of source and target phrase only but none of them are involved in the direct system.

Different categories demonstrated within the Table 4.5 show portions that have been derived from the bridge phrase-table. These categories have been included in the Table 4.5 with their labels which will help us with our results.

TABLE 4.5: Phrase pairs categorization of the portions extracted from the bridge phrase-table.

| Bridge phrase pairs cat | Src in direct | Trg in direct | Src and Trg in direct |
|-------------------------|---------------|---------------|-----------------------|
| cat-1 | ✓ | ✓ | ✓ |
| cat-2 | ✓ | ✓ | ✗ |
| cat-3 | ✓ | ✗ | ✗ |
| cat-4 | ✗ | ✓ | ✗ |
| cat-5 | ✗ | ✗ | ✗ |

4.7.2 ODBC Method Experiments

Here we will be presenting our results for the research conducted on our combination method between direct and bridge models. During this research, we used the *Moses* phrase-based technique introduced by Koehn et al. (2007). This strategy is used for creating a link between the direct model and the different bridge portions.

Later on, we use an in-domain parallel corpus containing (200K) sentences and (5M) words that were derived from *Open-Subtitles* parallel corpus (Tiedemann, 2012) for the purpose of following the direct *Persian-Spanish* SMT model. We also construct two SMT models while conducting the bridging experiments. One model is used to create a translation from the *Persian* to *English* while the other model focused on translating from *English* to *Spanish*.

The *English-Spanish* parallel corpus contains almost (2M) sentences (approximately 50M words) that have been derived from the *Europarl* parallel corpus (Koehn, P., 2005). We use an in-domain *Persian-English* parallel corpus that contained almost about (165K) sentences and (4M) words derived from *TEP*⁷ parallel corpus (Pilevar et al., 2011).

We use *fast-align* tool-kit for the purpose of conducting the word alignment. In the case of *Spanish* language modelling, almost (200M) words were derived and used from the *Europarl* corpus, in combination with the *Spanish* side of our training data. We sought help from the *KenLM* tool-kit so that all the implemented language models could be inserted with 4-grams. In order to cater with the *English* language modelling, we sought help from the *English* side of the *Europarl* corpus with 4-gram LM through the *KenLM* tool-kit as well.

Moses phrase-based SMT system was specifically used for the purpose of conducting all these experiments. We also sought help from MERT when we are about to decode the weights optimization. In the scenario where we have to tackle with both the *Persian-English* and *English-Spanish* translation models, we optimize the weights through a set of (5000) sentences. These sentences were derived from the parallel corpus and were then randomly checked for each model. While dealing with all of the models, we take care to only use the maximum phrase length of size (6), across all models.

Afterwards, we report the results on an in-domain *Persian-Spanish* evaluation set. This set included almost (500) sentences and two references. We conducted the evaluation by using the BLEU metric.

The phrase-based *Moses* provides us with the flexibility to use the multiple translation tables in the case of direct-bridge combination method experiments. During the scenario where translation options are collected from one particular table while other tables are used for the purpose of collecting the additional options, we use the *Couple* during the combination technique. However, the fact is that we can make our selection from the various options of combination techniques.

⁷The first free *English-Persian* parallel corpus, provided by the Natural Language and Text Processing Laboratory, University of Tehran, Iran.

If in any case, one translation option (identical source and target phrases) is found in multiple tables then we would create separate translation options for each occurrence. However, the score for each translation option will also be kept different.

4.7.2.1 Baseline Systems Evaluation

We compare the performance of sentence bridging (transfer) method against phrase bridging (triangulation) method with different filtering thresholds.

Generally, the triangulation method outperforms the transfer one even when we use a small filtering threshold of size (100). Moreover, the higher the threshold the better the performance but with a diminishing gain.

We use the best performing set-up across the rest of the experiments which is filtering with a threshold of (10K). The results are presented in Table 4.6.

TABLE 4.6: Transfer method versus triangulation with different filtering thresholds (100/1,000/10,000).

| Bridge scheme | BLEU score |
|----------------------------------|------------|
| Transfer | 20.21 |
| Triangulation (filtering 100) | 20.64 |
| Triangulation (filtering 1,000) | 21.18 |
| Triangulation (filtering 10,000) | 21.57 |

Figure 4.5 is a learning curve of the bridging systems which shows the increasing performance of the bridge-based systems between sentence bridging approach and three different sizes of phrase bridging approach through BLEU score.

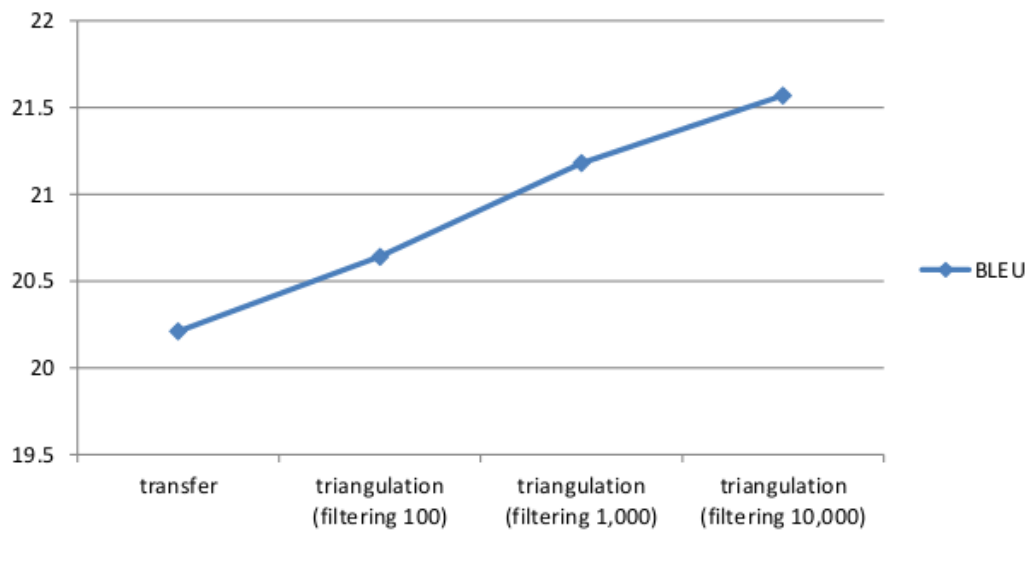


FIGURE 4.5: The performance learning curve of the bridging systems.

4.7.2.2 Baseline Combination

We start by the basic combination approach and then explore the gain/loss achieved from dividing the bridge phrase-table to five different categories. Table 4.7 illustrates

the results of the basic combination in comparison to the best bridge translation model and the best direct translation model.

TABLE 4.7: Baseline combination experiments between best bridge baseline and best direct model.

| Translation system | BLEU score |
|--|------------|
| Direct | 22.45 |
| Triangulation (filtering 10,000) | 21.57 |
| Interpolated (Direct+Triangulation filtering 10,000) | 22.81 |

As an interesting observation from the above table, direct translation system has a better performance than triangulation by filtering 10K sentences. The reason is related to the large size of parallel corpus for training direct system. In comparison with the previous set of experiment we can see that the difference between training data sizes have a direct effect on the performance of direct translation systems. The results shows that combining both models leads to a gain in performance.

Now the problem is finding a possibility to improve the quality by doing a smart choice of only relevant portion of the bridge phrase-table. We can overcome this problem through our proposed direct-bridge combination method.

Figure 4.6 shows the performance learning curve of the three translation systems mentioned in Table 4.7 according to BLEU score.

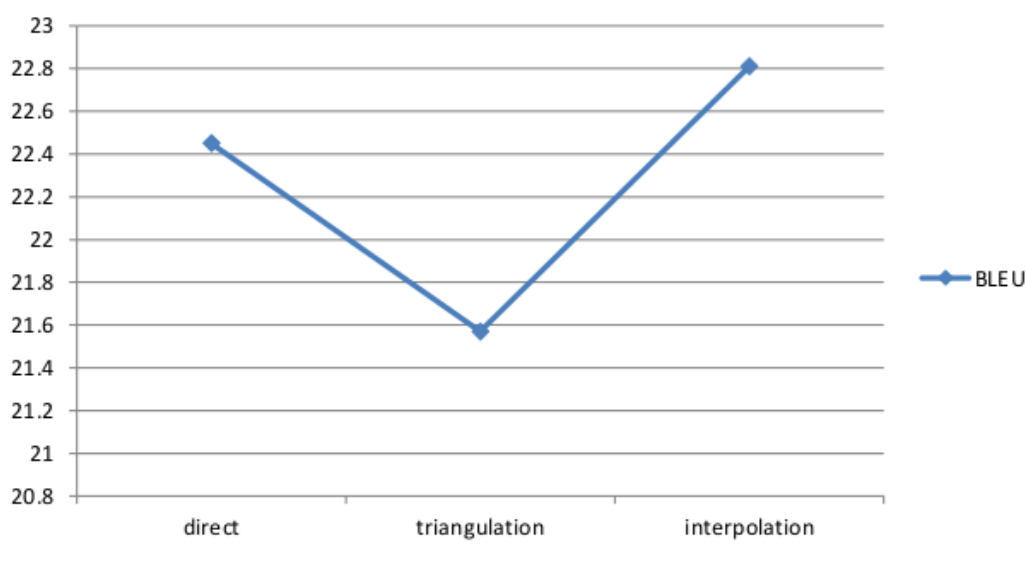


FIGURE 4.6: The learning curve of the bridging systems comparing the performance of direct, triangulation, and interpolated systems.

4.7.2.3 Direct-Bridge Combination

In this portion, we will ponder over the idea of creating a division of the bridge phrase pairs into five different categories. This division will be done according to the existence of source or target phrases within the direct system.

We first conduct a discussion of the results and then reveal the trade off that occurs between the quality of translation and the size of the different categories. These categories have been derived from the bridge phrase-table.

Table 4.8 reveals the results of the direct-bridge combination method experiments that have been demonstrated on the learning curve of 100% (200K sentences), 25% (50K sentences) and 6.25% (12.5K sentences) of the *Open-subtitles* parallel *Persian-Spanish* corpus.

TABLE 4.8: ODBC experiments results.

| Translation models | 12.5K sentences | 50K sentences | 200K sentences |
|----------------------|-----------------|---------------|----------------|
| direct | 15.85 | 20.01 | 22.45 |
| triangulation | 19.89 | 20.18 | 21.57 |
| baseline combination | 21.72 | 22.09 | 22.81 |
| cat-1 | 17.38 | 20.20 | 21.96 |
| cat-2 | 18.53 | 20.58 | 22.06 |
| cat-3 | 17.54 | 20.19 | 22.76 |
| cat-4 | 18.32 | 20.93 | 23.14 |
| cat-5 | 19.97 | 21.64 | 22.45 |

In the mentioned table, the first rows are revealing the outcome of the direct system. The second row reveals that outcome that we have gained from the best bridge system (triangulation). The third row reveals the outcome of the baseline combination experiments conducted along with the pattern of whole bridge phrase-table. Furthermore, the next set of rows reveals the results of our direct-bridge combination method experiments that have been derived on the basis of a different categorization. All scores are highlighted in BLEU. The bold scores has been used to mark a statistically significant result against the direct baseline system.

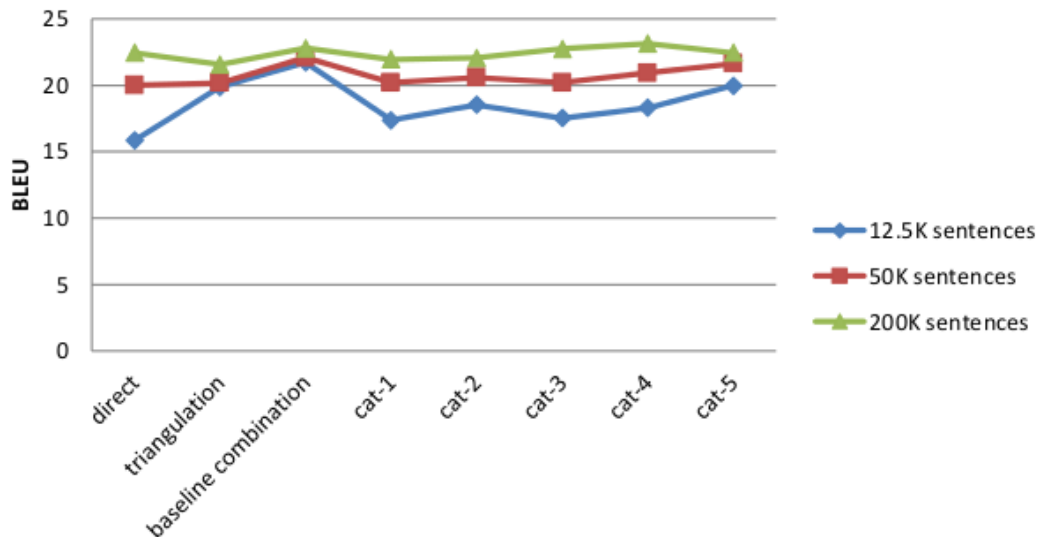


FIGURE 4.7: The performance of direct, triangulation, and interpolated models versus all types of direct-bridge combination proposed method.

Figure 4.7 illustrates the learning curve of the performance changes between all the translation systems considered for our proposed method and comparison with the other translation systems.

4.8 Discussion

The results further reveal that bridging is basically a technique considered to be robust because no or small amount of parallel corpora is present in it. When the direct translation model and the bridge translation model merge with each other in order to form a base combination, they end up giving a boost to the translation quality across the learning curve. So it can simply be expected that we will gain more from this combination if we use the smallest form of parallel corpus.

The results also reveal that some of the bridge categories provide more information gain in comparison to the other categories. It also happens sometimes that some of the categories damage the entire quality. For instance, (*cat-1*) and (*cat-2*) both heavily contribute towards damaging the quality of translation if they are combined with direct model that has gained training on 100% of the parallel data (200K sentences).

We have also gained an interesting observation from the results and that is we can achieve a better performance in comparison to a model trained on four times the amount of data (50K sentences) if we construct a translation system with only 6.25% of the parallel data (12.5K sentences).

