Mitigating the Problems of SMT using EBMT

Sandipan Dandapat

B. Tech, PGDCL, MS.

A dissertation submitted in fulfilment of the requirements for the award of

Doctor of Philosophy (Ph.D.)

to the

Dublin City University
School of Computing

Supervisors: Prof. Andy Way and Dr. Sara Morrissey

May 2012

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Ph.D. is entirely my own work, and that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed:

(Candidate) ID No.:

Date:

Contents

Abstract **x**

Acknowledgements **xi**

1 Introduction **1**

1.1 Research Questions 3

1.2 Roadmap 5

1.3 Publications 6

2 Background **9**

2.1 Example-Based Machine Translation 11

2.1.1 Approaches to EBMT 18

2.2 Statistical Machine Translation 23

2.3 Translation Memory 27

2.3.1 TM Technology 28

2.3.2 Synergy between MT and TM 30

2.4 Evaluation Metrics 33

2.4.1 BLEU 33

2.4.2 NIST 35

2.5 Data Sources 37

2.5.1 English–Bangla Patient Dialogue Corpus 37

2.5.2 Other Data 39

2.6 Summary 41

3 Proportional Analogy-Based EBMT **43**

3.1 The Underlying Idea of a PA-Based System 44

3.1.1 Proportional Analogies 44

3.1.2 Analogy-Based EBMT 45

3.1.3 Some Immediate Difficulties 46

3.2 Our Approach 47

3.2.1 System Architecture 48

3.2.2 Heuristics 49

3.2.3 Analogy Verifier 53

3.2.4 Analogy Solver 55

3.3 Experiments and Results 57

3.3.1 Experiments Conducted 58

3.3.2 Data Used for the Experiments 59

3.3.3 Results 60

3.3.4 Further Study with NE Transliteration 65

3.3.5 Observations 67

3.4 Summary 73

3.4.1 Contributions 75

4 EBMT using Templates **77**

4.1 Translation Templates 78

4.1.1 Similarity Translation Templates 79

4.1.2 Difference Translation Template 80

4.2 Our Approach 81

4.2.1 Learning Translation Templates 82

4.2.2 Translation Using Templates 88

4.3 Experiments and Results 91

4.3.1 Experiments Conducted 91

4.3.2 Data Used for Experiments 92

4.3.3 Results . . . 94
4.3.4 Observations . . . 97
4.4 Summary . . . 100
4.4.1 Contributions . . . 100

5 EBMT Using a Subsentential Translation Memory 102
5.1 Motivation . . . 103
5.2 Building a Subsentential Translation Memory . . . 105
5.3 Approach . . . 108
5.3.1 Matching . . . 108
5.3.2 Alignment . . . 110
5.3.3 Recombination . . . 113
5.4 Experiments . . . 114
5.4.1 Experimental Setup . . . 114
5.4.2 Data Sources . . . 115
5.4.3 Immediate Results and Observations . . . 116
5.5 Improvement . . . 118
5.5.1 System Combination . . . 118
5.5.2 Experiments and Results Using the Combined System . . . 119
5.6 Manual Evaluation . . . 121
5.7 Observations . . . 129
5.7.1 Assessment of Error Types . . . 131
5.7.2 Time Complexity . . . 134
5.8 Summary . . . 135
5.8.1 Contribution . . . 136

6 EBMT$_{\mathbf{TM}}$: Improving Scalability 137
6.1 Motivation . . . 138
6.2 Approach . . . 139
6.2.1 Grouping . . . 139

6.2.2 Indexing . . . 141
6.2.3 IR Engine . . . 142
6.3 Experiments . . . 144
6.3.1 Data Used for Experiments . . . 145
6.3.2 Results . . . 146
6.4 Observations and Discussions . . . 150
6.5 Summary . . . 152
6.5.1 Contribution . . . 154

7 Conclusion **156**
7.1 Contribution . . . 160
7.2 Future Work . . . 160

Bibliography **163**

List of Figures

2.1 The translation process of an EBMT system. 12

2.2 Aligned example from Kaji et al. (1992), with coupled Japanese–English word and phrase pairs identified by corresponding links. . . . 16

3.1 Analogy-based EBMT architecture. 48

3.2 Edit-distance matrix between the words *like* and *unlike*. 53

3.3 Pseudo-distance matrix between the words *like* and *unlike*. 54

3.4 Example of pseudo-distance-based analogy solver. 56

3.5 Analogy-based translation example from Chinese-to-English. 63

3.6 Translation output for English-to-Bangla system. 64

3.7 The effect of combined heuristics for NE transliteration using AEBMT system. 65

3.8 The effect of combined heuristics for NE transliteration using the AEBMT+SMT system. 66

3.9 The effect of running time (1 sec, 3 sec, 10 sec and 30 sec) in analogy-based EBMT (AEBMT) and in the combined EBMT_{TM} + SMT system with different heuristics and models. 69

3.10 Comparison of the total number of NEs transliterated, the total number of correct transliterations in the candidate output set and the correct number of transliterations at rank 1 for the no-heuristic setting, the H2 and H5 settings. 71

3.11 Erroneous Chinese-to-English translation at rank 1. 73

4.1 Example of matching based on edit-distance trace. 84
4.2 Decoding architecture. 89

5.1 Moses phrase equivalents with associated probabilities. 106
5.2 Moses lexical equivalents with associated probabilities. 107
5.3 Extraction of matched and non-matched segments between s and s_c. 110
5.4 BLEU score obtained by two different systems with different data sizes for English-to-Turkish translation. 130
5.5 Effect of FMS in the combined $\text{EBMT}_{\text{TM}} + \text{SMT}$ system. 131

6.1 Length-based selection of potential set of candidate examples to find the closest match. 140
6.2 Detailed workflow of the IR-engine integrated EBMT_{TM} system. . . . 142
6.3 Number of times $\text{EBMT}_{\text{TM}} + \text{SMT} + \textit{index}$ used in the hybrid system and the number of times the same closest-matching sentences are selected by the systems. a=FMS>0.85, b=FMS>0.85 & EqUS and c=FMS>0.80 OR (FMS>0.70 & EqUS). 152

List of Tables

2.1 Corpus Statistics. TTR: type-token ratio, Bn: Bangla, En: English, Fr: French, Tr: Turkish, Zh: Chinese. 41

3.1 Average number of analogical equations attempted or solved with different heuristics in source and target sides. 52

3.2 Solution associated with moves in pseudo-distance matrices 56

3.3 Example of transliteration. The numbers in bracket are the frequencies of each transliteration candidate as output. 61

3.4 Transliteration accuracies (in %) for English-to-Hindi with different models using different heuristics. RT: Average Running Time. 61

3.5 Translation scores obtained for English-to-Chinese MT with **AEBMT** system 62

3.6 Translation scores obtained for English-to-Chinese MT with **AEBMT+SMT** system 63

3.7 Translation scores obtained for Chinese-to-English MT 63

3.8 Transliteration accuracies (in %) for English-to-Hindi transliteration using different heuristics under different output-bounded combinations. RT: Average Running Time. 67

3.9 Example of transliteration with a tie in the highest frequency output. 73

4.1 Number of translation rules inferred using different data sets. 93

4.2 System accuracies obtained by different GEBMT models for English-to-Bangla MT. The subscript $score > x$ denotes the value of the translation score (q). 94
4.3 System accuracies using different GEBMT models for English-to-Turkish MT. The subscript $score > x$ denotes the value of the translation score (q). 95
4.4 System accuracy obtained with different translation score parameters in the English-to-Turkish GEBMT system. 96
4.5 System accuracies obtained by different translation scores (q) in English-to-Turkish GEBMT system. 98
4.6 Example translation using GEBMT and SMT systems. 98
4.7 System accuracies obtained using different GEBMT and GEBMT-PT models for English-to-Turkish MT. The subscript $score > x$ denotes the value of the translation score (q). 99

5.1 Source–target translation equivalents in TM 107
5.2 Baseline BLEU scores (%) of the two systems and the scores for EBMT_{TM} system. 116
5.3 Baseline NIST scores of the two systems and the scores for EBMT_{TM} system. 117
5.4 English-to-Turkish MT system results for the EBMT_{TM} + SMT system with different combining factors. The second column indicates the number (and percentage) of sentences selected from the EBMT_{TM} system during combination. 120
5.5 English-to-French MT system results for the combined EBMT_{TM} + SMT system with different combining factors. 121
5.6 Human MT evaluation scales 122
5.7 Average fluency and adequacy of the English-to-Bangla MT system on a scale of 1-5 (as in Table 5.6). 122

5.8 Manual inspection of reasons for improvement in English-to-Bangla translation. . . . 123
5.9 Average fluency and adequacy of the English-to-Turkish MT systems on a scale of 1-5 (cf. Table 5.6). n=number of sentences evaluated under a particular feature value. . . . 123
5.10 Reasons for improvement in English-to-Turkish translation. . . . 125
5.11 Average fluency and adequacy of the English-to-French MT systems on a scale of 1-5 (cf. Table 5.6). n=number of sentences evaluated under a particular feature value. . . . 126
5.12 Reasons for improvement in English-to-French translation. . . . 126
5.13 Examples of improved translations by EBMT_{TM} system over the baseline SMT system for different reasons. . . . 127
5.14 Examples of improved translations of the baseline SMT system over the EBMT_{TM} system for different reasons. . . . 128
5.15 Average running time (in *seconds*) of the two different systems. . . . 135

6.1 Average running time (in *seconds*) of different systems with English–Turkish IWSLT09 and English–French EMEA data sets. . . . 147
6.2 BLEU scores for the three different systems for English-to-Turkish and English-to-French under different conditions. i denotes the number of bins considered during grouping. . . . 148
6.3 System accuracies of the EBMT_{TM} + SMT + *index* system with different combining factors using English–French JRC-acquis data. . . . 149
6.4 The effect of indexing in selection of s_c and in final translation. . . . 153

Abstract

Statistical Machine Translation (SMT) typically has difficulties with less-resourced languages even with homogeneous data. In this thesis we address the application of Example-Based Machine Translation (EBMT) methods to overcome some of these difficulties. We adopt three alternative approaches to tackle these problems focusing on two poorly-resourced translation tasks (English–Bangla and English–Turkish). First, we adopt a runtime approach to EBMT using proportional analogy. In addition to the translation task, we have tested the EBMT system using proportional analogy for named entity transliteration. In the second attempt, we use a compiled approach to EBMT. Finally, we present a novel way of integrating Translation Memory (TM) into an EBMT system. We discuss the development of these three different EBMT systems and the experiments we have performed. In addition, we present an approach to augment the output quality by strategically combining EBMT systems and SMT systems. The hybrid system shows significant improvement for different language pairs.

Runtime EBMT systems in general have significant time complexity issues especially for large example-base. We explore two methods to address this issue in our system by making the system scalable at runtime for a large example-base (English–French). First, we use a heuristic-based approach. Secondly we use an IR-based indexing technique to speed up the time-consuming matching procedure of the EBMT system. The index-based matching procedure substantially improves run-time speed without affecting translation quality.

Acknowledgments

I would first like to express my profound sense of gratitude to my supervisors Prof. Andy Way and Dr. Sara Morrissey, for introducing me to this research topic and providing their valuable guidance and unfailing encouragement throughout the course of the work. Their sharp insight and enormous support not only helped to shape the work reported in this thesis, but also has constructed outstanding examples for my future career. I am also thankful to Prof. Harold Somers for his support and guidance during the first one and half years of my PhD. I am grateful to Prof. Josef van Genabith and Dr. Dónal Fitzpatrick, who provided insightful suggestions on my transfer report, much of which is incorporated into this thesis. I am also thankful to Prof. Mikel Forcada for his insightful ideas, constant support and inspiration throughout the course of this thesis.

I am also indebted to the post-doctoral researchers I work with. First of all, some of the work reported in this thesis would be impossible without the collaboration of Sudip Naskar, who has always been a keen researcher and a warm-handed friend. I am fortunate to have Declan Groves who helped me to give the thesis a concrete shape with his valuable suggestions. I owe another big thank to Özlem Çetinoğlu for helping me to understand the Turkish translation output.

Thanks to Yanjun, Yifan, Ankit, Pratyush, Robert, Sergio, Rejwanul, Hala, John, Ventsi, Jinhua and Rasoul as past and present members of my research group for their support and interest in my work. Thanks also to my wider research group, Debasis, Maria, Javed and Yalemisew for providing teatime distraction and some of the oddest conversation! Special thanks to Joachim for maintaining our computing cluster in excellent shape and consistently providing unix tips. Thanks to Eithne, Ríona, and Fiona for their kind help since the first day I arrived in Ireland.

Thanks to Sunandan, Sibansu, Debasis, Tirthankar, Rasoul, Erwan, Killian and Lorraine who helped conducting the manual evaluation of the machine translation output.