Another most important point that we derive from the learning curve is that if the source phrase in the bridge phrase-table does not exist in the direct model then we can easily achieve the best gains. Such an expectation arises in the scenario where by conducting an addition of the unknown source phrases, we succeed in decreasing the overall Out-Of-Vocabularies (OOVs).

Creating a reduction in the bridge phrase-table is considered to be an additional benefit when we relate it with the proposed direct-bridge combination method. If we analyse the Table 4.9 then we will come to know that the percentage of phrase pairs is basically derived from the original bridge phrase-table so that each bridge class across the learning curve could properly be denoted.

TABLE 4.9: Percentage of phrase pairs extracted from the original bridge phrase-table for each bridging category.

| Model | 12.5K sentences | 50K sentences | 200K sentences |
|-------|-----------------|---------------|----------------|
| cat-1 | 0.1% | 0.1% | 0.2% |
| cat-2 | 16% | 29% | 35.2% |
| cat-3 | 64.1% | 63.3% | 59.9% |
| cat-4 | 6.1% | 3.4% | 2.3% |
| cat-5 | 13.7% | 4.3% | 2.3% |

At this point, the group of the phrase pairs is extracted in the form of categories. This is done in order to make it clear that source phrases exist in the direct model which makes the least contribution. These source phrases also damage the overall combination performance sometimes.

The Direct-bridge combination method with target only category provides comparatively better results in BLEU while hugely reducing the size of the bridge phrase-table used (2.3% of the original bridge phrase-table), if it is viewed in accordance with large parallel data (200K sentences). However, in the case of smaller parallel data, the advantage is comparatively decreased but two new tools are introduced including the trade off between the quality of the translation and the size of the model.

We can easily create an improvement in the translation quality of minimal-resource SMT systems if the optimized direct-bridge combination method between bridge and direct models are proposed. We revealed that this method can result in creating a large reduction of the bridge model without affecting the performance in any positive way.

4.9 Summary

In this chapter we investigated the idea of bridge language technique to respond to minimal parallel-resource bottleneck reality. First, the transfer method and the triangulation one introduced, then a performance comparison between them demonstrated as well. Our experimental results showed that phrase-level bridging (triangulation) is the best bridging technique and outperforms the sentence-level bridging (transfer). Later, the interpolated model investigated as well to enhance the translation quality for minimal parallel-resource language pairs.

After that, we investigated the proposed improvement in bridge language technique. We applied the approaches of interpolation bilingual text as well as combination phrase-tables. We saw the improvement of translation performance from a low-resource language (source) by converting it into a high-resource language (bridge) with the help of bilingual text containing a limited number of parallel sentences for source-bridge and large bilingual text for bridge-target.

Finally, we proposed an optimized direct-bridge combination scenario to enhance the translation performance, and also we analysed the effects of this scenario on our considered case-study minimal parallel-resource SMT system. In this scenario we generated a new source-target model. Our method contains the triangulation technique from two models. We improved the translation quality by combining these portions with the direct model. We also explored how to merge bridge and a direct models built from a given parallel corpora into an effective combination using the direct-bridge combination method. This method help us in enhancing the coverage and bringing an improvement to the translation quality.

Chapter 5

Round-Trip Training Scenario for Minimal Parallel-Resource SMT

Relatively several Statistical Machine Translation (SMT) research has been reported on languages that lack resources, such as monolingual text, parallel text, translation dictionaries, syntactic and semantic parsing tools, which in the literature are often referred to as low-density or minimal parallel-resource. Even though obtaining the monolingual text data has become easier recently due to the advancement of the internet, still obtaining parallel text is a difficult task.

This chapter includes the most important contribution of the thesis which deals with the minimal-resource situation. In this situation, only the limited amount of bilingual text is considered. However, the large amount of monolingual text is also available for the purposes of source language and target one as well. In simple words it can be said that, this chapter deals with the research conducted based on retraining mechanism for the minimal parallel-resource SMT. These particular systems are used so that the translation quality could be improved.

It has previously been explained that SMT systems are heavily dependent on parallel data. This means that SMT system does not work in the situation where bilingual text lines are available in fewer than the million numbers. If by any chance, the bilingual text is small, then the performance of statistical models become very poor. It happens because of the phrase counts and sparse words that define their parameters. It must be kept in mind that tens of millions of bilingual sentence pairs are required due to the fact that SMT is making progress lately. Still, this fact cannot be denied that human labelling is a very expensive task.

We reviewed the approaches to tackle this training data bottleneck in detail while going through the Chapter 2. The approaches that we studied are the Active Learning (AL), Semi-Supervised Learning (SSL), Bridge (pivot) Language Technique, Bilingual Lexicon Induction, Monolingual Collocation, and Domain Adaptation with Monolingual Data. On the other hand, while going through the Chapter 4, we studied Bridge Language Technique in detail. This technique is considered to be a common approach that is used for the purpose of overcoming the training data scarcity in detail. We also presented an interesting proposal that revolved around the technique for making effective use of third language as bridge.

In the current chapter we introduce a *round-trip training scenario*. This scenario is introduced as a novel training mechanism so that SMT system relevant to the automatic learning from the unlabelled data through a two-way game can be enabled through it.

5.1 Introduction

The SMT systems that are considered to be the state-of-the-art, tend to rely on the aligned parallel training corpora. However, it must be kept in mind that collecting such parallel data in practice is very expensive. Another irritating fact is that they are usually available in limited scale which leads towards the constrained applications and research. Due to the presence of unlimited monolingual data in the Web, the performance of SMT systems can easily be boosted by leveraging that data.

Various methods have been presented for this purpose that can easily be grouped into two categories:

1. In the first category the training of the language model is conducted through the help of target monolingual corpora. Later on, it is integrated with the SMT model. This SMT model receive their training from the parallel bilingual corpora so that their translation quality could be improved.
2. The second category includes the pseudo bilingual sentence pairs that are basically created through the help of monolingual data. This creation becomes possible when training of the translation model is conducted from the aligned parallel corpora. Later on, the training data is enlarged through the pseudo bilingual sentence pairs for the purposes of subsequent learning.

The methods mentioned above could definitely bring an improvement to the SMT performance, still there are certain limitations. There are certain important points that must be kept in mind for instance; the methods in the first category train the language models through the help of monolingual data only. Even though the second category methods are able to increase the size of the parallel training data, they do not have any sort of control on the quality of the pseudo bilingual sentence pairs.

Our proposed round-trip training mechanism is inspired by the following observation; There are two translation tasks related to the SMT and they are: source-target translation task (forward direction), and target-source translation task (backward direction). The forward direction is used as an outbound-trip against the backward one which is basically used as an inbound-trip. There are certain significant traits of these outbound and inbound trips: they can contribute towards the formation of a closed loop, and they can help in generating the informative feedback signals so that the translation models could be trained.

If we analyse the round-trip training mechanism it becomes clear that one translation engine is used to represent the model for the outbound translation task, while we use the other translation engine so that model for the inbound translation task. Then they will be asked to provide guidance to each other through a learning process. The two models can be iteratively updated until convergence through the help of the feedback signals created during the entire process.

The round-trip training scenario that was proposed by us can leverage monolingual data in the most effective way and influential way possible. This can be done with both the source language and the target one. The mechanism proposed by us can enable this data to play a role that is similar to the parallel bilingual data. This helps in the gradual reduction of the requirement on parallel bilingual data during the training process. In common words, the round-trip training mechanism for SMT can be described as the following two-engine communication game:

- Before moving forward, we must keep in mind that the first translation engine understands the X language only. Due to its understanding of the language, it sends a message in language X to the second translation engine¹. This is done through a noisy-channel which helps in the conversion of the language X message to language Y through the help of a translation model.
- The second translation engine understands language Y only. This is the reason that it receives the translated message in this language only. After checking the message, it sends a notification to the first translation engine even if that message consists of the natural sentence belonging to the language Y . At the next step, it sends back the received message to the first translation engine. This is done through another noisy-channel. This noisy-channel then helps in converting the language Y message back to language X through the help of another translation model.
- After the first translation engine receives the message from the second one, it checks the message and then sends a notification to the second translation engine. It strictly follows this ritual irrespective of the fact that the message it has received is consistent with the original message or not. After receiving the feedback, both of the translation engines will know about the performance of the two communication channels and the two translation models. As a result of this feedback, they make the required changes.

This two-way communication game can also be started from the second translation engine which would contain the original message in language Y . After making a start from the Y language, both of the translation engines will then follow the symmetric process. They will also make changes as per the feedback they receive.

The above-mentioned descriptions teach us that although the two translation engines may not carry the bilingual corpora in aligned form, they are still entitled to receiving the feedback about regarding the quality of both of their translation models. As a result of this feedback, they can make collective improvement to the models.

This two-way communication game can be played for number of rounds that is considered to be arbitrary. So it can easily be said that both of the translation models will be improved while proceeding with the learning procedure. This method enables us to develop a learning framework for training SMT model with the help of a round-trip training algorithm.

This translation scenario further enables us to use the learning framework so that training could be provided to the translation models from the unlabelled data. Our work not only provides with the new chance to learn the translation method from the scratch but it also reduces the requirement on the aligned bilingual data. According to the results acquired from the experiments, our method seems like a promising method.

As the round-trip training scenario is basically a novel training approach, its structure contains both of the bootstrapping methods including self-training and co-training.

It contains self-training in its structure because of the fact that translation produced for the purposes of monolingual source sentences is actually produced by the

¹The second translation engine may not be able to verify the correctness of the translation since the original message is invisible to it

forward model (outbound-trip translation task). Later on, this translation is used to retrain itself. The structure of round-trip training scenario also includes the co-training because of the backward model (inbound-trip translation task). This model delivers a signal so that good translations from the *k-best* list of translation candidates could be selected and used for the retraining purposes of the forward model. For understanding of this procedure deeply, let's start from the bootstrapping methods analysis.

5.2 Bootstrapping Analysis

Statistical approaches used for Information Extraction (IE) and Natural Language Processing (NLP) tasks consist of vast amount of information required for the purpose of producing high-quality results and performing reliably. Here we must keep in mind that huge amount of corpora is required for the training of empirical algorithms. It is quite clear that algorithms require their training data to be annotated so the salient textual features could be extracted and learned. This strategy is used because corpus of annotated is not sufficient for the purpose of training.

Most of the NLP problems do not automatically get divided into various source texts (views). In such a case, it is mandatory to construct these views artificially through arbitrary feature divisions (Nigam and Ghani, 2000). On the other hand, translation carries natural division of views onto the labels. When we see the SMT in detail, it will become clear that labels are in fact the target translations for source texts. This is the reason that source text can be regarded as a view on the translation.

Other types of views that are also used for the purpose of producing a translation include existing translations of source text in different languages. For instance, if we take two different languages into consideration such as *Spanish* and *German* translation of its text, then both of them could be used as different views. However, either of them could be used for the purpose of creating target translation into *English*. When these views are labelled with their translations, they can help in training the learners for statistical translation models.

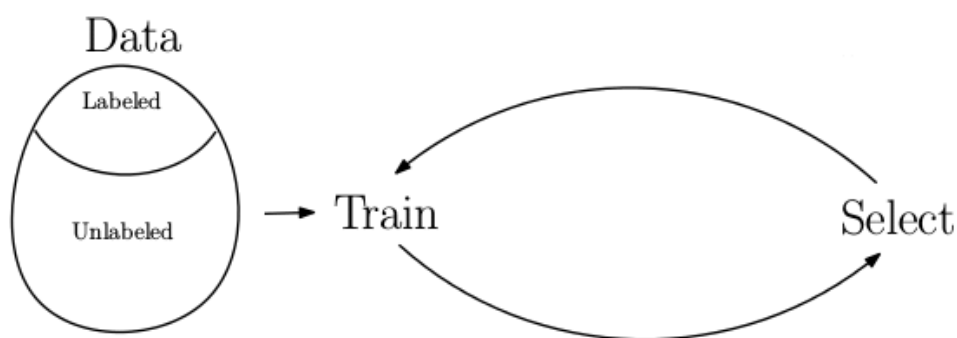


FIGURE 5.1: Bootstrapping high-level overview.

Throughout the history, Machine Learning has focused on three of the main learning patterns including supervised learning, unsupervised learning, and semi-supervised learning. However, during the recent years, another paradigm known by the name of *weakly-supervised learning* has started receiving attention (Zhou, 2017).

Weakly-supervised learning is basically a setting that contains the training data which differs from the test data. For instance, if we analyse the task of POS-tagging, we will get to see that test data includes the labelled sequences while training data is different as it contains the tagged word types. This entire idea is considered to be different from domain adaptation. Because in domain adaptation, even though the test and training data are drawn from the different populations but they belong to the same type.

There are various problems that have been suggested during the recent years so that the issue of weakly-supervised learning could be resolved. Here it must be kept in mind that these methods can be divided into two different categories. However, the bootstrapping methods must be excluded from these categories. These two categories are:

- The methods that bootstrap from a small number of tokens. These methods are also known as prototypes.
- The methods that constrain the underlying unsupervised learning problem.

This section focuses on two of the weakly-supervised algorithms i.e. *Self-training* and *Co-training*. Both of these are in fact the bootstrapping methods. These bootstrapping methods aim to improve the system's performance. They start their mission from a small set of labelled examples during which they also consider one or two weak classifiers (translation models) and make improvement through the incorporation of unlabelled data into the training dataset.

The motivation that compels us to use weakly supervised learning such as self-training and co-training in the case of complicated tasks is stronger than in the case of simple classification tasks. So if we need to achieve high-performance for SMT, then we will definitely require training data in large amount. However, the necessary labelled training data is available in limited quantity plus there are certain costs attached to the manual assembling.