I have had a consortium of supporters outside of DCU to whom I am most grateful for reminding me of the outside world. Thank you to my family, who have been supporting me through the ups and downs in the course of my education. I thank all my well-wishers who directly and indirectly contributed for the completion of this thesis.

Last but certainly not the least, I am thankful to Science Foundation of Ireland for the research grant (grant 07/CE/I1142, Centre for Next Generation Localisation) to support this research.

Chapter 1

Introduction

"... translation is a fine and exciting art, but there is much about it that is mechanical and routine."

Martin Kay (1997)

In the past two decades, Machine Translation (MT) has shown very promising results particularly using Statistical Machine Translation (SMT) techniques. The success of an SMT system mostly depends on the amount of parallel corpora available for the particular language pair. Large amounts of parallel resources (OPUS (Tiedemann and Nygaard, 2004), Europarl (Koehn, 2005), etc.) are available for the dominant languages of the world (English and other European languages). Developing such language data involves a lot of time, money and other resources, but such investment serves to increase the prominence and power of these languages and ignores the less dominant, minority languages (Ó'Baoill and Matthews, 2000). There exist a large number of languages which suffer from the scarcity of reasonably good amounts of parallel corpora, e.g., Indic languages, sign languages etc. Some of these languages (Hindi, Bangla/Bengali, etc.) are leading languages of the world in terms of number of speakers but are very poorly resourced (very little machine-readable parallel text exists).

Many SMT frameworks have shown low translation scores for these poorly re-

sourced languages (Islam et al., 2010; Khalilov et al., 2010). It is often the case that domain-specific translation is required to tackle the issue of scarce resources (Nießen and Ney, 2004). However, Example-Based Machine Translation (EBMT) systems perform better with homogeneous domain-specific data (Armstrong et al., 2006) especially when the amount of available resources is limited (Denoual, 2005). Although both SMT and EBMT systems are corpus-based approaches to MT, each of them has their own advantages and limitations. Typically, an SMT system works well when significant amounts of training data (i.e. parallel bilingual corpora) are available for the language pair. An SMT system has the advantage of incorporating a statistical language model (typically derived from a large monolingual corpus) directly into the system which improves the fluency of the translation, something which is absent from the majority of traditional EBMT systems. However, an exception can be found in the Pangloss EBMT system (Brown and Frederking, 1995) that uses a statistical language model in the target language but, unlike todays SMT-based systems, the Pangloss system has no bilingual statistical model to estimate the closeness of the translation given the source text. In addition, SMT systems use many features (e.g. phrase translation probabilities, word reordering probabilities, lexical weighting, etc.) which are extracted from data during training, within a statistical framework. In contrast, EBMT approaches, in general, lack a well-formed or well-defined probability model and restrict the use of statistical information during the translation process. However, EBMT approaches can be developed with fewer examples (Somers, 2003) compared to the amounts of training data needed in general by an SMT system;[1] furthermore, an EBMT system works well when training and test sets are quite close in nature (Marcu, 2001a) (sharing of surface words/phrases and similarity in grammatical structure of the sentences). This is because EBMT systems search the source side of the example-base for close matches to the input sentences and obtain corresponding target segments at runtime. These

[1] A notable exception was reported in (Popović and Ney, 2006). They showed that SMT can achieve acceptable translation accuracies using a small amount of parallel data (including dictionary, phrase book).

target segments are reused during recombination.

EBMT is often linked with the related concept of "Translation Memory" (TM). TM and EBMT have in common the idea of reusing examples from already existing translations. The main difference between EBMT and TM is that TM is an interactive tool for human translators, while EBMT is a fully automatic translation technique. EBMT generally uses a sentence-aligned parallel text as the primary source of data. TMs additionally make use of terminology databases and precomputed subsentential translation units. TM reduces the data sparsity problem using these additional resources. Further details on TM technology are outlined in Section 2.3.

Keeping these points in mind, it is important to be able to develop a reasonably good quality MT system based on limited amounts of data. It is often the case that EBMT systems produce a good translation while SMT systems fail and vice versa (Dandapat et al., 2010b). In order to effectively use both approaches, we employ a combination of both EBMT and SMT to improve translation accuracy. Although MT is our primary goal, we conduct an experiment on named entity (NE) transliteration[2] using one of our EBMT systems, the motivation being to showcase the power of EBMT for a task similar to MT.

1.1 Research Questions

The state-of-the-art phrase-based SMT approach has proven to be the most successful MT approach in MT competitions e.g. NIST,[3] WMT,[4] IWSLT[5] etc. However, the problem of low translation accuracy has been encountered for many language pairs especially those with fewer resources (Islam et al., 2010; Khalilov et al., 2010).

[2]NEs are essentially names of persons, locations and organizations. NE transliteration (Knight and Graehl, 1998) is defined as phonetic translation of names across languages which play a significant role in many NLP and Information Retrieval systems.

[3]National Institute of Standards and Technology: `http://www.itl.nist.gov/iad/mig/tests/mt/`

[4]Workshop on Statistical Machine Translation. `http://www.statmt.org/wmt11/`

[5]International Workshop on Spoken Language Translation. `http://www.iwslt2011.org/`

SMT systems discard the actual training data once the translation model and language model have been estimated. This further leads to their inability to guarantee good quality translations for sentences which closely match those in the training corpora. EBMT systems are capable of learning translation templates which are anticipated to be useful in overcoming some of the difficulties encountered by SMT systems, such as long-distance dependencies. EBMT systems are particularly good at capturing long-distance dependencies and at maintaining the linked relationships between source and target texts, through the use of these templates. We therefore raise the first research question of this thesis:

> **(RQ1)** *Can we exploit EBMT approaches to build better quality MT systems compared to purely SMT-based systems when working with limited resources?*

TM is widely used in computer-aided translation (CAT) systems to assist professional translators. CAT systems segment the input text to be translated and compare each segment against the TM database. A CAT system produces one or more target equivalents for the source segment and professional translators select and recombine them (perhaps with modification) to produce the desired translation themselves. It is likely to find a good TM match for an input sentence (i.e. one that is anticipated to require fewer edits by a human translator on the target side) if the test sentences are homogeneous with the stored example-base. After obtaining a good TM match, it may be possible to perform some of the edits (often manually done by professional translators) automatically using a subsentential TM database. This leads to our second research question:

> **(RQ2)** *Can we use a TM technology within an EBMT system for translating homogeneous data?*

In **RQ2**, we mainly consider integrating a TM into an EBMT system, similar to how TMs are typically used in a CAT system. However, this approach may work well with those input sentences that have a significantly similar translation example stored in the database of examples. When the TM selection is not adequate, we

can use the SMT paradigm to produce robust translation. This lead us to our third research question:

> **(RQ3)** *How effectively can we combine EBMT systems with state-of-the-art phrase-based SMT systems to handle the particular data sparsity in SMT?*

Finally, we need to keep in mind that search techniques often affect the performance of a TM-based system to retrieve the best fuzzy match in real time when using a large database of examples. This remains an area under active optimization, which leads us to the final research question of the thesis:

> **(RQ4)** *If the EBMT/TM-based approach successfully works with limited homogeneous data, can we effectively scale up the basic system to larger amounts of training data?*

In order to address RQ4, we plan to index the whole example-base using inverted indexing (Manning et al., 2008a) and intend to retrieve a potential set of candidate sentences (likely to contain the closest match) from the indexed example-base.

1.2 Roadmap

The remaining chapters of this thesis seek to address the research questions proposed in Section 1.1. We will also provide necessary background information and overviews of past approaches to make the thesis self-contained. The remainder of the thesis is broadly organized as follows:

Chapter 2 provides a general outline of the two main data-driven approaches to MT: EBMT and SMT. We describe the main processes carried out when performing EBMT and outline two approaches (a runtime EBMT system using proportional analogy and a generalized translation template-based EBMT model) used in our work. We include the description of the SMT framework which is used as a baseline for most of the experiments conducted in this thesis. In addition we also describe the

TM paradigm which is used in our work to develop a novel EBMT system. Finally, we describe the evaluation metrics and data used for the experiments in the thesis.

Chapter 3 describes our work on runtime EBMT using proportional analogy. We outline our particular approach and use of different heuristics within the analogy-based framework. Furthermore, we describe a combination of analogy-based EBMT and SMT to mitigate some of the problems of SMT using EBMT. We report a wide range of experiments with translation and transliteration tasks to show the effectiveness of the analogy-based EBMT approach.

Chapter 4 presents our work on a compiled approach to EBMT (Cicekli and Güvenir, 2001). We introduce a probabilistic score to produce ranked output in the translation process. Finally, based on this probability score, we combine this approach with SMT in order to improve the performance of the combined system.

Chapter 5 introduces a novel runtime EBMT system using TM for the translation of homogeneous domain-specific data. We also present an approach to improve output quality by strategically combining both EBMT and SMT approaches to handle issues arising from the sole use of SMT.

Chapter 6 presents two different methods to make the EBMT system scalable at runtime. First, we describe a heuristic-based approach. Subsequently we propose an information retrieval-based indexing technique to speed-up the time-consuming matching procedure of the EBMT system.

Chapter 7 concludes the thesis and outlines some future avenues of research.

1.3 Publications

The research presented in this dissertation was published in several peer-reviewed conference proceedings. Joint work in (Somers et al., 2009), reports reviews on

research using proportional analogy-based MT. Our experiments and findings for adopting a proportional analogy-based approach for translating homogeneous data is presented in (Dandapat et al., 2010b). (Dandapat et al., 2010c) presents our initial attempts towards using subsentential translation memory in an EBMT framework. Furthermore, our principal work involving two different EBMT systems and our approach to improve output quality by strategically combining EBMT approaches with SMT are presented in (Dandapat et al., 2011). Finally, (Dandapat et al., 2012) presents a novel strategy of integrating information retrieval technique in an EBMT framework in order to make an EBMT system scalable at runtime.

There are also a few additional papers which are related to the work in this thesis. The OpenMaTrEx system (a free/open-source MT system which we have used as a baseline SMT system throughout the course of this thesis) itself is presented in (Dandapat et al., 2010a). The development of 6-way parallel corpus (including English–Bangla) is reported in joint work in (Morrissey et al., 2010).

Publications from the thesis

- Dandapat, S., Morrissey, S., Way, A., and van Genabith, J. (2012). Combining EBMT, SMT, TM and IR Technologies for Quality and Scale. In *EACL 2012 Joint Workshop on Exploiting Synergies between Information Retrieval and Machine Translation (ESIRMT) and Hybrid Approaches to Machine Translation (HyTra)*, Avignon, France. pp. 48–58.

- Dandapat, S., Morrissey, S., Way, A., and Forcada, M. L. (2011). Using Example-Based MT to Support Statistical MT when Translating Homogeneous Data in a Resource-Poor Setting. In *Proceedings of the 15th Annual Meeting of the European Association of Machine Translation (EAMT 2011)*, Leuven, Belgium. pp. 201–208.

- Dandapat, S., Morrissey, S., Naskar, S. K., and Somers, H. (2010c). Statistically Motivated Example-based Machine Translation using Translation Mem-

ory. In *Proceedings of the 8th International Conference on Natural Language Processing (ICON 2010)*, Kharagpur, India. pp. 168–177.

- Dandapat, S., Morrissey, S., Naskar, S. K., and Somers, H. (2010b). Mitigating Problems in Analogy-based EBMT with SMT and vice versa: a Case Study with Named Entity Transliteration. In *Proceedings of the 24th Pacific Asia Conference on Language Information and Computation (PACLIC 2010)*, Sendai, Japan. pp. 146–153.

- Dandapat, S., Forcada, M. L., Groves, D., Penkale, S., Tinsley, J., and Way, A. (2010a). OpenMaTrEx: A Free/Open-Source Marker-Driven Example-Based Machine Translation System. In *Proceedings of the 7th International Conference on Natural Language Processing (IceTAL 2010)*, Reykjavik, Iceland. pp. 121–126.

- Morrissey, S., Somers, H., Smith, R., Gilchrist, S., and Dandapat, S. (2010). Building a Sign Language corpus for use in Machine Translation. In *Proceedings of the 4th Workshop on Representation and Processing of Sign Languages: Corpora for Sign Language Technologies,* 2010. Valetta, Malta. pp. 172–177.

- Somers, H., Dandapat, S., and Naskar, S. K. (2009). A review of EBMT using proportional analogy. In *Proceedings of the 3rd Workshop on Example-Based Machine Translation (EBMT 2009),* 2009. Dublin, Ireland. pp. 53–60.

Chapter 2

Background

Different machine translation (MT) techniques have emerged over time since Warren Weaver's (1949) first attempt at MT using mechanical approaches. The different approaches of MT can be primarily classified as either rule-based or data-driven. Although they represent different approaches to MT, today they borrow ideas heavily from each other. In this chapter, we discuss their differences as well as their similarities.