Due to the fact that SMT is a technique that focuses on parallel corpora for the purpose of inducing bilingual dictionaries and translation rules in an automatic way, a statistical model of the translation process can be approximated after the relative orderings of the texts is analysed. If the SMT systems need to achieve the level of translation quality that is acceptable then they must be trained on large corpora. However, the issue is that large bilingual corpora are not easily available. This is the reason that in order for SMT to exist between languages without parallel corpora, one of these two options must be selected:

- Assembling of additional parallel corpora is required.
- The current SMT techniques must be used for the linguistic resources in limited amount.

When we use the bootstrapping methods, the above mentioned needs are blended together. Large amount of parallel corpora is produced after combining small amounts of parallel text.

Most of the machine learning techniques are properly supervised. This is the reason that they heavily rely on labelled training data. In the case of supervised learning algorithms, we must take the class labelled examples as an input and learn from them how to predict the class labels for new unlabelled data. The end result of

the learning method arrives in the form of a model or a predictive function. Supervised learning is considered to be the most successful approach for automatic text categorization.

SMT is included into the category of supervised learning². Due to the fact that labelled data must be created from unlabelled data, the amount of unlabelled data available is frequently greater than the amount of labelled data. This is the reason that certain interest has developed in the field of weakly supervised learning where the unlabelled data is used along with the labelled data. Weakly supervised learning tries to create reduction in the cost that is associated with the automatic annotation of data by the learners.

If we start with a set of labelled and unlabelled data, it will become clear that the main aim of a bootstrapping algorithm is to improve the classification performance. It achieves this motive by extracting text (examples) from the unlabelled data and integrating them into the labelled data set. The class distribution in the labelled data is maintained within each iteration. This purpose is fulfilled by keeping a constant ratio across classes between the examples that have already been labelled and between the examples that have recently been included.

The basic idea of this step is that introducing imbalance in the training dataset should be avoided. Two different types of views (two different classifiers C_1 and C_2) are required by the algorithm for the purpose of co-training. These views help with the interaction during the bootstrapping process. If in any case, the number of views is limited to one (one general classifier C_1) then co-training transforms into a self-training process. Self-training process is the process where one single classifier learns as a result of its own output.

Self-training has always been used in NLP research because of its importance as a single-view semi-supervised learning method. The term self-training refers to a variety of schemes that are used for unlabelled data. Ng and Cardie (2003) implemented self-training by using the bagging and majority voting strategy. For instance, a committee of classifiers is first trained on the labelled texts. Later on, it classifies the unlabelled texts independently. The texts where the classifiers give same label are included to the training set and then these classifiers are retrained. This procedure keeps on repeating until it reaches the point where final condition is met. In simple words, self-training methods use the labelled data for the purpose of training an initial model and then that model is used to label the unlabelled data and retrain a new model.

Co-training as a multi-view learning case, is another weakly supervised paradigm which increases the amount of labelled data by using the large amount of unlabelled data. The basic idea of co-training is to check whether any sort of redundancy is present in the unlabelled data or not. Intuitively, this data can assist in representation with the help of two or more separate, but redundant views such as disjoint feature subsets. The two classifiers trained on two views of the data can help each other, by adding one's most confident texts into the other's training set. Due to the fact that these data are as informative as the other random texts it is better that the training should proceed. The important assumptions related to the applicability and effectiveness of this method are as follows:

1. Both of the feature sets are sufficient to classify the data in the own accordance.

²This category involves labelled sentences with their translations.

2. Both of the feature sets are conditionally independent when it comes to the class label.

In the case where these two assumptions are effective, Blum and Mitchell, (1998) proved that co-training can start from weak classifiers and can proceed by learning from both labelled and unlabelled data.

Another research conducted by Nigam and Ghani (2000) revealed that when the conditional independence and information redundancy assumptions are effective, co-training algorithms beat all the other algorithms by using the unlabelled data.

The close association between self-training and co-training increase the amount of labelled data by automatically annotating unlabelled dataset. After we limit the number of classifiers and the number of views to one, the co-training gets converted into self-training. This is the scenario during which a classifier learns from its own output. The main parameters for these bootstrapping procedures are as follows:

1. The number of iterations (I).
2. The pool size (P)³.
3. The growth speed (K)⁴.

If we explain in simple words then it can be said that co-training is different from self-training because it depends on multiple learners to perform the annotation. However, it must be emphasized that both co-training and self-training are bootstrapping methods.

The basic aim of both these methods is to improve the performance of a supervised learning algorithm through the incorporation of large amounts of unlabelled data within the training data set.

Algorithm 5 illustrates the general bootstrapping process:

Algorithm 5 General bootstrapping

Input: Training set of labelled text (L), Training set of unlabelled text (U), Classifiers (C_i).

- 1: Create a set of pool (U') by choosing P random dataset from U .
 - 2: **for** Iteration= I **do**.
 - 3: Use L to individually train the C_i , and label the text in U' .
 - 4: For each C_i select most confidently dataset (G) and add them to L , while maintaining the class distribution in L .
 - 5: Refill U' with text from U , to keep U' at a constant size of P dataset.
 - 6: **end for**
-

5.2.1 Self-Training Mechanism

In this part we move towards investigating the self-training, as a weakly supervised learning approach. This approach explores the use of monolingual source text along with names of the documents to be translated in order to bring an improvement to the under-resource SMT systems' performance.

³Which is the number of texts selected from unlabelled data for annotation in each iteration. One can label all the unlabelled data or only label a subset of it every time.

⁴Which is the number of texts added to the labelled set in each iteration.

An initial version of the translation system is used to translate the source text. Target sentences of low-quality are automatically identified and discarded within the generated translations. The reliable translations along with their sources are used as a new bilingual corpus for the purpose of providing training to an additional phrase translation model. This is the reason that translation system can be adapted to the new source data even if no other bilingual data is available in this domain.

Self-training is a method that is supervised by a single learner who retrain on the labels that it applies to unlabelled data itself. Due to this, self-training method is understood as a weakly supervised method. This approach basically focuses on the translation model that would be trained for a language pair from a source-target parallel corpus. It would then move on to producing the target translations for a set of source sentences.

In the next step, the machine translated source-target sentences would be included to the initial bilingual corpus. As a result the translation model would be retrained.

The basic idea on self-training was at first documented by Yarowsky (1995). The self-training technique included an initial translation model (classifier) that was mostly trained from the available labelled data. The classifier first labels the unlabelled data and then a metric is applied to decide which of predictions are trustful. Instances that have been labelled with the best confidence are then included to the former labelled data to obtain a new training set. A new classifier is trained by obtaining the current labelled data, and this process is then repeated until it meets the final condition.

Self-training is a practical recommended method that is applicable in the situations where the existing supervised model is hard to modify. Most of the NLP tasks have been treated with self-training, such as word sense disambiguation, spam detection, and machine translation.

A self-training method depends on the monolingual source-language data for the purpose of improving the SMT systems' performance. This procedure includes the following steps:

- Use an existing SMT system in order to translate the new source text.
- Estimate confidence of resulting translations.
- Identify reliable translations based on confidence scores.
- Train new translation model on reliable translations. Later on, this function can be used as an additional feature in the existing SMT system.

This procedure then helps the translation system to adapt to source-language text of a new type⁵ without requiring any sort of parallel training or development datasets in the target language.

Algorithm 6 shows is the general procedure of self-training mechanism:

⁵e.g., text discussing new topics not present in the text originally used to train the system, or employing a different style.

Algorithm 6 Self-training

Input: Training set of labelled data (L), Training set of unlabelled data (U), Underlying Classifier (C), The number of iteration times (t), The number of selected unlabelled data for next iteration (θ), The selection metric (M), The selection function ($S(U_t, \theta, C, M)$), and The maximum number of iterations (max).

- 1: $t = 0$.
 - 2: $L_t = L, U_t = U$. // Where L_t and U_t are the labelled and unlabelled data set at the t^{th} iteration.
 - 3: **repeat:**
 - 4: train C on L_t ;
 - 5: $S_t = S(U_t, \theta, C, M)$; // Where S_t is the selected unlabelled dataset;
 - 6: $U_{t+1} = U_t - S_t$;
 - 7: $L_{t+1} = L_t + S_t$;
 - 8: $t = t + 1$;
 - 9: **until** U_t is empty. // max reached.
-

At this point we must keep a note that the selection function is mainly used for the purpose of ranking the unlabelled dataset. It is further used to select a certain number of unlabelled dataset so that training instance set for the next iteration could be updated. The function is not only influenced by the underlying classifier (current translation model) that should have good ranking performance, but also affected by the selection metric.

Assuming that the SMT system, which is existing at the moment will be used for the translation of for instance *newswire* text and a large collection of such data becomes available on the other side in the source-language. Then this entire method can assist in the creation of additional training dataset. Once we adapt to some test dataset, the chances of the identification of the relevant parts of the new source data are quite high.

The process that helps in retraining the SMT system with respect to its own translations of a test dataset is the same process that adapts the system to this test dataset. This test corpus reinforces the certain phrases in the prevailing phrase-tables. These phrase-tables are necessary for the translation of the new data.

After the trash machine translated texts are filtered out from the translation machine, the high-quality phrases are reinforced only. As a result of this reinforcement, the chances of the occurrence of low-quality phrase pairs decrease. The most common examples of low-quality phrase pairs include overly confident singletons or noise in the table.

At this phase, it must be kept in mind that reliable parts for given test dataset are kept in focus by the probability distribution over the phrase pairs. This method comes in use for the purpose of converting the prevailing system to a new domain where there are no development data sets or bilingual training available.

Self-training also provides us with the option of a system that can provide assistance in learning the new phrase pairs. Assuming that the source-phrases including VW and XYZ independently materialize in the parallel text, so they can train the original phrase-table $P(s_1^j | t_1^i)$. If these source-phrases continuously materialize in the monolingual source-language data with the sequence of $VW XYZ$ then there is a high-chance that new source-phrases such as VWX , $WXY Z$, and $VWXY Z$ could be produced.

If the target-language translations that are produced for the purpose of these translations offer even a minimal reliability, then the learning and placement of new bilingual phrases into the phrase-table is highly encouraged. In such an instance, the system to learn translations of unknown source-language words is not allowed to occur in the new data. Only those words appear in the bilingual dataset that are already present in the phrase-tables.

The basic limitation of this approach is that it only encourages the learning of compositional phrases. It is not possible for this system to translate certain idioms properly into another language such as *it is raining horses and camels*. For the sake of argument we can say that "What if proper translation for *it is raining* and *horses and camels* is available in the phrase-table?". "Can we translate this idiom then?". The answer is no. Even if proper translation for this idiom is available in the phrase-table, still it cannot be translated by the system.

Generally speaking, during the process of self-training the source-side dataset is translated using the MT system. After which the trustworthy translations are automatically identified. Together with the help of their sources, these sentences then form a new bilingual text which is used for the purpose of training the new translation models. This provides us with the method that helps in adapting the existing SMT system to a new domain even in the cases where no bilingual training or development data sets from this domain is available.

5.2.2 Co-Training Mechanism

In this part we investigate the co-training method as a type of multi-resource translation theory, for SMT. At this point, independent views on the data are required for the co-training where each view is sufficient enough for the labelling task. This is the reason that source strings in multiple languages as views on translation are used.

Co-training (Abney, 2002) is another weakly supervised learning technique that heavily depends on having distinct views of the items being classified. It means that the features used by some learners to label an item must be divisible into independent views. On the top of that each view must be sufficient enough in itself for labelling the items.

The application of this approach has been done to the simple categorization tasks such as web page classification (Blum and Mitchell, 1998), base noun phrase identification (Cardie and Pierce, 1998), and named entity recognition (Collins and Singer, 1999). This approach was also applied to the task of parsing (Sarkar, 2001). MT is considered to be much more complex task if compared with previous applications of co-training.

In MT, source-strings can clearly be seen as labelled with their corresponded translations. These labels are not made up of a small finite number of symbols as are made in the classification tasks or parsing. In fact, the labels are regarded in terms of vocabulary items in the target language.

The motivation for using simple classification tasks is not as strong as used for weakly supervised learning such as co-training for complicated tasks or MT. That is why, we need a large amount of training data in order to achieve high-performance with SMT. However, the issue is that necessary labelled training text is limited and we might have to encounter various costs associated with manually assembling more data.

So the only desirable option is to use co-training for the purpose of automatically creating more labelled training data for such problems. However, this option can only be achieved provided if the labelled training data could be made to fit into a framework of different views required by co-training.

The informal description of co-training can also be provided in the following:

1. At first, we need to select two or more views of a classification problem.
2. Next, we need to create separate models for each view, and then train each of created models in accordance with the small set of labelled data.
3. After that, we must search for a sample from an unlabelled data, so we could find examples for each model to label independently.
4. The examples that have been labelled with high-confidence are then selected to be the new training data.
5. Later on, the models are re-trained on the updated training data⁶.

By picking the labelled data from each model for adding to the training data, one model is labelling data for the other. This procedure is completely in contrast to the self-training where a model is retrained only on the labelled data that are produced by it (Nigam and Ghani, 2000).

Actually, co-training uses small amount of human labelled data so that larger sets of automatically labelled training data could be bootstrapped. During this approach, multiple learners are used for the purpose of labelling new data and retraining each other's labelled data.

The use of multiple learners increases the opportunities of including useful information. The important point is that an example which is easily labelled by one learner might be difficult for the other. This is the reason that adding the confidently labelled data will provide information in the next training round.

Co-training for SMT is considered to be more complicated because in this case multiple translation models to translate a bilingual or multilingual corpus are used rather than using a single translation model to translate a monolingual text. For instance, translation models could be trained for *German-English*, *French-English*, and *Spanish-English* from appropriate bilingual corpora, and then used to translate a *German/French/Spanish* parallel data into *English*. Since there are three *English* translation candidates for each sentence alignment, the best translation out of these three can be selected and used to retrain the models.

There are three different views involved in the co-training formulations:

- **Vocabulary acquisition:** One of the biggest problems that arise as a result of small training dataset is the incomplete word coverage. Without a word occurring in its training corpus it is highly unlikely that a translation model will generate a reasonable translation of it. Because of the fact that initial training corpora can come from different sources, the chances of a translation models collection to have encountered a word before are more likely. This results in vocabulary acquisition during the process of co-training.