The rule-based paradigm of MT (RBMT) dominated the field until the end of the 1980s. During that time, MT showed success with many operational and commercial systems such as Systran (Elliston, 1979).[1] RBMT makes use of linguistic rules which are used to handle problems of morphology, syntactic analysis, lexical transfer, syntactic generation. As a result of this early success, subsequent MT research focused on the use of linguistic rules to develop advanced transfer-based (Vauquois and Christian, 1985) and interlingua-based systems (Muraki, 1987). However, the shortcomings of these approaches, such as the cost of developing rules for transfer-based systems and the problem of defining true interlingua, motivated researchers to look at empirical approaches. During this time, in the late 1980s, the dominance of RBMT lessened with the emergence of corpus-based approaches. Researchers borrowed ideas from the speech processing community to develop a new technique

[1] http://www.systran.co.uk/

for MT (Brown et al., 1988) and introduced the new statistical MT (SMT) (Brown et al., 1990) paradigm. At the same time, the use of examples for MT emerged from the work by Nagao (1984) and the approach came to be known as example-based MT (EBMT). EBMT and SMT represent the two threads of what is now known as data-driven MT.

Today, the field of research in MT is largely dominated by data-driven, or corpus-based approaches, with SMT, by far, being the most prevalent of the two. Corpus-based approaches derive knowledge from parallel corpora to translate new input. The existence of large machine-readable parallel corpora for many languages and powerful machines led to the development of good quality, robust translation systems.

The attractiveness of such data-driven approaches, in particular SMT, was due to their ability to perform translation without the need of explicit linguistic information. This meant systems could be developed relatively quickly and inexpensively compared to the previous costly rule-based approach. However, there remains some ongoing work in the area of RBMT and EBMT. Some recent examples of successful RBMT systems include Apertium (Forcada et al., 2011) and OpenLogos (Barreiro et al., 2011). Likewise, some successful EBMT systems include CMU-EBMT (Brown, 2011) and Cunei (Phillips, 2011).[2]

In this chapter we first outline the main data-driven approaches to MT which we are using in this thesis. In the next section, we describe the general EBMT approach and provide a brief review of the different EBMT techniques we adopt in our own work. We then discuss the SMT framework, with particular reference to the recent phrase-based SMT models (Koehn et al., 2003, 2007). We also discuss the concept of translation memory (TM) and its uses in translation. Finally, we devote a section to describing the tools, data and evaluation metrics used in our own work.

[2]Cunei is a hybrid MT platform that utilizes the concepts of EBMT and SMT.

2.1 Example-Based Machine Translation

The example-based approach to MT was first introduced by Nagao (1984) as "MT by analogy principle", stating:

> *"Man does not translate a simple sentence by doing deep linguistic analysis, rather, man does translation, first, by properly decomposing an input sentence into certain fragmental phrases, ... then by translating these phrases into other language phrases, and finally by properly composing these fragmental translations into one long sentence. The translation of each fragmental phrase will be done by the analogy translation principle with proper examples as its reference."* (Nagao, 1984, p.178)

According to the author, EBMT relies on the intuition that humans make use of translation examples they have previously encountered in order to translate new input sentences. The prerequisite for an EBMT system is a set of bilingual sentence-aligned parallel examples (also known as a 'bitext' or 'example-base') for the induction of translations of subsentential fragments. An EBMT system relies on past translations to derive the target output for a given input and performs the translation in three steps: *matching*, *alignment* and *recombination* (Somers, 2003):

- Matching: finds the example or set of examples from the bitext which most closely match the source-language string to be translated.
- Alignment: extracts the source–target translation equivalents from the retrieved examples of the matching step.
- Recombination: produces the final translation by combining the target translations of the relevant subsentential fragments.

An illustration of the working principle of an EBMT system is given in Figure 2.1. When translating the input sentence S, the system first searches the source side of the example-base and selects the closely matched sentences s_1 and s_2 from

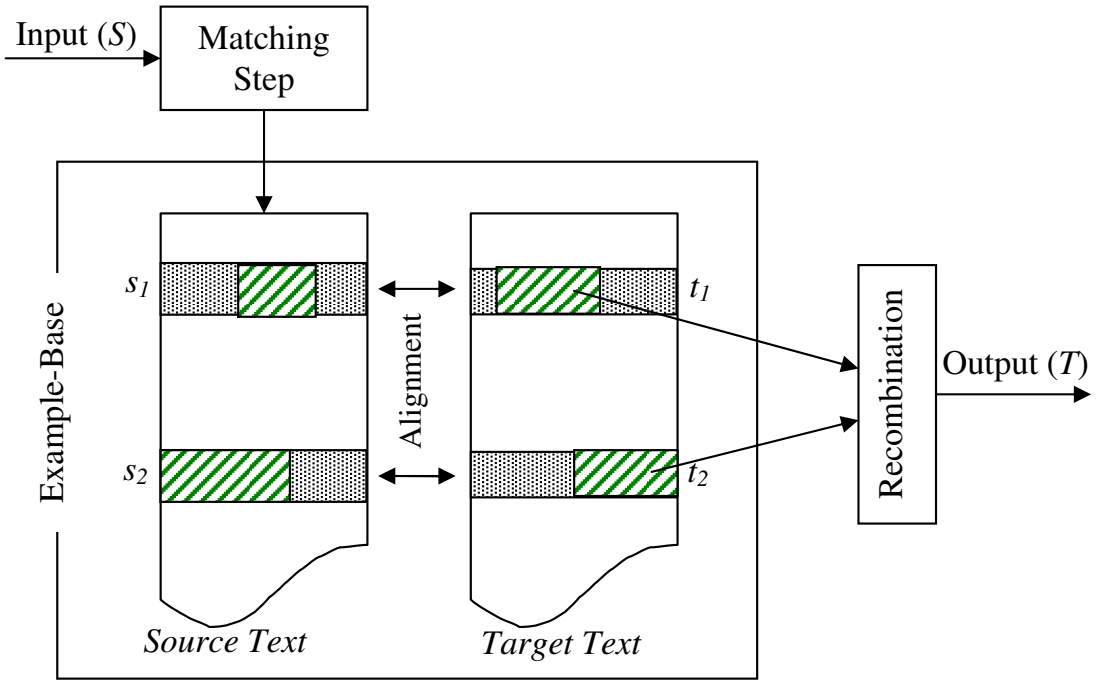


Figure 2.1: The translation process of an EBMT system.

the corpus for the input S. Then, the subsentential fragments (marked in green) that can be used to construct the input S are extracted from s_1 and s_2. During the alignment process, the target equivalents for the relevant fragments of s_1 and s_2 are extracted from their relevant target translation correspondences t_1 and t_2. Finally, the retrieved target-language fragments are fed into the recombination process to construct the final translation T.

The translation process of an EBMT system can be further illustrated by constructing the Turkish translation for the input English sentence *Where can I find tourist information.* For this example, we first assume a parallel corpus consisting of the two simple English sentences and their Turkish translations in (1).

(1) a. Where can I find ladies dresses⇔ payan kıyafetlerini nereden bulabilirim

LADY DRESSES WHERE-FROM FIND-CAN-I[3]

b. just in front of the tourist information⇔ turist bilgilerini hemen önünde

TOURIST INFORMATION JUST IN-FRONT-OF

[3]The English gloss of a foreign language sentence is represented in all upper case characters with the words mapped one-to-one to the foreign language sentence.

The useful bilingual fragments in (2) are extracted from the sentences in (1) applying the bilingual fragment extraction algorithm (Nirenburg et al., 1993; Somers et al., 1994). These fragments can be extracted using a very simple subsequence measure, such as Levenshtein distance (Levenshtein, 1965) during the alignment process.

(2) Where can I find ⇔ nereden bulabilirim

tourist information ⇔ turist bilgilerini

Following the extraction process, the relevant fragments in (2) are combined to produce a translation for the original input sentence, as shown in (3).

(3) Where can I find tourist information ⇔ turist bilgilerini nereden bulabilirim

Note that the sentence pair in (3) did not appear in the original corpus in (1). The new sentence pair in (3) can subsequently be added to the example-base so that if this same source sentence is encountered, its translation can be retrieved using exact sentence matching, by-passing the alignment and recombination steps. This is something that traditional SMT systems can not do – if they encounter a previously seen translation they process it in the same way as if they had not seen it before. More clearly, taking the example in (3), an SMT system will still investigate all possible segmentations for the input sentence, despite having previously translated the input. Although the SMT system may still produce the same (and possibly correct) translation, the SMT decoder clearly does not take any advantage of possible efficiencies, unlike EBMT systems.

It is important to note that EBMT systems differ widely in their matching stages. The difference in matching largely depends on how the translation examples are stored in the example-base. All matching procedures in EBMT systems involve a distance or similarity measure and can be based on a number of different algorithms as described below. We discuss some of the EBMT matching techniques in detail below.

Character-Based Matching:

A character-based distance measure can be employed when examples are stored as simple strings (Somers et al., 1994). The character-based string distance can be easily calculated using a well-established dynamic programming technique, such as the Levenshtein distance algorithm. The problem of determining the distance between strings of characters is equivalent to the edit-distance problem (Wagner and Fischer, 1974) and can be easily implemented. However, the approach has the disadvantage that it often ignores sentences that are closer in meaning, but have a larger edit-distance score, compared with a less meaningful, but "closer" sentence.

This can be illustrated with the example in (4). When attempting to find a match for the sentence in (4a) using character-based distance, the system will choose (4b) due to a smaller distance between *agree* and *disagree* when compared to the distance between *agree* and *concurs* in (4c). The system does not have any clue that *agree* and *concurs* are synonyms to guide the choice of (4c) as the preferable match for the input (4a).

(4) a. The President *agrees* with the decision.

b. The President *disagrees* with the decision.

c. The President *concurs* with the decision.

Word-Based Matching:

In order to avoid the problems of character-based distance metrics, many EBMT systems use the classical word-based similarity such as suggested by Nagao (1984). The word-based similarity measure uses dictionaries and thesauri to determine the relative word distance in terms of meaning (Sumita et al., 1990). Such a technique will be able to correctly identify (4c) as the preferred match for (4a) due to the relatively closer semantic distance between *agree* and *concurs* when compared to the semantic distance between *agree* and *disagree*. The usefulness of word-based matching is shown in the example of Nagao (1984), illustrated here in (5) when

an input sentence has two or more "competing examples" in the example-base. Considering the competing examples in (5) from (Somers, 2003, p.20), the system will correctly produce the Japanese translation for the English verb *eats* as *taberu* (eats food) in (6) using word-based similarity. This is captured using the semantic similarity between *A man* and *He*, and between *vegetables* and *potatoes*.

(5) a. A man eats vegetable ⇔ Hito wa yasai o taberu

b. Acid eats metal ⇔ San wa kinzoku o okasu

(6) He eats potatoes ⇔ Kare wa jagaimo o taberu

Although it is a useful method for EBMT matching, carrying out the necessary semantic analysis is not without its difficulties and requires language-specific analysis.

Pattern-Based Matching:

In many EBMT systems, similar examples are used to produce translation templates. In this process general translation patterns are created by replacing subsentential chunks with variables. These generalized translation patterns can be viewed as a type of transfer rule as used in an RBMT system. The use of generalized patterns increases the flexibility of the matching process.

To find generalized patterns, Brown (1999) uses the concept of equivalence classes, such as *person*, *date* and *city* along with some linguistic information, such as gender and number. Certain words which are members of a particular equivalence class are generalized with the corresponding class names to create template patterns. New input sentences are matched against these generalized template patterns. For example, the sentence in (7a) can be generalized recursively into (7b) and (7c) by replacing words with their membership equivalence classes.

(7) a. John Miller flew to Frankfurt on December 3rd.

b. ⟨FIRSTNAME-M⟩ ⟨LASTNAME⟩ flew to ⟨CITY⟩ on ⟨MONTH⟩ ⟨ORDINAL⟩.

c. ⟨PERSON-M⟩ flew to ⟨CITY⟩ on ⟨DATE⟩.

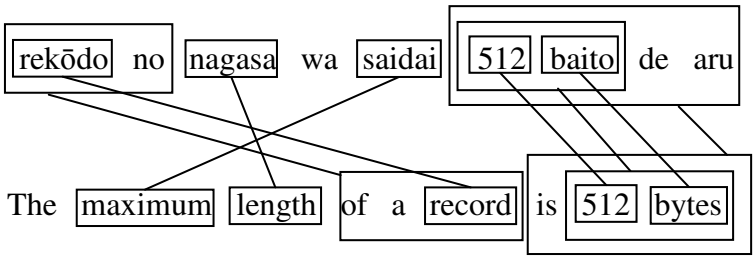


Figure 2.2: Aligned example from Kaji et al. (1992), with coupled Japanese–English word and phrase pairs identified by corresponding links.