⁶Here we must keep in mind that the procedure is repeated until the unlabelled data is exhausted.

- **Coping with morphology:** The problem mentioned above becomes extremely severe due to the fact that most current SMT formulations carry an incomplete treatment of morphology. This problem would arise from the fact that if training data for a *Spanish* translation model contained the masculine form of an adjective instead of feminine. Because of the issue that languages vary in their use of morphology, one language's translation model might possess the translation of a particular word form whereas other language's translation model would not. Thus co-training can lead to the increase in inventory of word forms and reduce the problem that morphology poses to statistical translation models that are simple.
- **Improved word order:** Word reordering problem is considered to be a significant reason of errors in SMT (Och, 1999). The word order between related languages is considered similar while word order between distant languages may differ significantly. If we require the translation models for distant languages to better learn word order mappings to the target language then its best to include more examples through co-training with related languages.

In all the above mentioned cases, the diversity afforded by multiple translation models increase the chances that were added by the machine translated sentences to the initial bilingual corpora.

The basic requirement of co-training is a set of unlabelled data. When we use the unlabelled data then it can provide us with two huge benefits:

- It can automatically be labelled by the learners
- It can be used for retraining

Algorithm 7 shows is the general overview of co-training mechanism:

Algorithm 7 Co-training

Input: Training set of labelled data (L), Training set of unlabelled data (U), Underlying Classifier (C), The texts in unlabelled data which are labelled positive (N_p), and The texts in unlabelled data which are labelled negative (N_q).

- 1: Train classifiers C_1 and C_2 on L .
 - 2: **repeat:**
 - 3: **for** each text $w=1,2$ **do**
 - 4: Remove N_p elements with greatest $C_w(p)$ from U , and add $(p + 1)$ to L .
 - 5: **end for**
 - 6: **for** each view $w=1,2$ **do**
 - 7: Remove N_q elements with smallest $C_w(p)$ from U , and add $(p - 1)$ to L .
 - 8: **end for**
 - 9: Retrain C_1 and C_2 using the updated L .
 - 10: **until** U becomes empty.
-

According to Algorithm 7, a subset of unlabelled data whose labels are assigned with high-confidence by the current classifiers (translation models) is selected in each iteration and they will be added to the set of labelled data in each iteration. The number of selected positive and negative instances is proportional to their ratio in the labelled sample. Then the classifiers are retrained based on this expanded training data. This process continues till it converges.

If we take a look at the Algorithm 7 then we will find that the selection method mentioned in that algorithm is unspecified. There are a certain methods that could help in selecting the best items for retraining. These methods usually include:

- Choosing the items containing unknown vocabulary.
- Making length-based selection.
- Choosing the translations possessing highest translation probabilities.

In simple words, it can also be said that co-training is a weakly-supervised learning technique because at the initial level it uses small amount of human labelled data. This data is used so that co-training could automatically bootstrap the automatically labelled training data in larger sets. When the co-training implementations are in process, multiple learners are used for the purpose of labelling new data and conducting retraining in accordance with the each other's labelled data.

The chances of the inclusion of useful information significantly improve if there is any increase in the use of multiple learners. Any example that might be easily labelled by one learner might not prove as easy for the other one. This is why including the confidently labelled data will prove to be quite informative in the next round of training.

5.3 Round-Trip Training Theory

Assume that we have given access to an initial bilingual corpus of a source-target language pair and also an access to a large or medium monolingual corpus in the source-side only.

If the size of our initial bilingual corpus is small, then the generated translation model from source to target and vice versa will be low-qualified, and if this low-qualified generated translation model is used to translate the mentioned large or medium source monolingual corpus to the target language, then the generated machine translated sentence pairs in the target-side will not produce high-qualified translation for their correspond source sentences.

Hence, if we pair this generated low-qualified machine translated sentence pairs in the target-side to the large or medium source monolingual corpus, and add this new generated pseudo bilingual sentence pairs to our small initial bilingual corpus, the quality of this added sentence pairs from the enlarged corpus will not be valuable and therefore, the noise of this section of the enlarged corpus will dominate the high-quality data of the whole enlarged corpus while retraining the translation system. Thus, it is more probable that the new generated model has lower quality than the former one.

Now the question is *how to separate the high-quality sentence pairs from all sentences in the generated pseudo bilingual sentence pairs?*. In another word, we are looking for selecting high-quality translations from all available sentences in the target-side of the pseudo generated bilingual sentence pairs, i.e. how to identify the high-quality translations from all noisy translations in the target-side of the pseudo generated bilingual sentence pairs?

In Chapter 2, some approaches have been mentioned in order to identify the high-quality translations from the whole noisy translations in the form of Semi-Supervised Learning (SSL) and Active Learning (AL).

In this section, a new solution named *round-trip training* is proposed to identify the high-quality translations from all the noisy ones, and a suitable way to optimize this round-trip training scenario in the generated pseudo bilingual sentence pairs is provided as well.

To translate the sentences from source to target, the idea of round-trip scenario is to find the high-quality sentences of source and target from among of whole noisy ones in the generated bilingual sentence pairs.

According to this idea we generate a *k-best* translation candidates list in the target-side for translating each source sentence to the target, and we expect that there is at least one high-quality translation for each correspond source sentence, in each *k-best* translations list.

Now the problem is finding this high-quality translation in each *k-best* list (applying outbound-trip and inbound-trip translation tasks). After finding the highest quality translation for each source sentence in the target-side of the pseudo bilingual corpus, we need to optimize this pseudo bilingual sentence pairs by finding the highest quality sentences in the source-side of the generated corpus (changing the translation path).

For this step, we need to generate a *k-best* list of candidates once again for each source sentence to find the high-quality sentences in the source-side of the generated bilingual sentence pairs. By doing so, as the result, we have the high-quality generated bilingual sentence pairs between source and target languages to adapt with the small initial bilingual source-target corpus, and retrain the new enlarged high-quality corpus via a translation system.

5.4 High-Quality Translations Selection

According to the round-trip training idea, to identify the high-quality translations part out of noisy translations in the target-side of the generated bilingual sentence pairs, considering two important points is essential. In fact according to the mentioned scenario, if a sentence as a translated one is considered a high-quality translation, it should consist of two characteristics as follow;

1. The considered sentence as an independent one in the target-language should be a well-formed sentence, i.e. in the target-language it should be an understandable and clear sentence even though it is not a correct translation of its correspond source. To evaluate this factor, target Language Model (LM) scores are being considered. Therefore, if the score of a sentence in a target-side under applying a language model is low, it is clear that this sentence is not a valuable sentence in the target-language and consequently it is not a suitable translation as well. For this reason, the first condition that the sentences are being considered as high-quality translations in the target-side for their correspond source sentences is that primarily these sentences should be suitable (well-formed) in the target-side. For instance, if we have access to a list of translation candidates in a target-language for a source correspond sentence, then we apply an *n-gram* LM to all those target-language sentences in the list, and rank those sentences by threshold according to their LM scores, those high-ranked sentences which are under the applied language model have higher scores, are selected as the high-quality sentences, and rest of the sentences will be considered as trash ones.

2. To consider a high-quality translation in the target-language, in addition to be a well-formed sentence in the target-side, this sentence should be a suitable (high-quality) translation for its correspond source sentence as well. This factor is being evaluated by the Translation Model (TM) scores in the inbound-trip (backward) translation direction (target-source), i.e. if we have access to a list of translations candidates in the target-side of a corpus for a source correspond sentence, we can do the backward translation process from each sentence of a list in target-side to the fixed correspond source sentence under a translation model, if the TM score of a sentence in this list to the mentioned source sentence is higher than the others in the inbound-trip (backward) translation process (from target to source), therefore, this sentence in the target-side is a high-quality translation for the source sentence. Hence, the second condition to have high-quality translations in the target-side is that we can regenerate the correspond source sentences through back translation model from target to source for each target sentence. Therefore, by having a suitable translation machine which can translate a sentence from the target-side to the source-side under a high TM score, we can show that the considered sentence in the target-side is not only a well-formed sentence in the target-side but also is a high-quality translation of its correspond source.

5.5 Round-Trip Training Mechanism

Now assume that we already generated a pseudo bilingual corpus between a source language and a target one, by having access to a large monolingual dataset in the source language, and using the source-target translation model extracted from a trained small initial bilingual corpus between the mentioned source-target language pair.

According to the round-trip training idea, first, we need to apply the outbound-trip translation task (forward translation process from source to target), and generate the *k-best* translation candidates lists instead of *single-best* translation for each correspond source sentence in the target-side.

In this stage we need to measure the target language model scores in order to check the well-formed characteristic for each sentence in each *k-best* list of translation candidates by applying an *n-gram* LM separately. After applying the *n-gram* LM, each sentence in each *k-best* list has an identified LM score. We keep these scores, and go for the next step at the same time.

Now we need to measure the translation model scores for each sentence (in each *k-best* list), through the backward translation direction. To do this, we need to apply the inbound-trip translation task (back translation process from the sentences in each *k-best* list in the target-side to source). After applying the inbound-trip translation model, we have an identified TM score for each sentence of each *k-best* translation candidates list to the correspond source sentence.

Each sentence in each *k-best* list has its own target LM score and inbound-trip TM score. In the next step of the round-trip training scenario, we need to combine both these LM and back-TM scores of each sentence in each *k-best* translation candidates list in the target-side to generate a total score ($R = LM\ score + TM\ score$). So, for each match, we will have;

$$R_1(x_1, y_1) \Rightarrow R_1 = (LM_{y_1} + TM_{y_1 \rightarrow x_1})$$

$$R_2(x_1, y_2) \Rightarrow R_1 = (LM_{y_2} + TM_{y_2 \rightarrow x_1})$$

...

$$R_k(x_1, y_k) \Rightarrow R_1 = (LM_{y_k} + TM_{y_k \rightarrow x_1})$$

For each match, we have a total score (R) in each k -best list. This total score is a combination of the target LM score, and the inbound-trip TM score for each sentence of a k -best list. So, in this stage we need to rank these R_i scores according to the threshold in order to recognize the highest R_i score between R_1 and R_k in each k -best list.

Based on this ranking we will select the highest R score for each list of translation candidates. Naturally the selected sentences with the highest R score include high-score inbound-trip TM and high-score target LM. Therefore, they are definitely reasonable and reliable.

In a simple word, for instance, for the sentence X_1 in the source-side of the pseudo bilingual sentence pairs, the correspond k -best translation candidates list is as $(X_1, Y_1), (X_1, Y_2), \dots, (X_1, Y_k)$. For each (X, Y_i) pair, we need to calculate the total score (R), which comes from the combination of backward-TM score of each pair, and target LM score of Y_i . (It means that, we need to apply an n -gram LM on each y_i sentence, and implement the backward translation process under the supervision of a phrase-based translation model for each pair of X_1 and Y_i). Then we need to rank the total scores (R_i), and select the highest total score among all (X, Y_i) pairs. We ignore the rest of the pairs with low R scores from the cycle.

The sentence with the highest total score will pair with its correspond source sentence. This procedure will repeat for each source sentence to examine their suitable correspond target sentence (until convergence).

At the end of this stage we have high-quality translations for each source sentence, but this generated bilingual data still suffers some noisy data, in other words this pseudo bilingual data is not optimized yet, and if we add this generated part to our initial high-quality bilingual data, the probable noise of the new part may dominate and capture the high-quality of the whole enlarged corpus and finally reduce the quality of the model. However, in the experimental framework, we will test the translation system using this low-quality data to compare the system's performance with the initial baseline translation system. So, in the next step, we need to optimize our generated pseudo bilingual corpus as well.

5.6 Round-Trip Training Optimization

There are many approaches toward optimization. Here, we will propose an interesting idea for this purpose, which is applying an n -gram LM only for the source sentences to find the best well-formed ones.

In this stage of the round-trip training scenario, we use the inbound-trip (backward) translation task. According to this idea, first, the translation path from source to target will be changed and convert to a new path (target to source). Then, we

need to generate the *k*-best lists of the translation candidates once again for all the new target (former source) sentences. Therefore, we apply an *n*-gram LM to each sentence of each *k*-best list, and re-rank the sentences of the *k*-best list according to their LM scores. The best well-formed sentence will be selected according to its high LM score.

Based on this fact, for each sentence in the new target-side (former source-side), we generate the *k*-best candidates list and apply an *n*-gram language model to all sentences in a *k*-best translations list, and then we rank these sentences according to their LM scores. Accordingly the sentence which holds the highest LM score in each *k*-best translations list will be recognized and selected.

Having this high-quality sentence, the rest of the sentences in each *k*-best translation candidates list will be ignored automatically, i.e. the best sentence in each *k*-best list will be selected according to its well-formed characteristic. In another word, this selected sentence is grammatically and understandably an ideal sentence in the considered new target (original source) language, so we can re-change the translation path to the initial state.

By doing so, the high-quality sentences in both source and target languages are selected. Now by pairing these high-quality source and target selected sentences, we have an optimized generated pseudo bilingual corpus, and we can add these optimized generated sentence pairs to the original small initial bilingual corpus in order to achieve an enlarged high-quality bilingual corpus for source and target languages.

At this stage, our generated large bilingual corpus is ready to be used for the purpose of retraining the baseline translation system. After retraining the system, we can compare the new results with the previous output from the initial baseline translation system.

Applying the round trip training idea shows that this method has a little of self-training technique in its structure, and co-training technique as well. It has self-training because the outbound-trip (forward) model generates translations for monolingual source sentences which are then used to retrain itself, and it also has co-training because the inbound-trip (backward) model gives signal by helping the translation system to select high-quality translations from the generated *k*-best translation candidates lists which are then used to retrain the forward model.

5.7 Round-Trip Training Algorithms

Assume that two monolingual corpora, C_X and C_Y which contain sentences from languages X and Y respectively are available⁷. On the other hand, imagine that we have access to two weak translation models that can translate sentences from X to Y and vice versa.