The template in (7c) can match any sentence that follows a similar pattern. For example, the sentence *Michael Milan flew to New Delhi on 15th January* matches with (7c) by replacing the instances with the equivalence classes. These equivalence classes were initially constructed manually by linguists to reduce the amount of data required for translation. Later, Brown (2000) used clustering techniques for automatic creation of equivalence classes from bilingual training corpora. The clustering technique used context on the source-side only. Brown (2000) used bilingual corpora, since the equivalence-class members include the corresponding translation.

Kaji et al. (1992) used syntactic categories to identify generalized patterns. They used source- and target-language parsers to construct parse trees (source and target language) for each translation pair in the example-base. Then, a bilingual dictionary was used to align syntactic units of the parsed structure to generate translation templates. Taking the aligned structure in Figure 2.2 (based on (Kaji et al., 1992, p.673)), the generalized examples in (8a) and (8b) can be extracted by replacing coupled pairs by variables incorporating information about their syntactic categories.

(8) a. X_1[NP] no nagasa wa saidai 512 baito de aru ⇔ The maximum length of X_1[NP] is 512 bytes

b. X_1[NP] no nagasa wa saidai X_2[N] baito de aru ⇔ The maximum length of X_1[NP] is X_2[N] bytes

New input sentences are matched against the source side of the translation template

to extract the corresponding target language pattern. Then a conventional MT system was used to translate words/phrases corresponding to the variables in the translation templates.

Gough and Way (2004) used the marker hypothesis (Green, 1979) to produce generalized templates. The syntax of a language is marked at the surface level by a set of *marker words* (closed category words or morphemes). Marker words are used to chunk the text:

(9) [<DET>**that** is almost] [<DET>**a** personal record] [<PREP>**for** <PRON> **me** <DET> **this** autumn] ⇔ [<DET>**c'** est pratiquement] [<DET>**un** record personnel] [<PREP>**pour** <PRON> **moi** <DET> **cet** automne]

Taking this marker-tagged sentence pair, marker chunks in (10) are automatically generated.

(10) a. <DET>**that** is almost ⇔ <DET>**c'** est pratiquement

b. <DET>**a** personal record ⇔ <DET>**un** record personnel

c. <PREP>**for** me this autumn ⇔ <PREP>**pour** moi cet automne

Taking the marker chunks in (10), a set of generalized templates can be inferred in (11) by replacing the marker word with its relevant tag.[4]

(11) a. <DET> is almost ⇔ <DET> est pratiquement

b. <DET> personal record ⇔ <DET> record personnel

c. <PREP> me this autumn ⇔ <PREP> moi cet automne

These generalized templates bring about more flexibility in the matching process. For example, the previously unseen substring *by me this autumn* can now be translated using the template (11c), by inserting the translation for *by* into the target side of the template. This process also generates a word-level lexicon using

[4]Note that according to (Gough and Way, 2004), each marker chunk must contain at least one content word, therefore the chunks *<PREP> for*, *<PRON> me* and *<DET> this autumn* are joined to form a single marker chunk in (10c). A similar process is applied to the target side.

the deleted marker words as in (12a). Additionally, marker chunks with a singleton content word on both source and target side can be added to the lexicon, as in (12b), as the content words can be assured to be translations of each other.

(12) a. <PREP> for ⇔ <PREP> pour

b. automn ⇔ automne

In addition to the above flavours of EBMT matching and alignment, other studies have also been proposed in the literature, such as Carroll's angle of similarity (Carroll, 1999), annotated word-based matching by Cranias et al. (1994) and tree structure-based matching (Maruyama and Watanabe, 1992; Hearne, 2005).

2.1.1 Approaches to EBMT

EBMT was first introduced as an analogy-based approach to MT. Apart from the term "analogy-based", EBMT has gone by various names, including "case-based", "memory-based" and "experience-guided" MT (Somers, 2003). Unlike SMT, EBMT lacks a well-defined unified modeling framework. The consequence of this is that a great variety of approaches exist under its name. However, the two main approaches to EBMT are distinguished by the inclusion or exclusion of a preprocessing/training stage (Carl and Way, 2003; Hutchins, 2005). Approaches that do not include a training stage are often referred to as "pure" EBMT approaches or "runtime" approaches (e.g. Lepage and Denoual, 2005b). These approaches have the advantage that they do not depend on any time-consuming preprocessing. On the other hand, their runtime complexity can be considerable. Approaches that incorporate a training stage are commonly called "compiled approaches" (e.g. Al-Adhaileh and Tang, 1999; Cicekli and Güvenir, 2001), as training usually consists of compiling units below the sentence level before runtime.

We describe below details of two different flavours of EBMT from each of the aforementioned approaches that we use in this thesis.

Pure EBMT

In this section, we report the work on pure/runtime EBMT approach using proportional analogy (PA) which we revisit in this thesis to mitigate some of the main translation problems in a statistical framework. Here we provide a brief review of the different work using PA. The details of the approach can be found in Chapter 3 which describes our particular attempts to use a PA-based system to address the proposed research questions.

Lepage and Denoual (2005c) introduced an EBMT system adhering to the runtime EBMT approach. They developed the system based on the concept of PA. This approach – a type of analogical learning – was attractive because of its simplicity; and reported considerable success. A PA is noted $A : B :: C : D$ in its general form and reads "A is to B as C is to D". The authors make use of the idea that four sentences of a language can form the pattern "A is to B as C is to D". For example, the authors constructed a proportional analogy in English as in (13).

(13)

I'd like to open these windows.	:	*Could you open a window?*	::	*I'd like to cash these traveler's cheques.*	:	*Could you cash a traveler's cheque?*
|		|		|		|
Ces fenêtres là, je peux les ouvrir.	:	*Est-ce que vous pouvez m'ouvrir une fenêtre?*	::	*Ces chèques de voyage, là, je peux les échanger.*	:	*Est-ce que vous pouvez m'échanger un chèque de voyage?*

Given the three entities (A, B, and C) of a PA, Lepage (1998) proposed an algorithm to solve an analogical equation to construct the fourth entity (D). Based on this idea, Lepage and Denoual (2005c) developed their "purest EBMT system". Given the translations for three out of four sentences in (13) that together form an analogical equation, the translation of the fourth can be obtained by solving the analogical equation in the target side.

The ALEPH system is an implementation of the research described in the three papers of Lepage and Denoual (2005a,b,c), and was tested on a corpus of 160K English, Japanese and Chinese sentences from the C-STAR project's Basic Travel Expression Corpus (BTEC). The system did very well on data from the IWSLT 2004 competition, coming a close second to the competition winner on all measures (Lepage and Denoual, 2005b, p.273). The ALEPH system evolved into a new system, named GREYC, with some modifications as described in Lepage and Lardilleux (2007), and (Lardilleux, 2011). The GREYC system also incorporated new heuristics and had an additional refinement of non-determinism to generate all possible solutions for a single analogical equation which otherwise had one solution in ALEPH, and, accordingly, is much slower. While Lepage and colleagues have had modest success using PA for a full translation task, the idea was adapted to translating unknown words in the context of another approach to MT as reported by Denoual (2007), Langlais and Patry (2007), and Langlais et al. (2009). Denoual's (2007) experiments attempt to translate all unknown words in a Japanese-to-English task and have reported that translation adequacy improves (in terms of NIST score (Doddington, 2002)), but fluency (as measured by BLEU score (Papineni et al., 2002)) remains unchanged or even decreases.[5] Langlais and Patry (2007) had more success in handling unknown words when the language pairs are quite close in morphological structure. Langlais and Yvon (2008) use PA to supplement the words and phrases for standard SMT when a word to be translated is not covered by the statistical model. Experiments involved translating individual words and phrases of up to five words for French-to-English translation. Their methods produce many candidate translations: hundreds, sometimes thousands for phrase translation. The average position of the first acceptable translation was 1,602nd out of 875,000 average candidate translations. Clearly some further filtering mechanism on the output is needed. They showed promising results supplementing the phrase table in an SMT

[5]BLEU and NIST are two of the most widely used metrics for automatic evaluation of MT systems. They are described in more detail in section 3.1.2.

system but failed to produce a good translation in the first position of the ranked list. Their approach is clearly unsuitable for proposing a single translation. Finally, Langlais et al. (2009) applied the method to the translation of medical terms between English and French, Spanish, Swedish and Russian, in both directions. Their results generally showed an improvement on purely statistical approaches.

While the approach seems fraught with difficulties as a standalone translation model, its use for the special case of unknown words, particularly names or specialist terms, seems much more promising (Langlais et al., 2009). This motivates the use of a PA-based system (a runtime EBMT approach) to mitigate the problems that SMT has with unknown words. Thus, a PA-based system is anticipated to address our research question RQ3 (which concentrates on effective combination of EBMT and SMT to handle data-sparsity problem). The detail of our work using a PA-based approach is presented in Chapter 3 of this thesis.

Compiled EBMT

The compiled approach to EBMT learns translation templates from parallel sentences. A translation template is a generalized translation example pair, where some components (e.g. words, stem and morpheme) are generalized by replacing them with variables. Consider the following two source and target English–Turkish example pairs in (14) from (Cicekli and Güvenir, 2001, p.58):

(14) a. *I will drink orange juice* : *portakal suyu içeceğim*
b. *I will drink coffee* : *kahve içeceğim*

Clearly, the English sides of these two examples share the word sequence *I will drink* and differ in the word sequence *orange juice* and *coffee*. Similarly on the target-side, the similar part is *içeceğim* and differing parts are *portakal suyu* and *kahve*. Based on this observation, the subsentential alignments in (15) can be captured:

(15) a. *I will drink* : *içeceğim*
b. *coffee* : *kahve*
c. *orange juice* : *portakal suyu*

By substituting the similar or differing sequence with variables, the templates in (16) can be obtained:

(16) a. *I will drink* X^S : X^T *içeceğim*

b. X^S *coffee* : *kahve* X^T

c. X^S *orange juice* : *portakal suyu* X^T

Cicekli and Güvenir (2001) proposed an approach to generalize over sequences of words. The underlying assumption is that given two parallel sentence pairs, translation templates can be learned based on the similarities in both the source and target sides. The same applies to the differing parts between two parallel sentences. Generalization using this approach consists of replacing the similar or differing sequences with variables and producing a set of translation templates (including *atomic translation templates* containing no variables, as in (15)). These translation templates are further used to translate new input sentences. Prior to the above approach, other research was carried out to learn translation templates based on syntactic generalization, e.g. Kaji et al. (1992). Some recent work has also focused on morphological generalization to learn EBMT templates (Phillips et al., 2007). Their method exploits the regular nature of a morphologically rich language (Arabic) to generalize every word in the corpus regardless of different morphological inflections of the same root word. The approach showed significant improvement in BLEU scores when translating Arabic into English.

Translation templates essentially reduce the data-sparsity problem by generalizing some of the word sequences. Gough and Way (2004) demonstrated that a set of automatically derived generalized templates can improve both coverage and translation quality. Thus, the approach of Gough and Way (2004) is anticipated to answer research questions RQ1 and RQ3 (cf. Chapter 1). This motivates us to use this approach (in Chapter 4) to overcome the data-sparsity problem of phrase-based SMT.

2.2 Statistical Machine Translation

Statistical machine translation (SMT) (Brown et al., 1990) has dominated the research landscape of MT for most of the last decade. Originally based on the noisy-channel approach for speech recognition, the SMT model exploits Bayes' Theorem, given in Equation (2.1), to formulate the translation problem.

$$p(t|s) = \frac{p(s|t).p(t)}{p(s)} \tag{2.1}$$

In Equation (2.1), $p(t|s)$ represents the probability that a translation will produce t in the target language given a source-language input sentence s. The denominator $p(s)$ can be ignored as it is a constant (independent of t). Therefore, the equation to find the most probable t can be simplified by maximising the probability of t in $p(t|s)$, as shown by the equation in (2.2).

$$\arg\max_t p(t|s) = \arg\max_t p(s|t).p(t) \tag{2.2}$$

In this equation, the system maximises the product of the two remaining probabilities: $p(s|t)$, the probability of the candidate translation t being translated as s, and $p(t)$, the probability that the sentence t would be produced in the target-language. These two models are known as the *translation model* and *language model*, respectively. The translation model assigns probabilities to the set of target-language words/phrases that can be generated as the translation of a source-language string. This tries to ensure the *faithfulness* of translation. On the other hand, the language model organises these target-language words to obtain the most likely word sequence for the output translation. This tries to capture the *fluency* of the system output. Thus, the translation process can be viewed as a search problem that finds the translation t that maximizes the product in Equation (2.2). This search problem is known as *decoding*.

An SMT system therefore requires three main components: a translation model

to compute $p(s|t)$, a language model to calculate $p(t)$ and a decoder to search the most likely translation t by maximizing the product of the translation and language models as in Equation (2.2).