The goal of round-trip training scenario is to enhance the accuracy of these two translation models by using the monolingual corpora instead of parallel corpus. The basic idea is to leverage the round-trip training of the two translation models.

Beginning from a sample sentence in any monolingual data, we first translate it forward (applying outbound-trip translation task) to the other language, and then

⁷These corpora may have no topical relationship with each other at all.

further translate backward (applying inbound-trip translation task) to the original language. By evaluating this round-trip training results, we will get a sense about the quality of the two translation models, and be able to improve them accordingly. This process can be iterated for many rounds until both translation models converge.

Algorithm 8 shows the round-trip training procedure:

Algorithm 8 Round-trip training

Input: Monolingual dataset in the source and target languages (C_X and C_Y), Initial translation models in both outbound and inbound trips (TM_{X-Y} and TM_{Y-X}), Language models in both source and target languages (LM_X and LM_Y), The trade off parameter between 0 and 1 (α), The number of best-translations (K), The maximum iteration (T).

Output: Pseudo bilingual sentence pairs for source and target languages. // Which is not optimized yet.

- 1: **repeat:**
 - 2: $T = t + 1$.
 - 3: Sample sentences S_X and S_Y from C_X and C_Y respectively.
 - 4: Set $S = S_X$. // Updating the model for the round-trip communication game starting from language X .
 - 5: Generate K sentences ($S_{sample,1}, \dots, S_{sample,K}$). // Generating top-translations according to translation model; $P_{X-Y}(X|Y)$.
 - 6: **for** $k = 1, \dots, K$ **do**
 - 7: $R_{1,k} = LM_Y(S_{sample,k})$. // Set the target language model score for the k^{th} sampled sentence.
 - 8: $R_{2,k} = TM_{Y-X}(S|S_{sample,k})$. // Set the back translation model score for the k^{th} sampled sentence.
 - 9: $R_k = \alpha R_{1,k} + (1 - \alpha)R_{2,k}$. // Set the total score of the k^{th} sample sentence using the hyper-parameter.
 - 10: **end for**
 - 11: Set $S = S_Y$. // Updating the model for the round-trip communication game starting from language Y .
 - 12: Go through line 5 to line 10 symmetrically.
 - 13: **until** convergence.
-

Suppose corpus C_X contains N_X sentences, and C_Y contains N_Y sentences. Denote TM_{X-Y} and TM_{Y-X} as two statistical translation models as described in Section 3.2.1. Assume that we already have two well-trained n -gram language models for languages X and Y which are very easy to obtain since they only require monolingual data, each of which takes a sentence as input and output. A real value to indicate how confident the sentence is a well-formed sentence in its own language. Here the language models can be trained either using other resources, or just using the monolingual data C_X and C_Y .

For starting the round-trip communication game beginning with a sentence in C_X , denote S as a translation output sample. This step has an immediate score $R_1 = LM_Y(S_{sample})$, indicating how well-formed the output sentence is in language Y . Given that sample translation output, (S_{sample}), we use the probability value of S recovered from the S_{sample} as the score of the translation model.

Mathematically, $R_2 = TM_{Y-X}(S|S_{sample})$. We simply adopt the LM score and the back TM score as the total score, e.g.,

$$R_{total} = \alpha R_1 + (1 - \alpha) R_2 \quad (5.1)$$

Where α is an input hyper-parameter.

As the reward of the round-trip training game can be considered as a function of S , S_{sample} , and translation models TM_{X-Y} and TM_{Y-X} in both directions, we can optimize round-trip training scenario by changing the translation path, scoring the languages models of new target sentences and re-ranking them according to the threshold in order to select the highest quality sentences.

Algorithm 9 shows the optimization procedure of the round-trip training scenario:

Algorithm 9 Round-trip training optimization

Input: Generated pseudo bilingual sentence pairs (according to the output of Algorithm 8), The back translation model (TM_{Y-X}), The language model just in the source language (LM_X), The number of best-translations (K), The number of maximum iteration (T).

Output: Optimized generated pseudo bilingual sentence pairs for source and target languages. // Which is ready to add to the initial small parallel corpus for retraining the translation system.

- 1: Change the translation path from source to target ($X - Y$) to target to source ($Y - X$). // The new source language is now the former target one.
 - 2: **repeat:**
 - 3: $T = t + 1$.
 - 4: Sample sentence S_X and S_Y from the new target-side and the source-side of pseudo bilingual data respectively.
 - 5: Set $S = S_X$. // Optimizing the round-trip training model starting from language Y
 - 6: Generate K sentences ($S_{sample,1}, \dots, S_{sample,K}$). // Generating top-translations according to back translation model; $P_{Y-X}(y|x)$.
 - 7: **for** $k = 1, \dots, K$ **do**
 - 8: $R_k = LM_X(S_{sample,k})$. // Apply the n -gram LM on the sample sentence of the new target-side which is the former source-side, and set the new target LM score for the k^{th} sampled sentence.
 - 9: Re-rank each k sentences according to threshold.
 - 10: Select high-quality sentences according to the LM scores.
 - 11: **end for**
 - 12: Set $S = S_Y$. // Optimizing the round-trip training model starting from language X .
 - 13: Go through line 6 to line 11 symmetrically.
 - 14: **until** convergence.
-

Considering that random sampling sometimes brings unreasonable results in SMT. We use beam-search (Sutskever et al., 2014) to obtain more meaningful results (more reasonable sample translation outputs) by generate k -best high-quality sample translation outputs.

The optimized round-trip training game can be repeated for many rounds. In each round, one sentence is sampled from C_X and one from C_Y , and we update the two models according to the game beginning with the two sentences respectively.

5.8 Experimental Framework

We conduct two sets of experiments to test and prove the quality of our proposed optimized round-trip training scenario for phrase-based SMT systems.

We compare our optimized round-trip training method (optimized-SMT) with two translation systems; the first one is the standard phrase-based SMT system (baseline-SMT), and the second one is the phrase-based SMT which generates pseudo bilingual sentence pairs from monolingual corpora to assist the training step (pseudo-SMT).

We evaluate the proposed approach on two sets of language pairs; *Spanish-English* (and vice versa) as a large-scale language pair, and *Persian-Spanish* (and vice versa) as a minimal-resource one.

Our experiments show that optimized round-trip training technique works very well on *Spanish* \leftrightarrow *English* translation tasks as well as *Persian* \leftrightarrow *Spanish* ones. By learning from monolingual data, it achieves a comparable accuracy to phrase-based SMT trained from the full bilingual data for all the translation tasks.

5.8.1 Data Preparation

For the large-scale set of experiments, we use the *Spanish-English* bilingual corpora from *WMT13*, which contains approximately (10M) sentence pairs extracting from four different datasets; *Europarl* corpus, *News Commentary* corpus, *UN* corpus, and *Common Crawl* corpus. We also concatenate *news-test2011* and *news-test2012* as the validation and testing data sets respectively.

On the other hand, for the minimal-resource set of experiments we use the *Persian-Spanish* small bilingual corpora from *Tanzil* corpus, which contains about (65K) parallel sentence pairs⁸. We also use *Open-Subtitles2012*, and *Open-Subtitles2013* as the tuning and testing data sets respectively.

For both the large-scale and minimal-resource sets, we use the *Open-Subtitles2016* corpus, as large monolingual data.

5.8.2 Baseline Phrase-Based SMT Architecture

Very generally, the SMT paradigm has, as its most important elements, the idea; that probabilities of the source and target sentences can find the best translations. Frequently used paradigms of SMT on the log-linear model are the phrase-based, the hierarchical phrase-based, and the *n-gram* based. In our experiments we use the phrase-based SMT system with the maximum entropy framework (Berger et al., 1996):

$$\hat{t}_1^I = \arg \max_{t_1^I} P(s|t) \quad (5.2)$$

As we mentioned in Chapter 2, the phrase-based SMT model is an example of the noisy-channel approach, where we can present the translation hypothesis, t , as the target sentence (given s as a source sentence), maximizing a log-linear combination

⁸This is a collection of *Quran* translations compiled by the Tanzil project.

of feature functions:

$$\hat{t}_1^I = \arg \max_{t_1^I} \left\{ \sum_{m=1}^M \lambda_m h_m(s_1^J | t_1^I) \right\} \quad (5.3)$$

The Equation (5.3) called the log-linear model, where λ_m corresponds to the weighting coefficients of the log-linear combination, and the feature functions $h_m(s, t)$ to a logarithmic scaling of the probabilities of each model.

The translation process involves segmenting the source sentences into source phrases, translating each source phrase into a target phrase, and reordering these target phrases to yield the target sentence.

The decoder is used to search the most likely translation \hat{t} according to the source sentence, phrase translation model, and the target language model. The search algorithm can be performed by beam-search. The main algorithm of beam-search starts from an initial hypothesis. The next hypothesis can be expanded from the initial hypothesis which is not necessary to be the next phrase segmentation of the source sentence. Words in the path of hypothesis expansion are marked. The system produces a translation alternative when a path covers all words. The scores of each alternative are calculated and the sentence with highest score is selected. Some techniques such as hypothesis recombination and heuristic pruning can be applied to overcome the exponential size of search space.

5.8.3 Implementation

Generally, for the pseudo-SMT we use the trained phrase-based SMT model to generate pseudo bilingual sentence pairs from monolingual data, then remove the sentences with more than (80) words (for unifying the length of all sentences), optimize, and merge the generated data with the original parallel training dataset, and then retrained the model for testing.

Our proposed method needs a language model for each language. We train a *4-gram* language model based on *KenLM* for each language using its corresponding monolingual corpus. Then the language model is fixed, and the score of a received message is used to score the translation model.

While playing the round-trip training game, we initialize the channels using warm-start translation models (e.g., trained from initial small bilingual data corpora), and see whether optimized round-trip training system (optimized-SMT) can effectively improve the baseline-SMT accuracy.

In our experiments, in order to smoothly transit from the initial model trained from small bilingual data to the model training purely from monolingual data, we adopt the following strategy:

- At the beginning of the round-trip training process, we use half sentences from monolingual data and half sentences from bilingual data (sampled from the dataset used to train the initial model). The objective is to maximize the sum of the scores based on monolingual data.
- When the training process goes on, we progressively increase the percentage of monolingual sentences, until no bilingual data were used at all. Specifically,

we test one setting in each experiment i.e. for the large-scale language pair we use all the (10M) bilingual sentences pairs, and also for the minimal resource language pair we use all the (65K) parallel sentences. That is the warm-start model is learnt based on full bilingual data.

In the last step of the experiments, we need to retrain our baseline-SMT systems by enlarging the initial small bilingual corpus by adding the optimized generated pseudo bilingual sentence pairs to the initial parallel corpus. The new translation system (enlarged-SMT) contains both the initial and optimized pseudo bilingual corpora.

For each translation task we train our optimized round-trip training scenario. *Moses* package (Koehn et al., 2007), is employed for training our phrase-based SMT systems.

Through employing *Moses* decoder, *fast-align* approach (Dyer et al., 2013), is applied for word alignment. We employ *4-grams* language model for all SMT systems and they are developed by means of the *KenLM* tool-kit (Heafield, 2011).

In addition, we set the beam-search size to be (500) in the translation process, and the distortion limit to (6). All the hyper-parameters in the experiments are set by cross validation. We restrain the maximum target phrases to (6) that are loaded for each source phrase, and we draw on the same other default features of *Moses* translation engine.

For evaluating the systems performance we use the *BLEU* as the evaluation metric (Papineni et al., 2002).

5.9 Results Analysis and Evaluation

Four baseline systems for *Spanish-English* (and back translation) and *Persian-Spanish* (and back translation) are trained separately, while our optimized-SMT conducts joint training. We summarize the overall performances in Table 5.1.

TABLE 5.1: Translation results using BLEU for Spanish-English and back translation tasks.

| Translation Systems | Es-En | En-Es | Pe-Es | Es-Pe |
|---------------------|-------|-------|-------|-------|
| baseline-SMT | 34.92 | 36.27 | 27.45 | 26.80 |
| pseudo-SMT | 34.28 | 36.54 | 28.89 | 27.85 |
| optimized-SMT | 41.95 | 42.22 | 38.94 | 38.84 |

From Table 5.1 we can see that our optimized-SMT system outperforms the others in all the translation tasks.

For the task of translation from *Spanish* to *English*, optimized-SMT system outperforms the baseline-SMT one by about (7.03) BLEU points (the relative increase is approximately 12%), and outperforms pseudo-SMT system by about (7.67) BLEU points (the relative increase approximately is 12.2%). The improvement is significant. For the back translation from *English* to *Spanish*, our optimized-SMT system also outperforms baseline-SMT and pseudo-SMT ones by about (5.95) and (5.68)

BLEU points respectively (the relative increases are approximately 11.6% and 11.5% respectively).

On the other hand, for the other translation task from *Persian* to *Spanish*, we surprisingly with only (65K) bilingual data, the optimized-SMT achieves comparable translation accuracy as baseline-SMT. This system outperforms the baseline-SMT one by about (11.49) BLEU points (the relative increase approximately is 14.1%), and also outperforms pseudo-SMT system by about (11.05) BLEU points (the relative increase is approximately 13.4%). For the back translation direction from *Spanish* to *Persian*, the improvement of the optimized-SMT system compare with the baseline-SMT one is more significant. The optimized-SMT outperforms baseline-SMT and pseudo-SMT by about (12.04) and (10.99) BLEU points respectively (the relative increases approximately are 14.4% and 13.9%).

For both the large-scale language pair the minimal-resource one, optimized-SMT systems achieve comparable translation accuracy as baseline-SMT ones for all translation tasks. These results demonstrate the effectiveness of our optimized round-trip training scenario.