In word-based SMT (Brown et al., 1990), the translation elements were words. Given a sentence-aligned parallel corpus, algorithms were designed to learn word-to-word correspondences (which help generating the translation from a source-language sentence word by word) which induced a set of mappings between source and target sentences (Brown et al., 1988, 1990). This process is known as *word alignment*. However, the word-based translation models had problems of translating between languages with high 'fertility' (the number of target words generated by a source word). The development of phrase-based SMT systems (Och and Ney, 2002) resolved this issue. Incidentally, EBMT approaches have always used the concept of phrases since their very inception. Phrase-based SMT allows the mapping between a word sequence of n source language words (SMT phrases) with the sequence of m target-language words. However, these phrase pairs are still learned using an extension of the original word alignment technique. The decoding technique of the phrase-based SMT system is done in the same fashion as the word-based model by searching for the most likely target-language sequence given the source-language string by maximizing the product of the translation model and the language model.

The end-to-end translation process of a phrase-based SMT system can be categorized into the following pipelined stages:

- Given a parallel corpus, a set of **word alignments** are learned between the source- and target-language sentences of the parallel corpus (Brown et al., 1993; Och and Ney, 2003).
- After obtaining the word-aligned sentence pairs, equivalent phrase pairs are learned to build a **translation model** (Och and Ney, 2003).
- A **language model** is separately built from the target language text (Stolcke, 2002).[6]

[6]The language model of the target language is sometimes estimated only using the target-

- Finally, for a given input test sentence, the **decoder** finds the most likely target language translation using the translation and language model (Koehn et al., 2007).

The preferred model of SMT has now moved away from the classical noisy channel model into the log-linear model (Och and Ney, 2002) originally introduced by (Berger et al., 1996). The log-linear model throws away the structural dependencies of the generative noisy-channel model and computes $p(t|s)$ directly using feature functions. Equation (2.3) represents the log-linear SMT model.

$$p(t|s) = \arg\max_{t}\{\sum_{i=1}^{n} \lambda_i h_i(t,s)\} \tag{2.3}$$

where $h_i(t,s)$ denotes a feature function, λ_i is the corresponding weight factor, and n is the number of features.

The log-linear model enables the combination of several different models and integrates them into the system with the additional benefit of extending the number of features over the noisy-channel model. The noisy-channel model can be considered as a special case of the log-linear framework.[7] The noisy-channel SMT approach expressed in the log-linear framework with two feature functions (i.e. language model $p(t)$ and translation model $p(s|t)$) is given in Equation (2.4) considering $\lambda_1 = \lambda_2 = 0.5$. Using the model on held-out data automatically determines the relative importance of each feature.

$$p(t|s) = \arg\max_{t}\{\lambda_1 \log p(t) + \lambda_2 \log p(s|t)\} \tag{2.4}$$

We can see that each feature function in the log-linear approach is multiplied by a scaling factor λ_i. The different value of each of the λ_i ($\sum_{i=1}^{n} \lambda_i = 1$) determines the relative importance of each feature.

language side of the parallel training corpus, but in practice the language model is estimated from a much larger monolingual corpus.

[7] (Way and Hearne, 2011) provides a detail description of how the translation process by noisy-channel model can be expressed in a log-linear framework.

Moses (Koehn et al., 2003, 2007) is the most widely used open-source implementation of SMT.[8] Moses uses a log-linear model for translation. Phrase pairs are used as the translation unit in Moses. Each phrase pair consists of a source phrase s and an equivalent target phrase t. In its standard configuration, Moses uses a total of eight features in the log-linear model to perform translation. Five of these features are assigned to each phrase pair within the translation model:

- the inverse phrase translation probability $p(s|t)$ and the direct phrase translation probability $p(t|s)$, estimated from relative frequencies calculated over the aligned phrase pairs.

- $lex(s|t)$ and $lex(t|s)$ estimate the phrase translation probability based on phrase-internal word alignments.

- a phrase penalty constant (always $exp(1) = 2.718$) so as to favour the use of longer phrases.

The remaining three features are used during decoding to combine phrases:

- a language model score $p(t) = p(t_1, t_2, t_3...t_n) = p(t_n|t_1t_2t_3...t_{n-1})...p(t_2|t_1)p(t_1)$

- a distortion penalty $d(t, s)$ to limit reordering, $d(t, s) = \sum_i(start_i - end_{i-1} - 1)$ for each phrase i

- a word penalty $w(t)$ to balance the language model's bias towards short sentences, $w(t) = exp(length(t))$

A more detailed description of these features is given in Koehn et al. (2003) and (Koehn, 2010). Moses has been extended over time and additional features have been incorporated as part of various pieces of research (e.g. Koehn et al. (2005)). However, the eight features listed above have been found to perform consistently well in the log-linear model.

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

For our experiments in this thesis, we used OpenMaTrEx (Dandapat et al., 2010a),[9] an open-source MT system which provides a wrapper around the standard log-linear phrase-based SMT system Moses (Koehn et al., 2007) so that it can be used as a decoder for a merged translation table containing Moses phrase and marker-based chunk pairs. OpenMaTrEx uses GIZA++ (Och and Ney, 2003) for word alignment. The phrase and the reordering tables are built on the word alignments using Moses training scripts. We do not aim to give a detailed description of the SMT system as our work primarily focuses on the EBMT system, treating SMT as a black box. A detailed description of OpenMaTrEx can be found in (Dandapat et al., 2010a; Banerjee et al., 2011).

2.3 Translation Memory

A translation memory (TM) is essentially a database that stores source- and target-language translation pairs, called *translation units* (TUs), for effective reuse of previous translations. It is widely used in Computer-Aided Translation (CAT) systems to assist professional translators. When a new sentence is to be translated, a TM engine retrieves an entry from the database whose source side is most similar to the input string and presents it to the human translator. The similarity between the input string and the source-side TUs in the TM is often calculated using the edit-distance-based fuzzy match score (Sikes, 2007; He et al., 2010) as in (2.5).

$$\text{FuzzyMatch}(t) = 1 - \min_{s_i} \frac{\text{EditDistance}(s, s_i)}{\max(|s|, |s_i|)} \tag{2.5}$$

where s is the source-side segment to be matched with the TM, s_i is a TU in the TM and t is the TM hit based on the fuzzy-match score.

If a TU in the TM matches the input segment exactly, the translation of this TU can be directly reused without any further processing. In the case of partial matching, the translation is extracted from the database as a skeleton translation

[9] `http://www.openmatrex.org/`

which is post-edited by a human translator to produce the correct translation. In this case, the matched and unmatched parts are presented to the human translator using different colour-codes or highlighting markers in the front-end CAT system.

The TM paradigm emerged when professional translators realized the need to use previously translated material due to the limitations of MT systems at the time. This idea was originally proposed by Martin Kay (1980). He suggested to exploit parts of the previously translated text that contain similar material relevant to the text to be translated. Kay (1980) used the example of anaphora resolution to illustrate the difficulty of decision making in the translation process. During that time, human assistance was required to handle a large number of such problems to produce good-quality translation. He also pointed out the lack of an efficient algorithm for MT at the time, by comparing the complexity of the dictionary search with translation for MT. Later, the problem of exact MT-decoding was proved to be NP-complete[10] (Knight, 1999).

The above arguments were made 30 years ago, when computer systems were not powerful enough (in contrast to the computing power available today) and when the concept of an SMT system did not exist. However, the major points highlighted by Kay (1980) still hold. The paradigm of human-centric translation to support human translators using TM continues to evolve.

2.3.1 TM Technology

The success of a TM-based system depends on how helpful the retrieved TUs from the TM are to assist a human translator producing a translation for the corresponding input segment. This primarily relies on three technologies: (i) efficient storage and acquisition of data, (ii) fast and efficient source-segment searching in the TM database, and (iii) guidance for target-side changes.

A TM system is often helpful when a segment in the database has a high fuzzy-

[10]NP-complete is a class of decision problem in computational complexity theory. A problem τ is said to be NP-complete if the problem belongs to the set of NP problems and every problem in the set NP is reducible to τ in polynomial time.

match score when compared with the input segment. This happens when enough in-domain relevant data is stored in the TM (He, 2011). It was reported that TMs are quite useful when there exists a large portion of exact matches and may be useless when the TM is full of low fuzzy matches. Thus, it is important to collect a significant amount of data for the TM to obtain high fuzzy match scores. In general, TM users do this in two ways. Firstly, the data can be collected from the translation process itself. In addition to the initial TM database (source–target sentence pairs), additional subsentential entries are added based on the edits performed by a human translator in the CAT tool. Secondly, the TM users can share and exchange TM data with each other (although this practice is more common between individual users rather than translator departments). This is possible due to the wider acceptance of the TMX (Translation Memory eXchange) format[11] in the industry. Some such other widely-accepted formats for storing TM data are XML Localization Interchange File Format (XLIFF)[12] and Universal Terminology eXchange (UTX).[13]

The second factor that affects the performance of a TM is search. TM users need to find the best match from the database in real time. This area is still under active research with a few recent efforts, e.g. Koehn and Senellart (2010b) used an n-gram-based matching method to find the potential candidates from a large database. Then, A*-search was applied to filter some candidates, and finally they used A* parsing to validate the matched segment. Their method outperformed the baseline (the dynamic programming solution of the string edit distance problem) by a factor of 100 in terms of the speed of lookup time.

Another factor that affects the source-side fuzzy-match score is the strictness of matching. Often, two words are considered to be matched if they have the exact same surface form. Some systems relax this premise. For example, SDL Trados[14] assigns some credit to partially matched words. Using the example in (17), using

[11] http://www.gala-global.org/oscarStandards/tmx/tmx14b.html
[12] http://www.opentag.com/xliff.htm
[13] http://www.aamt.info/english/utx/index.htm
[14] http://www.translationzone.com/en/translation-agency-solutions/translation-memory/

strict matching, words *restoring* and *restored* will not be considered as matches, as their forms are different. However Trados considers *restoring* and *restored* to be partial matches and adds a fraction into the segment-level fuzzy-match score.

(17) a. **Source Segment:** Determines whether a recovery point is valid or corrupt before restoring it

b. **The Fuzzy Match:** Verifies whether a recovery point is valid or corrupt before it is restored

The third factor that can increase the performance of a TM system is the target-side alignment. In general, the TM system looks for a source-side match for the input segment and highlights the difference between the best-matched TUs and the input segment. However, this does not provide any clue about the segments in the target language that need to be changed. Some recent work has explored the possibility of marking possible changes in the target segments to assist the human translator. For example, Esplà et al. (2011b) used word alignments to predict which target words have to be changed and which should be kept unedited. They showed that their approach worked with high precision for higher fuzzy match scores. Furthermore, Esplà et al. (2011a) computed the alignment strength using an MT system to provide the target-language edit hints.

2.3.2 Synergy between MT and TM

Although TMs are widely used in CAT systems to assist professional translators, they are often linked with EBMT. TMs can be used to store examples for EBMT systems. EBMT systems first find the example (or a set of examples) from the TM which most closely matches the source-language string to be translated (Somers, 2003). After retrieving a set of examples with associated translations, EBMT systems automatically extract the translation of the suitable fragments and combine them to produce a grammatical target output. On the other hand, CAT systems segment the input text to be translated and compare each segment against the TUs

in the TM (Bowker, 2002). CAT systems produce one or more target equivalents for the source segment and professional translators select and may optionally recombine them (perhaps with modification) to produce the desired translation. Both EBMT and CAT-based systems are developed based on a similar premise, but in an EBMT approach selection and recombination is done automatically to produce the translation without the help of a professional translator.

Phrase-based SMT systems (Koehn, 2010) produce a source–target aligned subsentential phrase table which can be adapted as an additional TM to a CAT environment (Simard, 2003; Bourdaillet et al., 2009). SMT phrases have also been used to populate the knowledge database of an EBMT system (Groves and Way, 2006). Biçici and Dymetman (2008) used *Dynamic Translation Memory* (DTM) to improve the translation quality of a phrase-based SMT system. The DTM method does the following:

(i) Looks for the best matching source–target pair $\langle s', t' \rangle$ from the TM for the input s.

(ii) Finds the longest common substring (P_s) between s and s'.

(iii) Identifies the target correspondence P_t of P_s using word alignment.

(iv) Dynamically adds the $\langle P_s, P_t \rangle$ pair to the phrase-table of the SMT system and produces the translation for s.

Note, that the substring P_s can be a non-contiguous word sequence. The SMT system used by Biçici and Dymetman (2008) had the advantage of handling non-contiguous phrase pairs. Similar work was also carried out by Simard and Isabelle (2009) to improve translation quality by adding translational information from fuzzy matches. They used single best source–target fuzzy matching pairs from the TM for an input sentence to compute all possible admissible phrase pairs.

Koehn and Senellart (2010a) used TM to retrieve matches for input segments, and replaced the mismatched parts using an SMT system to fill the gaps in the target-side. Zhechev and van Genabith (2010) used a sub-tree alignment technique to align source–target pairs from the TM to detect gaps with the new input segment

and used the SMT system to fill those gaps to maximize performance.