For each sentence in the test set, we translated it forth and back using the models and then checked how close the back translated sentence is to the original sentence using the BLEU score. We also used beam-search to generate all the translation results. Furthermore, we have the following observations:

- The pseudo-SMT systems outperform the baseline-SMT ones in all the experiments except in the case of *Spanish-English* translation direction because of the noise in the generated bilingual sentence pairs as mentioned in Section 5.5. The improvements of the pseudo-SMT systems over the baseline-SMT ones are significant in rest of the experiments. In fact, our hypothesis is to optimize the pseudo bilingual sentence pairs generated from the monolingual data by filtering and selecting the high-quality sentence pairs to get better performance for the pseudo-SMT systems. So, by doing optimization process we overcome the performance limitation of the pseudo-SMT system through the optimized-SMT one.
- When the size of our parallel bilingual data is small, the optimized-SMT makes larger improvement. This shows that the round-trip training mechanism makes very good utilization of monolingual data. Thus we expect the optimized-SMT will be more helpful for language pairs with smaller labelled parallel data. So the optimized-SMT opens a new learning view to translate from scratch.

Tables 5.2 shows the performances of two translation systems; the baseline SMT and the enlarged-SMT, that the latter one contains the training data sets of the baseline-SMT and the optimized-SMT as well.

TABLE 5.2: Comparing the baseline-SMT and the enlarged-SMT systems using BLEU for Spanish-English (and back translation) and Persian-Spanish (and back translation) tasks.

| Translation Systems | Es-En | En-Es | Pe-Es | Es-Pe |
|---------------------|-------|-------|-------|-------|
| baseline-SMT | 34.92 | 36.27 | 27.45 | 26.80 |
| enlarged-SMT | 40.88 | 41.07 | 36.33 | 37.13 |

It can be easily seen from Table 5.2 that the BLEU scores of our enlarged-SMT systems are much higher than baseline-SMT ones in all the cases. In particular, our enlarged-SMT outperforms baseline-SMT by about (5.96) BLEU points in the *Spanish-English* translation direction (the relative increase approximately is 11.7%), while outperforms by about (4.8) BLEU points in the back translation direction as well (the relative increase approximately is 11.3%). The enlarged-SMT also for *Persian-Spanish* translation task outperforms the baseline-SMT by about (8.88) BLEU points (the relative increase is approximately 13.2%), while the mentioned system outperforms the baseline-SMT one by about (10.33) BLEU points in *Spanish-Persian* translation direction as well (the relative increase is approximately 13.8%).

The significant improvements show that our proposed optimized round-trip training scenario not only is a promising technique but also is an ultra reliable approach to tackle with the training data scarcity limitation, and help us to improve the translation quality through SMT paradigm.

According to the experiments results we plot the BLEU scores with respect to the performance of different translation systems in Figure 5.2.

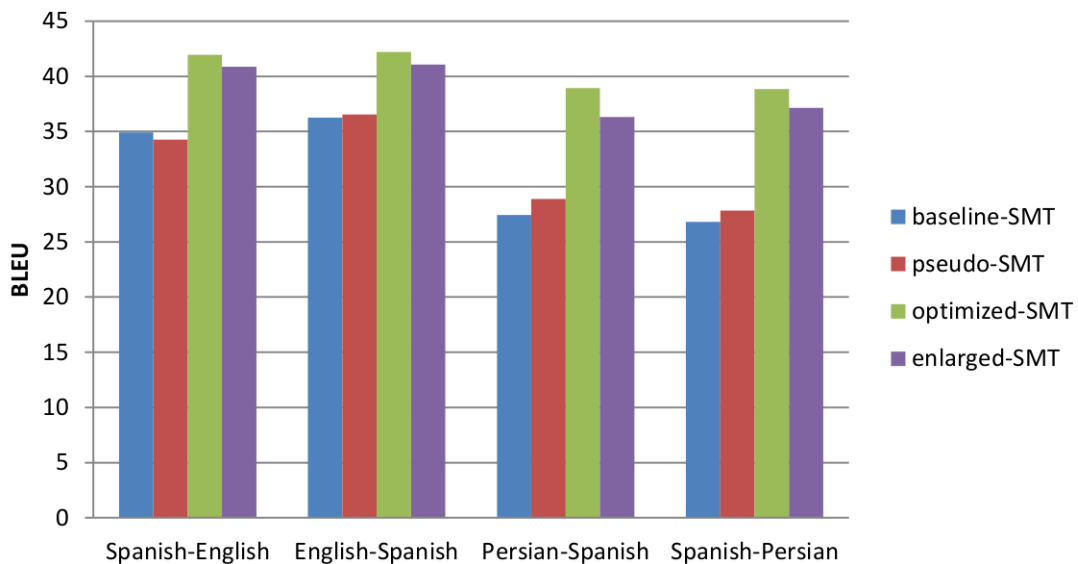


FIGURE 5.2: Performance comparison of the translation systems.

Figure 5.2 shows that after applying the retraining process, the enlarged-SMT systems (involves baseline-SMT and optimized-SMT training data sets) outperforms the baseline-SMT systems in all the translation tasks. Also this chart shows that, under all conditions, our optimized-SMT systems outperform all the other SMT systems.

So the proposed optimized round-trip training scenario works very well, specially for minimal parallel-resource language pairs.

Figure 5.3 illustrates the variation learning curve for translation systems in forward and backward directions.

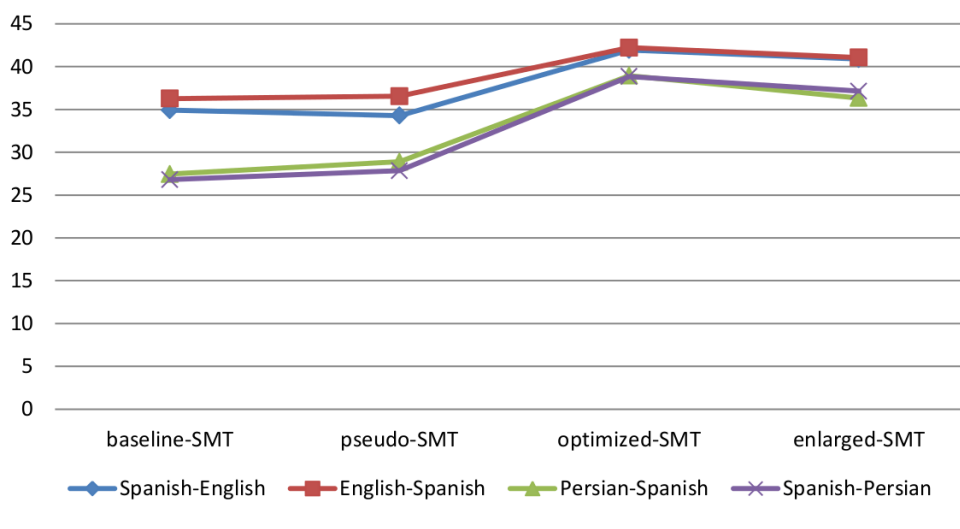


FIGURE 5.3: Learning curve of the translation systems performance.

5.10 Discussion

Although we have focused on SMT in our work, the basic idea of round-trip training scenario is generally applicable as long as two tasks are in round-trip form, we can apply this mechanism to simultaneously learn both tasks from unlabelled data using reinforcement learning algorithms.

Actually, many Artificial Intelligence (AI) tasks are naturally in round-trip form, for example, speech recognition versus text to speech, image caption versus image generation, question answering versus question generation, etc. It would be interesting to design and test round-trip algorithm for more dual tasks beyond SMT.

On the other hand, our technology is not restricted to two translation tasks only. Actually, our key idea is to form a closed loop so that we can extract feedback signals by comparing the original input data with the final output data. Therefore, if more associated tasks can form a closed loop, we can apply our technology to improve the model in each task from unlabelled data. For example, for an *English* sentence x , we can first translate it to a *Persian* sentence y , then translate y to a *Spanish* sentence z , and finally translate z back to an *English* sentence x . The similarity between x and x can indicate the effectiveness of the three translation models in the loop, and we can once again apply optimization methods to update and improve these models based on the feedback signals during the loop.

5.11 Summary

In this chapter we provided the main contribution of the current thesis. First we analysed the self-training as well as the co-training, as the bootstrapping methods in order to reach at a high-level insight and better understanding about the retraining concept with the help of systems' own outputs. Later on, we presented our proposed round-trip training technique based on retraining approach as a novel training method theoretically and algorithmically.

This proposed scenario is one of the optimal possibilities in order to overcome the data-scarcity bottleneck in training SMT systems; specially for those pairs of languages without enough Natural Language Processing (NLP) training resources.

We used two different language pairs; one pair with high training resources, and the other pair with low training resources. These pairs of languages are different with each other based on the amount of their training data availability. Our hypothesis is to prove that regardless the amount of training resources, the proposed technique works well and outperforms the baseline systems under all conditions.

All the results, charts, and learning curves show that the optimized round-trip training mechanism is promising and better utilizes the monolingual data. This proposed scenario has a competitive behaviour in comparison with other approaches related to state-of-the-art for minimal-resource SMT, and ironically in several cases it is preferable.

Chapter 6

Conclusions

In our research, we worked on improving the quality of Statistical Machine Translation (SMT) performance especially for those pairs of languages with minimal parallel-resources. We developed a learning framework based on retraining hypothesis as well as a bridging framework based on bridge language theory. We proposed methods to improve each component, separately.

In this chapter the research carried out and the improvements obtained in the context of this thesis are reviewed. Finally, some possible research directions for future work are discussed.

6.1 Overall Review

This thesis is about the topic of high-quality statistical machine translation systems for dealing with minimal parallel-resource language pairs entitled *Reliable Training Scenarios for Dealing with Minimal Parallel-Resource Language Pairs in Statistical Machine Translation*.

The main challenge we targeted in our approaches is parallel data scarcity, and this challenge is faced in different aspects. For the learning framework, first, we generated pseudo bilingual sentence pairs, then, we optimized these generated sentence pairs to identify high-quality parts of sentences. For the bridging framework, first, we selected the relevant portions of the bridge translation model that do not interfere with the more trusted direct translation model, then, we optimized our combination model in order to maximize the information gain.

Following is a summary of main contributions of this thesis:

- Phrase-based Translation Models Comparison:** We investigated the performance of two phrase-based translation models for SMT systems; the first one is Classical (standard) phrase-based translation model, and the other one is Hierarchical phrase-based translation model, on three language pairs, separately. For *Spanish-English* task and back translation, we showed that, under all conditions, the Hierarchical translation model outperforms the Classical one in the forward direction. However, in the backward direction yet the Classical model is preferable. For *English* and *Persian* language pair, the Classical model has better performance than the Hierarchical model in *Persian-English* translation task, while in *English-Persian* translation the Hierarchical model works well. Although for *Persian-Spanish* translation direction the Classical phrase-based system has a better performance than the Hierarchical phrase-based system, for the back translation task, the Hierarchical model outperforms the Classical one. In other words, we have shown that for the *Spanish-Persian* translation task, *Cdec* as an open-source Hierarchical decoder based on Synchronous Context-Free Grammars (SCFGs), is able to achieve high-quality translation

outputs than *Moses* that is based on Classical phrase-based model, 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 *Persian* language. Comparative performance of our considered language pairs' translation based on Classical and Hierarchical translation models conducts to set as state-of-the-art for further researches on phrase-based SMT. These comparative performance between the mentioned pairs of languages show that independent of applied statistical language models changes, the performance of translation models is directly influenced by the structure and word order of the source and target languages.

- **Direct-Bridge Models Combination Scenario:** We developed a smart technique to combine bridge and direct models. This scenario is based on bridge language technique and is an extension version of interpolated model by making effective use of domain adaptation methods. This proposed scenario is provided to effectively combine both a direct translation system and a bridge translation system built from a given parallel training dataset to achieve high-quality translation output. According to this scenario, we maximized the information gain by choosing the relevant portions of the bridge-based model that do not interfere with the direct-based model which is trusted more indeed. We showed that the proposed combination technique can lead to a large reduction of the bridge-based model without affecting the performance if not improving it. Also we have shown the effects of recent improvements on bridge language technique to enhance the minimal parallel-resource SMT performance i.e. bilingual text interpolation with possible repetitions of original bilingual text for balance, and phrase-table combination where each bilingual text is used to build separate phrase-table, as two general strategies in order to use bilingual text of one natural language to enhance SMT performance for some related languages are used.
- **Round-Trip Training Scenario:** We developed a learning framework by proposing this scenario. The round-trip training scenario is based on retraining approach to improve the performance of minimal parallel-resource SMT systems. According to this scenario, we automatically learnt from unlabelled data through a round-trip communication game. We used two independent translation systems in order to represent the model for either both forward and backward translation tasks, and then we asked them to learn from each other through a round-trip training process. These two translation tasks could form a closed loop, and generated informative feedback signals to train the translation models, even if without the involvement of a human labeller. The general idea of this scenario is to take advantage of the readily available generated text in building a high-quality SMT model in an iterative manner. Also we analysed Bootstrapping methods as well. We showed that how self-training and co-training, as two weakly-supervised learning algorithms, can improve the performance of translation systems. They start their mission from a small set of labelled examples during which they also consider one or two weak translation models and make improvement through the incorporation of unlabelled data into the training dataset. In round-trip training scenario we exploited bootstrapping methods; the outbound-trip (forward) model benefited self-training because this model produces translations for monolingual source sentences, which are then will be used to retrain itself, while the inbound-trip (backward) model generate signals which are using to retrain the outbound-trip model

by helping high-quality selected translations from the *k-best* translation candidates lists. Our proposed round-trip training scenario is a new category much like both self-training and co-training which are categories by themselves.

Some of the contributions of this thesis have been submitted and/or published in international conferences and journals. These are listed below:

Conference Papers:

- Benyamin Ahmadnia, Javier Serrano, and Gholamreza Haffari. (2017). "Persian-Spanish Low-resource Statistical Machine Translation System through English as Pivot Language". In Proceedings of *the 17th International Conference on Recent Advances in Natural Language Processing (RANLP 2017)*.
- Benyamin Ahmadnia and Javier Serrano. (2016). "Direct translation vs. Pivot language translation for Persian-Spanish Low-resourced Statistical Machine Translation System". In Proceedings of *the 18th International Conference on Artificial Intelligence and Computer Sciences (ICAICS 2016)*.
- Benyamin Ahmadnia and Javier Serrano. (2015). "Hierarchical Phrase-based Translation Model vs. Classical Phrase-based Translation Model for Spanish-English Statistical Machine Translation System". In Proceedings of *the 31st Conference on the Spanish Society for Natural Language Processing (SEPLN 2015)*.