However, to the best of our knowledge, the use of SMT phrase tables within an EBMT system as an *additional* subsentential TM has not been attempted so far. Some work has been carried out to integrate MT in a CAT environment to translate the whole segment using the MT system when no matching TU is found in the TM. The TransType system (Langlais et al., 2000) integrates an SMT system within a text editor. The TransType2 system (Macklovitch, 2006) combines the positive aspects of the MT and CAT paradigm within a single environment. TransType2 includes a data-driven MT engine to assist the translator with suggestions. Our approach attempts to integrate the TM obtained from an SMT system within an EBMT system.

It is often the case with homogeneous data that a large segment of the input test sentence matches one of the sentences in the example-base. This approach seems to be effective for a sentence with a high similarity to the example-base, as only a small change is required to produce the output. Thus, this approach is anticipated to answer research questions RQ1, RQ2 and RQ4. In contrast, SMT relies on a probabilistic model of words/phrases to produce translations, which does not guarantee capturing homogeneity inherent in the data. Allowing word/phrase of varying length to compete with each other in determining the most probable path through the decoding space means that an SMT system does not guarantee the selection of those subsegment matches (chunks/phrases) which have the longest coverage. When dealing with homogeneous data we want to take full advantage of this homogeneity by exploiting longer subsegment matches to help improve the quality of the MT output by minimising errors, a characteristic inherent in EBMT systems.

2.4 Evaluation Metrics

Evaluation of MT output is essential in the development of an MT system. During evaluation, machine-translated texts need to be judged on their clarity, style and accuracy. Conducting this task manually by human evaluators is difficult and time-consuming. Nowadays, automatic evaluation metrics have become an integral component for the development of any MT system. These metrics use the principle that the closer the hypothesis translation[15] is to the professionally produced reference translation[16], the better the quality. The use of automatic evaluation metrics makes this task faster and cheaper by comparing the output translation to one or more reference translations. In addition, automatic metrics allow a large-scale analysis of an MT system. Some such widely used automatic MT evaluation metrics are: Sentence Error Rate (SER), Word Error Rate (WER), BLEU (Papineni et al., 2002), NIST (Doddington, 2002), METEOR (Banerjee and Lavie, 2005), Translation Edit Rate (TER) (Snover et al., 2006), etc. BLEU, METEOR and TER represent three different design considerations: BLEU uses n-gram precision to ensure translation fluency and fidelity; METEOR relies on unigrams and linguistic resources; and TER measures number of edits between candidate and reference translations. In our experiment, we choose two of the most widely used (extensively used in large-scale MT evaluation campaigns) metrics: BLEU and NIST.

2.4.1 BLEU

The BLEU (Papineni et al., 2002) metric estimates translation quality by comparing the MT output against one or more reference translations. It uses n-gram co-occurrence statistics i.e. the number of n-grams that occurs in both the output translation and in the reference translation. BLEU rewards those translations which contain longer n-gram matches. The main score calculated by this metric is

[15]Hypothesis translations are the candidate translations produced by MT system.

[16]Human translations which serve as the gold standard are called reference translations. Reference translations are used to evaluate the quality of hypothesis translations.

a modified n-gram precision for each candidate translation and its reference(s). The modified n-gram precision p_n is calculated based on Equation (2.6).

$$p_n = \frac{|c_n \cap r_n|}{|c_n|} \tag{2.6}$$

where,

- c_n is the multiset of n-grams occurring in the candidate translation.
- r_n is the multiset of n-grams occurring in the reference translation.
- $|c_n|$ is the number of n-grams occurring in the candidate translation.
- $|c_n \cap r_n|$ is the number of n-grams occurring in c_n that also occurs in r_n such that the elements occurring j times in c_n and i times in r_n occur maximally i times in $|c_n \cap r_n|$.

While p_n can be calculated for any value of n, Papineni et al. (2002) combined the individual scores for all values of n into a single metric for greater robustness. It is often the case that the value of p_n decreases exponentially as the value of n increases. This is because typically fewer matches are found between the MT output and the reference translation with a higher value of n. BLEU uses a weighted average of logarithm of p_n for a range of values for n,[17] using a uniform weight $1/N$ as given in Equation (2.7).

$$p_N = \exp(\sum_{n=1}^{N} \frac{1}{N} \log p_n) \tag{2.7}$$

Candidate translations that are longer (in words) that their reference(s) are implicitly penalized when calculating p_n. In addition, BLEU also uses a brevity penalty (BP), as given in Equation (2.8) to penalize candidate translation (C) that are shorter in length (in words) compared to its reference translation (R).

$$BP = \exp^{\max(1-\frac{length(R)}{length(C)},0)} \tag{2.8}$$

[17]Papineni et al. (2002) found that a maximum value of $n = 4$ is sufficient for adequate correlation with human evaluation.

If the candidate translation C and the reference translation R have the same number of words, the BP is 1.0, and this value increases with shorter Cs compared to the reference translation R. The BP is calculated over the entire test set to avoid individually punishing shorter sentences. Finally, this penalty is multiplied with the modified precision score p_n to produce a single score for the entire candidate translation set as in Equation (2.9).

$$BLEU = BP.p_N \tag{2.9}$$

Note that the value of BLEU ranges from 0 to 1. However, it is often reported as a percentage score between 0% and 100%.

2.4.2 NIST

The NIST (Doddington, 2002) metric is a variation of the BLEU metric with three changes.

First, NIST addresses the issue of n-gram informativeness. BLEU assigns equal weights to all n-grams when calculating the modified n-gram precision. In contrast, NIST assigns greater weight to those n-grams which are infrequent relative to their $(n-1)$-gram prefix, i.e. they are less predictable given the immediately preceding context. These n-gram counts are estimated from a very small reference translation set (usually in the order of two to three thousand). Thus, in practice, long n-grams receive very low weights. As a result of the generally lower weights assigned to longer n-grams, the NIST score obtains the bulk of its value from unigram matches. This emphasis on unigram matches is the reason for its greater correlation with adequacy than with fluency. The informativeness is calculated based on Equation (2.10) and accordingly incorporated into the modified n-gram precision score in Equation (2.11) based on Equation (2.6).

$$I_{w_1 \ldots w_n} = \log_2 \left(\frac{count(w_1 \ldots w_{n-1})}{count(w_1 \ldots w_n)} \right) \tag{2.10}$$

$$p_n = \frac{\sum_{\forall w_1 \ldots w_n \in |c_n \cup r_n|} I_{w_1 \ldots w_n}}{|c_n|} \tag{2.11}$$

Secondly, during the combination of all p_n into a single score p_N, BLEU uses the sum of the logarithm of each value of p_n and multiplies by a weight $1/N$ in order to make p_N more sensitive to larger values of n. Doddington (2002) points out that this method is equally sensitive to varying co-occurrence frequencies regardless of the value of n. He suggested a simple arithmetic average of all the values of p_n to estimate the single combined score p_N as in Equation (2.12).

$$p_N = \sum_{n=1}^{N} p_n \tag{2.12}$$

Finally, an alternative brevity penalty was suggested to minimize the changes in score due to small variations in length. This is done by introducing a variable β, as in Equation (2.13). The value of β is chosen in such a way that BP becomes 0.5 when the ratio of the number of words in candidate translation C and the average length of words in the reference translation set R is $\frac{2}{3}$.

$$BP = \exp\left(\beta \cdot log_2[min(\frac{length(R)}{length(C)}, 1)]\right) \tag{2.13}$$

The BP is then multiplied by the single average modified n-gram precision score p_N to obtain the final NIST score.

Although NIST is a variant of BLEU metric, these two metrics differ in the following ways: (a) BLEU assigns equal weights to each n-gram pair, while NIST assigns higher weights to the less predictable (i.e. more informative) n-grams, (b) BLEU calculates the logarithmic average of n-gram precision, while NIST calculates the arithmetic average, (c) BLEU and NIST differ from each other with respect to how they calculate the brevity penalty (used to prevent shorter candidate translations from receiving too high scores).

2.5 Data Sources

In this section we provide the background of the data that has been used for the experiments in this thesis. Due to the unavailability of large amount of parallel data for some language pairs, and on the basis of research questions RQ1 and RQ2 (cf. Section 1.1), we have used limited amounts of homogeneous domain-specific data. Furthermore, in connection to our research question RQ4, we used much larger data sets in order to test the scalability of the proposed methods.

2.5.1 English–Bangla Patient Dialogue Corpus

The English–Bangla patient dialogue corpus was constructed in-house for the purpose of developing an MT system to assist patients with limited English in a healthcare scenario. This is a multimedia six-way parallel corpus (Morrissey et al., 2010) and two of these dimensions are English text and its Bangla translation. The creation of this corpus involved two different tasks.

Our first task was to collect an English-language corpus of patient–receptionist dialogue. It is difficult to collect medical data due to the involvement of personalized information. Thus we had to consider a number of confidentiality and related ethical issues. This difficulty has long been recognized in medical training where "standardized patients" (SPs) are used with medical students, i.e. actors trained to simulate consistently the responses of a patient in a particular medical setting. Training SPs is, of course, a major undertaking in itself necessarily involving experts, so we made a compromise in that we engaged an experienced GP's receptionist to participate in a number of role-play sessions with native English speakers. These were all recorded and later transcribed. Thus, we believe that our corpus contains samples that are realistic, and offer a broad coverage of our target domain. Due to the involvement of the aforementioned stages, it is time-consuming and expensive to collect a large amount of medical receptionist dialogue. Thus our corpus comprises 380 dialogue turns. In transcription, this works out at just under 3,000 words (a

very small corpus by any standard). Each sentence turn is on average 8 words.

The next stage in the process was to manually translate our English corpus into Bangla. Translation between any languages, whether related or not, involves cases where closely following the source text (a "literal" translation, within the grammatical constraints of the target language) can result in a stilted, unnatural or incorrect translation. This is especially the case when translating medical receptionist dialogue between English and Bangla which differ greatly syntactically. This has serious implications for our approach to MT. A good example is the dialogue in (18):

(18) a. Which doctor would you prefer?

b. I don't mind.

The response (18b) can be translated in the following ways as shown in (19).

(19) a. আমি কিছু মনে করব না।

Ami kichhu mane karba nA.

I ANYTHING MIND WILL NOT.

I don't mind.

b. যে কোনো একজনকে দেখালেই হবে।

ye kono ekajanake dekhAlei habe.

ANY NULL ONE CAN-SEE BE-Future

Can see either of them.

The literal translation (19a) without the context would be misleading or meaningless. Therefore, in Bangla, the literal translation (19a) is less preferable than the more explicit translation (19b). We keep (19b) as the translation of (18b) even though none of the English words have an equivalent in the target side. This scenario might affect an SMT system trained on such a corpus. However, to ensure the closeness and fluency in the dialogue we concentrate on meaningful translation in context. This issue occurs quite frequently while translating the English dialogue corpus into

Bangla. Along with this we have found other difficulties such as lexical choice and the translation of borrowed words.

Although this corpus is very small, we have found that the medical receptionist dialogue is comprised of very similar sentences. This is illustrated in examples (20) and (21).

(20) a. Is it possible to book an appointment *later this week*?

b. Is it possible to book an appointment *with the nurse*?

(21) a. The doctor told me to *come* back *for* a *follow up appointment*.

b. The doctor told me to *call* back *in* a *week*.

The portions in italics are the only differences between (a) and (b). Thus, it may be helpful to reuse the translation of the common parts when translating a new sentence. The above observation informs our decision to use EBMT for the translation of homogeneous domain-specific data.

2.5.2 Other Data

In addition to the above in-house data, we availed of data from other sources which have been widely used over the years for many MT experiments. The following additional data sources have been used in our experiments in this thesis:

BTEC Data: The Basic Traveller Expression Corpus (BTEC) was developed as a part of the C-STAR (International Consortium for Speech Translation Advanced Research)[18] project. The corpus comprises tourism-related sentences similar to those that are usually found in phrasebooks for tourists going abroad. We used the portion of data that has been released for the International Workshop on Spoken Language Translation (IWSLT09)[19] evaluation campaign. We used data for two language pairs:

[18]Main website `http://www.c-star.org`, corpus website `http://cstar.atr.co.jp/cstar-corpus/`.

[19]`http://mastarpj.nict.go.jp/IWSLT2009/2009/12/evaluation-campaign.html`

English–Chinese and English–Turkish. The IWSLT09 English–Chinese and English–Turkish data consists of 44,164 and 19,972 training sentences, respectively. For both language pairs, we used IWSLT09 development sets as test sets in our experiments. The IWSLT09 development sets consist of 489 sentences for English–Chinese, and 414 for English–Turkish. Note, that this data belongs to a single domain (travel) and is, therefore, homogeneous in nature. Details of the corpus are given in Table 2.1. We compare the type-token ratio (TTR)[20] on the source side (English) between these corpora against the Europarl data (selecting the same number of sentences randomly). The low TTR indicates that sentences in the corpora share many surface words between them.

EMEA Data: This corpus was created using documents from European Medicines Agency (EMEA)[21] (Tiedemann and Nygaard, 2009). The corpus is available as both translation memory files (TMX) and plain text files. The corpus originally consisted of 1.09 million parallel sentence pairs. However, there were a large number of duplicate sentences in the original corpus. We removed all the duplicate sentences creating a set of 260,806 unique sentence pairs to use in our experiments. We randomly selected a set of 10,000 examples[22] for testing and the remaining 250,806 examples were used for training in the experiments conducted in this thesis. This corpus also belongs to a single domain with homogeneous examples. Table 2.1 shows that the TTR for this corpus is much lower compared to the TTR obtained from Europarl data using the same number of randomly selected sentences.

JRC-acquis Data: The JRC-acquis (Steinberger et al., 2006)[23] is a freely available parallel corpus developed by the Joint Research Centre (JRC)[24] using legal doc-

[20]the type-token ratio is a measure of vocabulary variation within a written text. $Type-token\,ratio = (number\,of\,types/number\,of\,tokens) \times 100\%$

[21]`http://opus.lingfil.uu.se/EMEA.php`

[22]Compared to the standard size of testsets (of the order of 2000-3000 sentences), we use a larger testset (comprising 10000 sentences) to come up with more reliable results . However, for some of the experiment we used 2000 examples for faster evaluation of the translation system.

[23]`http://optima.jrc.it/Acquis/`

[24]`http://langtech.jrc.ec.europa.eu/`

Table 2.1: Corpus Statistics. TTR: type-token ratio, Bn: Bangla, En: English, Fr: French, Tr: Turkish, Zh: Chinese.

Corpus	no. of sentences	avg. length (in words)	TTR	TTR in Europarl
In-house En–Bn	380	8.51	16.12	24.92
IWSLT09 (En–Zh)	44,164	8.87	2.15	3.72
IWSLT09 (En–Tr)	19,972	9.45	3.81	5.38
EMEA (En–Fr)	250,806	18.8	1.22	1.89
JRC-acquis (En–Fr)	753,323	23.84	1.14	0.95

uments from the Acquis Communautaire (AC). The data was crawled from selected websites of the European Commission and converted into UTF-8 encoded XML format. These automatically crawled documents were aligned using HunAlign (Varga et al., 2005), a language-independent sentence aligner. This corpus represents a larger amount of data and from a less homogeneous domain. We use English–French sentence pairs from this corpus. The English–French corpus originally consisted of 1.25 million sentence pairs. Here, we also removed duplicate sentences from the entire corpus resulting in a set of 755,323 sentence pairs. We randomly selected a set of 2,000 sentence pairs as a test set and the remaining 753,323 examples were used to train the MT systems.

In addition to the sentence-level MT task, we also address the Named Entity (NE) transliteration task using PA-based EBMT. This data is taken from the NEWS2009 English-to-Hindi NE transliteration shared task data (Haizhou et al., 2009). The data consists of 10,000 parallel NEs for training and 1,000 NEs for testing. More details of the task and the data are given in Section 3.3.

2.6 Summary

Data-driven approaches to MT now dominate the field of research. In this chapter, we review the two main data-driven approaches to MT: EBMT and SMT. Both of these corpus-based approaches facilitate the quick and inexpensive development of an MT system without the need for vast linguistic expertise that was required for previous transfer-based approaches.

Furthermore, we discussed the main principles behind EBMT, including how different matching techniques are performed to find useful examples from the example-base. We described in detail two different EBMT approaches in particular (a pure EBMT approach using proportional analogy and a compiled EBMT approach using translation templates) that have been used in our work. We explained how these two EBMT approaches can work effectively in a limited resource setting with homogeneous data (particularly using proportional analogy on IWSLT data).

In terms of the SMT framework, we briefly discussed the earlier word-based translation models and the different components of the more recent phrase-based SMT system using generative models. The more recent system uses log-linear discriminative models that have the advantage of being able to use a larger number of features compared to the earlier noisy-channel models.

We also looked at the TM paradigm and its use in a CAT system. Much recent research focuses on integrating TM and MT in order to improve each of these paradigms. SMT phrases can be used as an additional TM to improve a CAT system. In contrast, TM-based matching example pairs have also been used to improve SMT systems.

Finally, we described the different evaluation metrics and data sets that are used in this thesis. Based on the analyses of the EBMT, SMT and the TM paradigms and the data we are interested in, in what follows, we first present our work using an EBMT system in Chapter 3 and 4. Subsequently, in Chapter 5 and 6, we propose a novel way of using TM within an EBMT system to meet our translation needs.

Chapter 3

Proportional Analogy-Based EBMT

In this chapter we describe a proportional analogy-based EBMT system. In 2005, a number of papers by Lepage and Denoual (2005a,b,c) reported an experimental implementation of an EBMT system using proportional analogy (PA). This approach, a type of analogical learning, was attractive because of its simplicity; and the paper reported considerable success with the method using various language pairs. However, the approach has the problem of low recall and suffers badly with a long run time when the size of the example-base is increased. While the approach seems fraught with difficulties as a stand-alone translation model, its use for the special case of unknown words, particularly names or special terms, seems much more promising. This motivates us to use a PA-based system (as part of a runtime EBMT approach) to mitigate the problems that SMT has with unknown words. Thus a PA-based system is anticipated to address research question RQ1 (which focuses on exploiting EBMT approaches in resource-poor settings) and research question RQ3 (which concentrates on effective combination of EBMT and SMT to handle problems of data sparseness). In our own work, we have developed an analogy-based EBMT system from scratch as no open-source PA-based system existed. Furthermore, we have developed a new heuristic and compared all the proposed heuristics

to understand their effectiveness within the runtime EBMT approach. Finally, we combine the PA-based system with a state-of-the-art SMT system for effective use of the individual system.

The organization of the chapter is as follows: First we describe the underlying concept of using PA-based EBMT system in Section 3.1. Then we present our particular approach for developing an EBMT system using analogy. Finally, we demonstrate different experiments conducted and present the experimental results and assessment of error types.

3.1 The Underlying Idea of a PA-Based System

The PA-based EBMT system was developed based on the idea of constructing and solving analogical equations at runtime. This particular approach to MT was introduced by Lepage and Denoual (2005c). We will first outline the theory of PA then how this idea can be implemented in an EBMT system.

3.1.1 Proportional Analogies

PAs are global relationships between four objects as shown in 3.1.

$$A : B :: C : D \tag{3.1}$$

read as "A is to B as C is to D". The symbol '::' is sometimes replaced with an equal sign (=) to denote (3.1) in the form of an equation. This formulation as an equation can have zero, one, or more solutions if any of the objects (usually D) is considered as a variable. Noted long ago by the likes of Aristotle and Plato, PAs are often seen as a means of knowledge representation in Artificial Intelligence (Gentner, 1983) due to their power to represent world knowledge and the lexical relations encoded within them. In natural language processing, analogies are used as an instrument to explain inflectional and derivational morphology including complexities such as those found

in Semitic languages (Lepage, 1998). Lepage (1998) developed an algorithm that could solve analogical equations over strings or characters, based on finding the longest common subsequences, and measuring edit distance. Lepage showed with examples from various languages that his algorithm could handle insertion/deletion of prefixes and suffixes (22a), exchange of prefixes/suffixes (22b), infixing (22c) and parallel infixing (22d).

(22) a. (French) *répression* : *répressionnaire* :: *réaction* : $x \Rightarrow$ *x=réactionnaire*

b. *wolf* : *wolves* :: *leaf* : $x \Rightarrow$ *x=leaves*

c. (German) *fliehen* : *floh* :: *schließen* : $x \Rightarrow$ *x=schloß*

d. (Proto-Semitic) *yasriqu* : *sariq* :: *yanqimu* : $x \Rightarrow$ *x=naqim*

3.1.2 Analogy-Based EBMT

In the EBMT workshop in Phuket, Lepage and Denoual (2005c) presented "The 'purest' EBMT system ever built: no variables, no templates, no training, examples, just examples, only examples". This purely data-driven approach to MT uses the notion of PA. The idea introduced in Lepage and Denoual (2005c) is explained in considerably more detail in Lepage and Denoual (2005a,b).

Lepage and Denoual (2005a,b,c) showed how an EBMT system can be built based on the algorithm proposed by Lepage (1998). Treating a sentence as a string of characters, they note that PAs can be handled as in (23):

(23)

They swam in the sea	:	*They swam across the river*	::	*It floated in the sea*	:	*It floated across the river*

For the purpose of EBMT, the PA-based approach assumes a database of example pairs, where each pair is a source and target language translation equivalent. For the first three sentences in (23), the translation equivalents in Spanish are given in (24).

(24) a. Nadaron en el mar. b. Atravesaron el río nadando. c. Flotó en el mar.

Suppose now that we want to translate the sentence D=*It floated across the river*. The translation process is as follows:

1. Find a pair $\langle A, B\rangle$ of sentences in the example set that satisfies the PA in Equation (3.2).

$$A : B :: C(?) : \textit{It floated across the river} \tag{3.2}$$

   Solving this results in C = *It floated in the sea*.

2. Take the translation corresponding to A, B and C (noted A', B' and C').

3. Solve Equation (3.3): D' represents the desired translation.

$$A' : B' :: C' : D' \tag{3.3}$$

Substituting the three sentences in (24) into Equation (3.3), we have a solvable equation with D'=*Atravesó el río flotando*, which is an acceptable translation.

3.1.3 Some Immediate Difficulties

The process outlined in the previous section has some difficulties in solving analogical equations. First, due to the unconstrained nature of PA, there is always a possibility of solving "false analogies", i.e. set of strings for which the analogy holds, but which do not represent a valid linguistic relationship. Example (25) illustrates this phenomenon, where the $A : B$ relationship is a simple one-character substitution (p for a), mirrored in the case of $C : D$.

(25) *Yea* : *Yep* :: *At five a.m.* : *At five p.m.*

However, Lepage (2004) reported that there are very few analogies of this kind (less than 4% in BTEC corpus). Secondly, there might be multiple solutions to the Equation (3.3). To take another example from (Lepage, 2004), the solution to (26) could be any of the strings in (27):

(26) *May I have some tea please?* : *May I have a cup of coffee?* :: *I'd like some strong tea please.* : *x*

(27) a. *I'd like a strong cup of coffee.* b. *I'd like a cup of strong coffee.*
c. *I'd like a cup strong of coffee.* d. *I'd like a cstrongup of coffee.*
e. *I'd like a custrongp of coffee.* *etc.*

The equation requires us to substitute *May I have* with *I'd like … strong*, and *some tea please* with *a cup of coffee*, but nothing in the algorithm tells us where to insert the word *strong*, and, remembering that we are treating the sentences as strings of characters rather than strings of words, nothing prevents the word from being inserted as in (27d,e) etc. in addition to the desired solution in (27b). The proportional analogy method can consider the examples to be either strings of characters, or strings of words. The latter approach of course eliminates the possibility of outputs such as (27d,e), but also means that correspondences such as *walks* : *walked* :: *floats* : *floated* as in (28) would not be captured.

(28) *It walks across the street* : *It walked across the street* :: *It floats across the river* : *It floated across the river*

3.2 Our Approach

In this section we describe the system architecture of our implementation of a PA-based EBMT system. Our particular architecture has a clear separation between the main components of an analogy-based EBMT system. These components essentially represent some knowledge for solving valid analogical equations first.

3.2.1 System Architecture

We have implemented the EBMT system using PAs based on Lepage (1998) and Lepage and Denoual (2005a). The proposed architecture integrates the main components of an analogy-based system in a modular fashion with a heuristic-based prevalidation for identification of valid analogical equations.

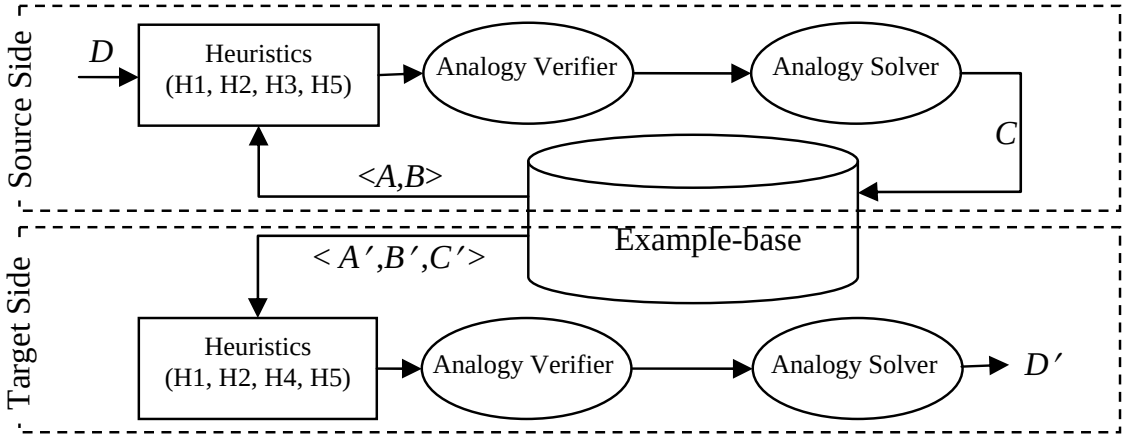


Figure 3.1: Analogy-based EBMT architecture.