Journal Articles:

- Benyamin Ahmadnia and Javier Serrano. (2017). "Employing Pivot Language Technique through Statistical and Neural Machine Translation Frameworks: The Case of Under-resourced Persian-Spanish Language Pair". *International Journal on Natural Language Computing (IJNLC)*. Vol.(6), No.(5).
- Benyamin Ahmadnia, Gholamreza Haffari, and Javier Serrano. (2017). "Round-Trip Training Scenario to Enhance the Performance of Minimal-Resource Statistical Machine Translation Systems". (*Submitted*)

6.2 Future Work Directions

There are a number of possible further improvements to the techniques presented in this thesis related to enhance the performance of minimal parallel-resource SMT systems.

In Chapter 4, we proposed the direct-bridge combination scenario between direct and bridge models to enhance the translation quality. We showed that this scenario can lead to a large reduction of the bridge model without affecting the performance if not improving it. In the future, we plan to investigate categorizing the bridge model based on morphological patterns extracted from the direct model instead of just the exact surface form.

Regarding the direct-bridge combination scenario for SMT systems with minimal parallel-resources, a direction to prune the bridge phrase-table, is to train a binary category on any available parallel data between source and target languages to prune bridge phrase pairs in a way that is directly related to the translation quality, and can take advantage of several feature functions that account for different aspects of phrase pair quality.

As another line, we are interested to explore some other features to assess produced phrase pairs quality between source and target languages, as a big potential for improvement our work on bridge language technique.

In Chapter 5, we provided our proposed round-trip training scenario to improve the performance of minimal parallel-resource SMT systems. We showed that this scenario can enable an SMT system to learn from unlabelled texts through a round-trip training game automatically, and our results prove that this technique works very well on low-resource as well as high-resource translation tasks. In the future, we plan to learn translations directly from monolingual texts of two languages (from scratch).

Regarding the round-trip training scenario for SMT systems with minimal parallel-data, the basic idea can also be applied to Neural Machine Translation (NMT) as well, and we will pay attention to this orientation. We will extend our approach to jointly train multiple translation models for more than two languages using monolingual data.

In recent years, NMT as a promising approach, has made rapid progress, and has improved the state-of-the-art in many settings of MT, but it requires large amounts of training data to generate reasonable output.

NMT as a suitable alternative to phrase-based SMT can be used for minimal parallel-resource languages as well as SMT, by introducing more local dependencies and using word alignments to learn sentence reordering during translation.

Bibliography

- Abdul-Rauf, S. and H. Schwenk (2009). "On the use of comparable corpora to improve SMT performance". In: *Proceedings of the 12th Conference of the European Chapters of the ACL (EACL)*, pp. 16–23.
- Abney, S. (2002). "Bootstrapping". In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Abney, S. and S. Bird (2010). "The human language project: building a universal corpus of the world's languages". In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 88–97.
- Ahmadnia, B. and J. Serrano (2015). "Hierarchical phrase-based translation model vs. classical phrase-based translation model for Spanish-English statistical machine translation system". In: *Proceedings of the 31st Conference on the Spanish Society for Natural Language Processing (SEPLN)*.
- Ahmadnia, B., J. Serrano, and G. Haffari (2017). "Persian-Spanish low-resource statistical machine translation through English as pivot language". In: *Proceedings of Recent Advances in Natural Language Processing (RANLP)*, pp. 24–30.
- Aho, A. V. and J. D. Ullman (1969). "Syntax directed translations and the pushdown assembler". In: *Journal of Computer and System Sciences* 3.1.
- Ahsan, A., P. Kolachina, D. Misra Sharma, and R. Sangal (2010). "Coupling statistical machine translation with rule-based transfer and generation". In: *Proceedings of the Annual Meeting and Symposium of the Antenna Measurement Techniques Association (AMTA)*.
- Al-Onaizan, Y., J. Curin, M. Jahr, K. Knight, J. D. Lafferty, D. Melamed, F. J. Och, D. Purdy, N. A. Smith, and D. Yarowsky (1999). *Statistical machine translation*.
- AleAhmad, A., H. Amiri, M. Rahgozar, and F. Oroumchian (2009). "Hamshahri: A standard Persian text collection". In: *Journal of Knowledge-Based Systems* 22.5, pp. 382–387.
- Aswani, N. and R. Gaizauskas (2005). "Aligning words in English-Hindi parallel corpora". In: *Proceedings of the Association for Computational Linguistics (ACL), Workshop on Building and Using Parallel Texts: Data-driven Machine Translation and Beyond*, pp. 115–118.
- Babych, B., A. Hartley, S. Sharoff, and O. Mudraya (2007). "Assisting translators in indirect lexical transfer". In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*.

- Bahdanau, D., K. Cho, and Y. Bengio (2014). "Neural machine translation by jointly learning to align and translate". In: *Journal of arXiv preprint arXiv:1409.0473*.
- Bahl, L., J. Baker, P. Cohen, F. Jelinek, B. Lewis, and R. Mercer (1978). "Recognition of continuously read natural corpus". In: *Proceedings of Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP*, pp. 422–424.
- Berger, A., S. A. Della Pietra, and V. J. Della Pietra (1996). "A maximum entropy approach to natural language processing". In: *Journal of Computational Linguistics* 22.1, pp. 39–71.
- Biemann, C., G. Heyer, U. Quasthoff, and M. Richter (2007). "The Leipzig corpora collection monolingual corpora of standard size". In: *Journal of Corpus Linguistic*.
- Birch, A., M. Osborne, and P. Koehn (2008). "Predicting success in machine translation". In: *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*.
- Bird, S. and G. Simons (2003). "Seven Dimensions of Portability for Language Documentation and Description". In: *Journal of LANGUAGE* 79, pp. 557–582.
- Blum, A. and T. Mitchell (1998). "Combining labeled and unlabeled data with co-training". In: *Proceedings of the 11th Annual Conference on Computational Learning Theory*.
- Borin, L. (1999). "Pivot alignment". In: *Proceedings of NODALIDA Workshop on Processing Historical Language*, pp. 41–48.
- Brown, P. F., J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin (1990). "A statistical approach to machine translation". In: *Journal of Computational Linguistics* 16.2, pp. 79–85.
- Brown, P. F., S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer (1993). "The mathematics of statistical machine translation: Parameter estimation". In: *Journal of Computational Linguistics* 19.2, pp. 263–311.
- Callison-Burch, C., D. Talbot, and M. Osborne (2004). "Statistical machine translation with word and sentence-aligned parallel corpora". In: *Proceedings of the Association for Computational Linguistics (ACL)*, pp. 175–182.
- Carbonell, J., S. Klein, and D. Miller M. Steinbaum (2006). "Context-based machine translation". In: *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, pp. 19–28.
- Cardie, C. and D. Pierce (1998). "Error-driven pruning of treebank grammars for base noun phrase identification". In: *Proceedings of the joint Conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics (COLING-ACL)*, pp. 218–224.

- Chen, B., M. Zhang, A. Aw, and H. Li (2008). "Exploiting n-best hypotheses for SMT self-enhancement". In: *Proceedings of Human Language Technologies: The 9th Annual Conference of the North American Chapters of the Association for Computational Linguistics (HLT-NAACL)*, pp. 157–160.
- Cherry, C. and D. Lin (2003). "A probability model to improve word alignment". In: *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Chiang, D. (2005). "A hierarchical phrase-based model for statistical machine translation". In: *Proceedings of Association for Computational Linguistics (ACL)*, pp. 263–270.
- Chiang, D. (2007). "Hierarchical phrase-based translation". In: *Journal of Computational Linguistics* 33.2, pp. 201–228.
- Cocke, J. (1969). *Programming languages and their compilers: Preliminary notes*.
- Collins, M. and Y. Singer (1999). "Unsupervised models for named entity classification". In: *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing (EMNLP), and Very Large Corpora*, pp. 100–110.
- Dabre, R., F. Comieres, S. Kurohashi, and P. Bhattacharyya (2015). "Leveraging small multilingual corpora for SMT using many pivot languages". In: *Proceedings of the Human Language Technology Conference of the North American Chapters of the Association of Computational Linguistics (HLT-NAACL)*, pp. 1192–1202.
- Daume III, H. and J. Jagarlamudi (2011). "Domain adaptation for machine translation by mining unseen words". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 407–412.
- Daume III, H. and D. Marcu (2006). "Domain adaptation for statistical classifiers". In: *Journal of Artificial Intelligence Research (JAIR)* 26, pp. 101–126.
- De Gispert, A. and J. B. Mariño (2006). "Catalan-English statistical machine translation without parallel corpus: bridging through Spanish". In: *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC)*, pp. 65–68.
- Dempster, A., N. Laird, and D. Rubin (1977). "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the Royal Statistical Society* 39, pp. 1–38.
- Deneefe, S. and K. Knight (2009). "Synchronous tree adjoining machine translation". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 727–736.
- Diab, M. and P. Resnik (2002). "An unsupervised method for word sense tagging using parallel corpora". In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 255–262.

- Dirix, P., I. Schuurman, and V. Vandeghinste (2005). "METIS-II: Example-based machine translation using monolingual corpora - System description". In: *Proceedings of the 2nd Workshop on Example-Based Machine Translation*.
- Dou, Q. and K. Knight (2013). "Dependency-based decipherment for resource-limited machine translation". In: *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1668–1676.
- Dou, Q., A. Vaswani, and K. Knight (2014). "Beyond parallel data: Joint word alignment and decipherment improves machine translation". In: *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, pp. 557–565.
- Dyer, C., A. Lopez, J. Ganitkevitch, J. Weese, H. Setiawan, F. Ture, V. Eidelman, P. Blunsom, and P. Resnik (2010). "cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models". In: *Proceedings of the 48th Annual Meeting of the Association for Computer Linguistics (ACL), System Demonstrations*, pp. 7–12.
- Dyer, C., V. Chahuneau, and N. A. Smith (2013). "A simple, fast, and effective reparameterization of IBM model 2". In: *Proceedings of the North American Chapters on Association for Computational Linguistics (NAACL)*, pp. 644–648.
- Eck, M., S. Vogel, and A. Waibel (2004). "Language model adaptation for statistical machine translation based on information retrieval". In: *Proceedings of the 4th International Conference on language resources and evaluation (LREC)*, pp. 327–330.
- Eck, M. and S. Vogel and A. Waibel (2005). "Low cost portability for statistical machine translation based on n-gram coverage". In: *Proceedings of MTSummit X*.
- Eisner, J. (2003). "Learning non-isomorphic tree mappings for machine translation". In: *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL)*, pp. 205–208.
- Federico, M., N., and M. Cettolo (2008). "IRSTLM: an open source toolkit for handling large scale language models". In: *Proceedings of INTERSPEECH*, pp. 1618–1621.
- Foster, G., R. Kuhn, and H. Johnson (2006). "Phrasetable smoothing for statistical machine translation". In: *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*.
- Fraser, A. and D. Marcu (2006). "Semi-supervised training for statistical word alignment". In: *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 769–776.
- Galley, M., J. Graehl, K. Knight, D. Marcu, S. DeNeefe, W. Wang, and I. Thayer (2006a). "Scalable inference and training of context-rich syntactic translation models". In: *Proceedings of the 44th Annual Meeting on Association for Computational Linguistics (ACL)*.

- Galley, M., M. Hopkins, and K. Knight (2006b). "What's in a translation rule?" In: *Proceedings of Human Language Technologies: The 7th Annual Conference of the North American Chapters of the Association for Computational Linguistics (HLT-NAACL)*, pp. 273–280.
- Gao, Q. and S. Vogel (2008). "Parallel implementations of word alignment tool". In: *Proceedings of the 46th Annual Meeting on the Association for Computational Linguistics (ACL). Software Engineering, Testing, and Quality Assurance for Natural Language Processing Workshop*, pp. 49–57.
- Goldwater, S. and D. McClosky (2005). "Improving statistical MT through morphological analysis". In: *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, pp. 676–683.
- Gross, A. (1992). *Limitations of computers as translation tools computers in translation: A practical appraisal*.
- Guzmán, F., S. R. Joty, L. Màrquez, and P. Nakov (2017). "Machine translation evaluation with neural networks". In: *Journal of CoRR* abs/1710.02095.
- Haffari, G., M. Roy, and A. Sarkar (2009). "Active learning for statistical phrase-based machine translation". In: *Proceedings of the Annual Conference of the North American Chapters of the Association for Computational Linguistics (NAACL)*, pp. 415–423.
- Heafield, K. (2011). "KenLM: Faster and smaller language model queries". In: *Proceedings of the 6th Workshop on Statistical Machine Translation*.
- Hildebrand, A. S., M. Eck, S. Vogel, and A. Waibel (2005). "Adaptation of the Translation Model for Statistical Machine Translation based on Information Retrieval". In: *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*.
- Hopkins, M. and J. May (2011). "Tuning as ranking". In: *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1352–1362.
- Huang, F. (2009). "Confidence measure for word alignment". In: *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics (ACL), and the 4th IJCNLP of the AFNLP*, pp. 932–940.
- Irvine, A. and C. Callison-Burch (2015). "End-to-end statistical machine translation with zero or small parallel texts". In: *Journal of Natural Language Engineering* 1.1, pp. 1–34.
- Irvine, A. and C. Callison-Burch (2016). "A comprehensive analysis of bilingual lexicon induction". In: *Journal of Computational Linguistics*.
- Jurafsky, D. and J. H. Martin (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition (Prentice Hall Series in Artificial Intelligence)*. 1st ed. Prentice Hall.

- Kasami, T. (1966). *An efficient recognition and syntax-analysis algorithm for context-free languages*.
- Kishida, K. and N. Kando (2003). *Two stages refinement of query translation for pivot language approach to cross lingual information retrieval: A trial at CLEF 2003*.
- Knight, K. (1999). "Decoding complexity in word-replacement translation models". In: *Journal of Computational Linguistics* 25.
- Koehn, P. (2004). "Statistical significance tests for machine translation evaluation". In: *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*. Vol. 4, pp. 388–395.
- Koehn, P., F. J. Och, and D. Marcu (2003). "Statistical phrase-based translation". In: *Proceedings of North American Chapters of Association for Computational Linguistics (NAACL)*, pp. 127–133.
- Koehn, P., H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst (2007). "Moses: open source toolkit for statistical machine translation". In: *Proceedings of the 45th Annual Meeting of the Association for Computer Linguistics (ACL)*, pp. 177–180.
- Koehn, P. (2005). "Europarl: A parallel corpus for statistical machine translation". In: *Proceedings of the 10th Machine Translation Summit*, pp. 79–86.
- Kumar, S., F. J. Och, and W. Macherey (2007). "Improving word alignment with bridge languages". In: *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing (EMNLP), and Computational Natural Language Learning*, pp. 42–50.
- Kunchukuttan, A., R. Pudupully, R. Chatterjee, A. Mishra, and P. Bhattacharyya (2014). "The iit bombay smt system for icon 2014 tools contest". In: *Proceedings of Natural Language Processing Tools Contest at ICON*.
- Lambert, P., H. Schwenk, C. Servan, and S. Abdul-Rauf (2011). "Investigations on translation model adaptation using monolingual data". In: *Proceedings of the 6th Workshop on Statistical Machine Translation*, pp. 284–293.
- Leavitt, J., D. W. Lonsdale, and A. M. Franz (1994). "A reasoned interlingua for knowledge-based machine translation". In: *Proceedings of Canadian Artificial Intelligence Conference*.
- Li, Z., C. Callison-Burch, C. Dyer, J. Ganitkevitch, S. Khudanpur, L. Schwartz, W. N. G. Thornton, J. Weese, and O.F. Zaidan (2009). "Joshua: An open source toolkit for parsing-based machine translation". In: *Proceedings of the 4th European Chapters of the Association for Computational Linguistics (EACL), Workshop on Statistical Machine Translation*, pp. 135–139.
- Liu, T., H. Wu, D. Dong, W. He, X. Hu, D. Yu, H. Wu, and H. Wang (2010). "Improving statistical machine translation with monolingual collocation". In: *Proceedings*

- of the 48th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 825–833.
- Liu, Y., Y. Huang, Q. Liu, and S. Lin (2007). “Forest-to-string statistical translation rules”. In: *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 704–711.
- Lopez, A. (2008). “A survey of statistical machine translation”. In: *Journal of Computing Surveys* 40.3.
- Lopez, A. and P. Resnik (2005). “Improved HMM alignment models for languages with scarce resources”. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 83–86.
- Marcu, D. and W. Wong (2002). “A phrase-based, joint probability model for statistical machine translation”. In: *Proceedings of Empirical Methods for Natural Language Processing (EMNLP)*, pp. 133–139.
- Martin, J., R. Mihalcea, and T. Pedersen (2005). *Word alignment for languages with scarce resources*.
- Marton, Y. and P. Resnik (2008). “Soft syntactic constraints for hierarchical phrasal-based translation”. In: *Proceedings of the Human Language Technology Conference of the North American Chapters of the Association of Computational Linguistics (HLT-NAACL)*, pp. 1003–1011.
- Matusov, E., M. Popovic, R. Zens, and H. Ney (2004). “Statistical machine translation of spontaneous speech with scarce resources”. In: *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, pp. 139–146.
- Matusov, E., G. Leusch, R. E. Banchs, N. Bertoldi, D. Dechelotte, M. Federico, M. Kolss, Y. S. Lee, J. B. Mariño, M. Paulik, S. Roukos, H. Schwenk, and H. Ney (2008). “System combination for machine translation of spoken and written language”. In: *Proceedings of IEEE: Transactions on Audio, Speech and Language Processing*, pp. 1222–1237.
- McKeown, K. R. and D. R. Radev (2000). “Collocations”. In: *Proceedings of the Robert Dale, Hermann Moisl, and Harold Somers (Ed.), A Handbook of Natural Language Processing*, pp. 507–523.
- Melamed, I. D., G. Satta, and B. Wellington (2004). “Generalized multitext grammars”. In: *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics (ACL)*.
- Mohit, B. and R. Hwa (2007). “Localization of difficult-to-translate phrases”. In: *Proceedings of the Second Workshop on Statistical Machine Translation*, pp. 248–255.
- Munteanu, D. S. and D. Marcu (2006). “Extracting parallel sub-sentential fragments from non-parallel corpora”. In: *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 81–88.

- Mustafa, S. H. (2005). "Character contiguity in N-gram-based word matching: the case for Arabic text searching". In: *Journal of Information Processing and Management* 41.4, pp. 819–827.
- Nakov, P. and H. T. Ng (2012). "Improving statistical machine translation for a resource-poor language using related resource-rich languages". In: *Journal of Artificial Intelligence Research (JAIR)*, pp. 179–222.
- Nettle, D. (1998). "Explaining global patterns of language diversity". In: *Anthropological archaeology* 17.4, pp. 354–374.
- Neubig, G. and K. Duh (2014). "On the elements of an accurate tree-to-string machine translation system". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 143–149.
- Ng, V. and C. Cardie (2003). "Weakly supervised natural language learning without redundant views". In: *Proceedings of the Meeting of the North American chapters of the Association for Computational Linguistics (NAACL)*, pp. 173–180.
- Niesen, S., H. Ney, R. Aachen, and R. Aachen (2004). "Statistical machine translation with scarce resources using morpho-syntactic information". In: *Journal of Computational Linguistics* 30.2, pp. 181–204.
- Nigam, K. and R. Ghani (2000). "Analyzing the effectiveness and applicability of co-training". In: *Proceedings of International Conference on Information and Knowledge Management (CIKM)*, pp. 86–93.
- Nirenburg, S., S. Beale, and C. Domashnev (1994). "A full-text experiment in example-based machine translation". In: *Proceedings of the International Conference on New Methods in Language Processing*, pp. 78–87.
- Och, F. J. (1999). *An efficient method for determining bilingual word classes*.
- Och, F. J. and H. Ney (2002). "Discriminative training and maximum entropy models for statistical machine translation". In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL)*, pp. 295–302.
- Och, F. J. (2003). "Minimum error rate training in statistical machine translation". In: *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL)*. Vol. 1, pp. 160–167.
- Och F. J. and H. Ney (2003). "A systematic comparison of various statistical alignment models". In: *Journal of Computational Linguistics* 29.1, pp. 19–51.
- Och F. J. and H. Ney (2004). "The alignment template approach to statistical machine translation". In: *Journal of Computational Linguistics*.
- Oepen, S., E. Velldal, J. T. Luning, P. Meurer, V. Rosen, and D. Flickinger (2007). "Towards hybrid quality-oriented machine translation. On linguistics and probabilities in MT". In: *Proceedings of the 11th Conference on Theoretical and Methodological Issues in Machine Translation*, pp. 144–153.

- Papineni, K., S. Roukos, and T. Ward (1998). "Maximum likelihood and discriminative training of direct translation models". In: *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (IEEE)*. Vol. 181. 1, pp. 189–192.
- Papineni, K., S. Roukos, T. Ward, and W. J. Zhu (2002). "BLEU: A method for automatic evaluation of machine translation". In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL)*, pp. 311–318.
- Paul, M., H. Yamamoto, E. Sumita, and S. Nakamura (2009). "On the importance of pivot language selection for statistical machine translation". In: *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapters of the Association for Computational Linguistics (HLT-NAACL)*, pp. 221–224.
- Paul, M., A. Finch, and E. Sumita (2013). "How to choose the best pivot language for automatic translation of low-resource languages?" In: *Proceedings of the ACM Transactions on Asian Language Information*.
- Pauls, A. and D. Klein (2011). "Faster and smaller n-gram language models". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 258–267.
- Pickhardt, R., T. Gottorn, M. Korner, and S. Staab (2014). "A generalized language model as the combination of skipped n-grams and modified kneser-ney smoothing". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1145–1154.
- Pilevar, M. T., H. Faili, and A. H. Pilevar (2011). "TEP: Tehran English-Persian parallel corpus". In: *Proceedings of 12th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing)*.
- Popovic, M. and H. Ney (2005). "Exploiting phrasal lexica and additional morpho-syntactic language resources for statistical machine translation with scarce training data". In: *Proceedings of the 10th Annual Conference of the European Association for Machine Translation (EAMT)*, pp. 212–218.
- Popovic, M., D. Vilar, H. Ney, S. Jovicic, and Z. Saric and (2005). "Augmenting a small parallel text with morpho-syntactic language resources for Serbian–English statistical machine translation". In: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 41–48.
- Quirk, C., A. Menezes, and C. Cherry (2005). "Dependency treelet translation: Syntactically informed phrasal SMT". In: *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 271–279.
- Ravi, S. and K. Knight (2011). "Deciphering foreign language". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 12–21.
- Sanchez-Cartagena, V. M., F. Sanchez-Martínez, and J. A. Perez-Ortiz (2011). "Enriching a statistical machine translation system trained on small parallel corpora

- with rule-based bilingual phrases". In: *Proceedings of Recent Advances in Natural Language Processing (RANLP)*, pp. 90–96.
- Saralegi, X., I. Manterola, and I. Vicente (2011). "Analyzing methods for improving precision of pivot based bilingual dictionaries". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 846–856.
- Sarkar, A. (2001). "Applying co-training methods to statistical parsing". In: *Proceedings of the Meeting of the North American chapters of the Association for Computational Linguistics (NAACL)*.
- Scannell, K. P. (2007). "The Crubadan Project: Corpus building for under-resourced languages". In: *Proceedings of the 3rd Web as Corpus Workshop*. Vol. 4, pp. 5–15.
- Schafer, C. and D. Yarowsky (2002). "Inducing translation lexicons via diverse similarity measures and bridge languages". In: *Proceedings of the Conference on Natural Language Learning (CoNLL)*, pp. 874–881.
- Schroeder, J. (2007). "Experiments in domain adaptation for statistical machine translation". In: *Proceedings of Association for Computational Linguistics (ACL)*, pp. 224–227.
- Senellart, J. and P. Koehn (2010). "Convergence of translation memory and statistical machine translation". In: *Proceedings of AMTA Workshop on MT Research and the Translation Industry*, pp. 21–31.
- Smith, J. R., C. Quirk, and K. Toutanova (2010). "Extracting parallel sentences from comparable corpora using document level alignment". In: *Proceedings of Human Language Technologies: The 11th Annual Conference of the North American Chapters of the Association for Computational Linguistics (HLT-NAACL)*, pp. 403–411.
- Snover, M., B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul (2006). "A study of translation edit rate with targeted human annotation". In: *Proceedings of Association for Machine Translation in the Americas*, pp. 223–231.
- Stolcke, A. (2002). "SRILM-An extensible language modeling toolkit". In: *Proceedings of International Conference on Spoken Language Processing*, pp. 257–286.
- Sutskever, I., O. Vinyals, and V. Q. Le (2014). "Sequence to sequence learning with neural networks". In: *Proceedings of Advances in Neural Information Processing Systems*, pp. 3104–3112.
- Tiedemann, J. (2012). "Parallel data, tools and interfaces in OPUS". In: *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC)*.
- Tinsley, J., M. Hearne, and A. Way (2009). "Exploiting parallel treebanks to improve phrase-based statistical machine translation". In: *Proceedings of the 10th International Conference on Computational Linguistics and Intelligent Text Processing*, pp. 318–331.

- Tofghi Zahabi, S., S. Bakhshaei, and S. Khadivi (2013). "Using context vectors in improving a machine translation system with bridge language". In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 318–322.
- Tufis, D., R. Ion, A. Ceausu, and D. Stefanescu (2005). "Combined word alignment". In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 107–110.
- Ueffing, N., G. Haffari, and A. Sarkar (2007). "Transductive learning for statistical machine translation". In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*, pp. 25–32.
- Utiyama, M. and H. Isahara (2007). "A comparison of pivot methods for phrase-based statistical machine translation". In: *Proceedings of the Human Language Technology Conference of the North American Chapters of the Association of Computational Linguistics (NAACL)*, pp. 484–491.
- Vilar, D., D. Stein, M. Huck, and H. Ney (2010). "Jane: Open source hierarchical translation, extended with reordering and lexicon models". In: *Proceedings of the Joint 5th Workshop on Statistical Machine Translation and Metrics (MATR)*, pp. 268–276.
- Wang, H., H. Wu, and Z. Liu (2006). "Word alignment for languages with scarce resources using bilingual corpora of other language pairs". In: *Proceedings of COLING-ACL, Main Conference Poster Sessions*, pp. 874–881.
- Wu, H. and H. Wang (2007). "Pivot language approach for phrase-based statistical machine translation". In: *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 856–863.
- Xiong, D., M. Zhang, A. Aw, and H. Li (2009). "A syntax-driven bracketing model for phrase-based translation". In: *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics (ACL) and the 4th IJCNLP of the AFNLP*, pp. 315–323.
- Yarowsky, D. (1995). "Unsupervised word sense disambiguation rivaling supervised methods". In: *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 189–196.
- Younger, D. H. (1967). "Recognition and parsing of context-free languages in Time". In: *Journal of Information and Control* 10.2, pp. 189–208.
- Zhang, B., D. Xiong, and J. Su (2016). "Recurrent neural machine translation". In: *Journal of CoRR* abs/1607.08725.
- Zhou, Z. H. (2017). *A brief introduction to weakly supervised learning*.
- Zhu, X., Z. He, H. Wu, and C. Zhu (2014). "Improving pivot-based statistical machine translation by pivoting the co-occurrence count of phrase pairs". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1665–1645.

Zollmann, A. and A. Venugopal (2006). "Syntax augmented machine translation via chart parsing". In: *Proceedings of the Workshop on Statistical Machine Translation*, pp. 138–141.