First the system requires some knowledge about choosing relevant $\langle A, B\rangle$ pairs from the example-base to ensure that the better candidate analogical equations from the potential set of all possible analogies are solved first, and that some of the unsolvable analogies are filtered out before verification. We adopt different heuristics to ensure this, as discussed below. Secondly, there is an *Analogy Verifier*, which decides the solvability of an analogical equation. The third component solves the analogy as in Equation (3.2) based on the triplet $\langle A, B, D\rangle$ and produces C. Note that D is the input sentence to be translated. We call this module the *Analogy Solver*. Once C is produced on the source side, we find the translation equivalents $\langle A', B', C'\rangle$ on the target side for the source-side $\langle A, B, C\rangle$ triplet. Then, we apply the three components on the target-side in the same order to obtain one candidate translation D' as in Equation (3.3). Collecting all D', we rank them by frequency as different analogical equations might produce identical solutions.

An EBMT system using PAs must address the issue of computational complexity for real-time translation. The first step of the process, mentioned in Section 3.1.2,

can choose any $\langle A, B\rangle$ pair for the input D. Thus, a total of $O(n^2)$ possible pairs need to be examined for D, which itself is a very time consuming process as n is in general a large number and denotes the number of lines in the example-base. Furthermore, one analogical equation is verified and solved based on finding the longest common subsequences and measuring edit distances. These two processes also exhibit quadratic time complexity. To cope with this large time complexity we only look for time-bounded solutions, i.e. allow the process to continue for a fixed amount of time. However, we may still spend time verifying/solving equations which will not converge to any solution. Thus, we apply different heuristics to filter out some of the analogical equations and to try better candidates first by ranking the equations. Section 3.2.2 describes the heuristics in detail.

Since no off-the-shelf implementation is available for solving analogies, we have implemented our own EBMT system using PA. It is often the case that a PA-based system suffers from low recall. First, we tried to improve the PA-based system by introducing new heuristics to overcome low recall. Furthermore, we have improved the system accuracy by combining an SMT-based system with the PA-based system.

3.2.2 Heuristics

We adopted different heuristics from the literature to understand their relative performance in translation tasks under the time-constrained model. Heuristics essentially prune some of the analogical equations that will not produce a fruitful solution. This will effectively reduce the time wasted for verifying and/or solving some analogies. The heuristics do this in different ways, and with varying success. We first choose the heuristic from Lepage and Denoual (2005a,b) which selects a relevant pair $\langle A, B\rangle$ based on a length comparison with the input D.

H1: *Consider as candidates only sentences whose length is more than half and less than double the length of the input sentence. Formally,* $|D|/2 \leq |A|, |B| \leq 2|D|$, *where* $|x|$ *is the length of* x.

The second heuristic is based on that of Lepage and Lardilleux (2007), which speeds up the process of searching relevant $\langle A, B\rangle$ pairs. This is done by sorting the corpus based on the sentence to be translated (D), using edit distance for the selection of As and selecting Bs based on their *inclusion score* (Lepage, 1998, p.730), i.e. length of B minus its similarity to D.

H2: *Consider as candidates primarily sentence pairs where A has a low edit distance w.r.t. D, and B has a low inclusion score w.r.t. D.*

In the third heuristic, we adopt a "trick" described by Langlais and Yvon (2008), called SOURCE-TRICK, relies on the property expressed in (29).

(29) $[A : B :: C : D] \Rightarrow$
$A[1] \in \{B[1], C[1]\} \vee D[1] \in \{B[1], C[1]\}$
$A[\$] \in \{B[\$], C[\$]\} \vee D[\$] \in \{B[\$], C[\$]\}$
where $S[1]$ and $S[\$]$ are the first and last symbols, respectively, in the string S.

The trick is to limit the search to triples $\langle A, B, C\rangle$ that pass this test.

H3: *Consider as candidates only pairs where B or C share the same first and last symbol with A or D.*

The fourth heuristic relates to the effort of solving target-side analogical equations $A' : B' :: C' : D'$ based on Langlais and Yvon (2008) character count property, called TARGET-TRICK. Formally, it can be stated as:

H4: *Whenever a symbol occurs more frequently in A' than it does in B' and C', the analogical equation is bound to fail and need not be solved.*
$[A' : B' :: C' : x] \neq \phi$ if $|A'|_c \leq |B'|_c + |C'|_c, \forall c \in \{A', B', C'\}$

Finally, we have developed a new heuristic based on a modification of H2. Here also, we speed up the process of searching for relevant $\langle A, B \rangle$ pairs. We choose $\langle A, B \rangle$ pairs based on a smaller edit distance with respect to the input sentence to be translated (D). This is done by sorting the examples based on the edit distance with respect to D and choosing the top two candidates as the $\langle A, B \rangle$ pair from the sorted examples. Edit-distance essentially indicates the measure of closeness. Choosing the $\langle A, B \rangle$ pair based on smaller edit-distance to D indicates two similar sentences are used to form the analogical equation in the source side. This also indicates that A and B may be quite close to each other as they are the two most similar sentences to input D. It is quite likely to find a valid solution to these analogies.

H5: *Consider as a candidate a pair sentences where A and B have a low edit distance w.r.t D such that $A \neq B$.*

Comparison of Heuristics

In order to understand the effectiveness of the different heuristics mentioned above, we compare the average number of analogical equations constructed and solved both in the source and target sides in a time-constrained environment (the number of equations attempted or solved within 1 second). We used English–Hindi Named Entity (NE) transliteration data[1] for the comparison of heuristics. The data consists of 10,000 NEs for training and 1,000 names for testing. Table 3.1 summarizes the average number of equations attempted or solved and the average number of analogical equations that produce potential output while different heuristics are used in the system.

Note that when no heuristic is applied, to transliterate one input NE, the average number of analogical equations attempted within 1 second is around 600k equations on the source side and 40k equations on the target side. Out of these 40k target-

[1]The detail of the NE data is provided in Section 3.3. Note that NE transliteration is similar to the machine translation task. However, NEs in general have a shorter length compared to a proper sentence of a language, so we anticipate that PA-based system will work well for the NE transliteration task.

Table 3.1: Average number of analogical equations attempted or solved with different heuristics in source and target sides.

Heuristic	source-side	target-side	output
No heuristic	600,142	40,308	0.692
H1	705,711	3,621	0.335
H2	788,185	42,634	176
H3	791,155	10,203	8.75
H4	703,912	33,291	0.382
H5	673,928	10,705	1900

side equations, the average number of analogical equations that generate the final solution is only 0.692. As we will see, the various heuristics affect the number of equations attempted or solved, ideally cutting down the effort wasted on comparisons which will not contribute to a useful solution.

With the help of **H1**, we are able to solve around 705k analogical equations on the source side and around 3k equations on the target side in 1 second. This heuristic solves more equations on the source side but effectively reduces the number on the target side and the average number of equations that produce output is 0.335.[2] This is reflected in the overall output of the experiments shown in Table 3.4 (in Section 3.3).

We are able to solve around 788k and 42k analogical equations in the source and target sides, respectively, within 1 second with the help of **H2**. We found that with this heuristic, the average number of analogical equations that lead to output are 176. Thus, this is expected to work well with our experimental setups.

The average numbers of source- and target-side analogical equations solved within 1 second with the help of **H3** are around 791k and 10k, respectively, and the average number of analogical equations which produce output is 8.75.

Using **H4**, the average numbers of source- and target-side analogical equations solved within 1 second are around 703k and 33k, respectively. The average number of analogical equations which produce output is 0.382.

[2]The average number of equations indicates the ratio between the total number of solved analogical equations on the target side and the total number of input sentences attempted to be translated using analogy. The number is less then one when the analogy-based approach is unable to find any solution to a large number of input sentences.

We are able to solve around 673k and 10k source- and target-side analogical equations with the help of **H5** within 1 second. However, we found that with this heuristic, the average number of analogical equations that lead to output are 1900. Thus, this is expected to work best with our experimental setup.

3.2.3 Analogy Verifier

It is often the case that an analogical equation has no solution. Thus, we need to verify the solvability of an analogical equation beforehand to avoid the time-consuming procedure of solving the equation. We developed our analogy verifier based on the description in Lepage (1998). An analogical equation, $A : B :: C : x$ has no solution if some characters of A appear neither in B nor in C. Conversely, all characters of A need to appear either in B or in C to form a solvable analogy. Like Lepage (1998), we also compute a *pseudo-distance* matrix and *similitude* to verify the solvability of an analogy.

Pseudo-distance is a variation of the edit-distance (Wagner and Fischer, 1974) with an insertion cost of 0. The pseudo-distance can be computed exactly as the edit-distance with an insertion cost of 0. We refer to this number as $pdist(A, B)$. For instance the edit-distance between the words *like* and *unlike* is 2. The bottom-right element of the array in Figure 3.2 contains the answer after finding the minimum edits (insertion, deletion and substitution) between the two strings of characters. The insertion of character 'u' and 'n' changes 'like' into 'unlike'. Thus the $edit - dist(like, unlike) = 2$.

	u	n	l	i	k	e
l	1	2	2	3	4	5
i	2	2	3	2	3	4
k	3	3	3	3	2	3
e	4	4	4	4	3	2

Figure 3.2: Edit-distance matrix between the words *like* and *unlike*.

	u	n	l	i	k	e
l	1	1	0	0	0	0
i	2	2	1	0	0	0
k	3	3	2	1	0	0
e	4	4	3	2	1	0

Figure 3.3: Pseudo-distance matrix between the words *like* and *unlike*.

As the insertion cost for pseudo-distance is 0, the two insertion operations (insertion of character 'u' and 'n' into 'like') do not add any value to the pseudo-distance. The result is in the right-bottom row of the matrix in Figure 3.3.

Similitude between A and B ($sim(A, B)$) is the length of their longest common subsequence. This is equal to the length of A, minus the number of its characters deleted or replaced to produce B. This number is essentially the pseudo-distance between A and B. Thus,

$$sim(A, B) = |A| - pdist(A, B) \tag{3.4}$$

A valid analogy will hold if the sum of the similitudes of A with B and C is greater than or equal to the length of A.

$$sim(A, B) + sim(A, C) \geq |A| \tag{3.5}$$

Substituting Equation (3.4) in Equation (3.5), we get

$$|A| \geq pdist(A, B) + pdist(A, C) \tag{3.6}$$

When the length of A is greater than the sum of the pseudo-distances, some subsequences of A are common to B, C, x (that has been built so far) in the same order. Such subsequences have to be copied in solution x. We compute the sum of the length of such subsequences. We refer to this as $com(A, B, C, x)$. Thus, an analogical equation $A : B :: C : x$ will hold iff:

$$|A| = pdist(A, B) + pdist(A, C) + com(A, B, C, x) \tag{3.7}$$

3.2.4 Analogy Solver

The first algorithmic solution to an analogical equation was developed by Lepage (1998). He proposed the algorithm for solving analogies between words. We adopted the same algorithm for solving analogies between sentences. The solution proposed by Lepage (1998) works on strings of characters. Thus, the same algorithm works for solving analogies between sentences.

To solve an analogy $A : B :: C : x$, A is compared with B and C to construct the output x. The method works in two steps:

(i) Look for those parts in B which are not common to A and parts in C that are not common to A.

(ii) Put these differing parts in the right order to construct the solution x.

The example in (30) illustrates the above steps.

(30) $reader : \underline{un}read\underline{able} :: \overline{do}er : x => x = \underline{un}\overline{do}\underline{able}$

In this example, the uncommon parts in $B(unreadable)$ compared with A $(reader)$ are $\underline{un}$ and $\underline{able}$. Similarly, the uncommon part in $C(doer)$ compared with A is $\overline{do}$. These three uncommon parts ($\underline{un}$, $\underline{able}$ and $\overline{do}$) are combined in the second step to produce the solution $\underline{un}\overline{do}\underline{able}$.

The algorithm first computes the pseudo-distance matrices between A and B, and A and C. After constructing the matrices, the algorithm computes the solution (x) of the analogy by traversing all possible paths similar to the output of an edit-distance trace. The traversal starts from the bottom to the top in both the matrices in parallel. During each move a character is copied to the solution x (in reverse order) according to the traversal rules. These rules indicate which character from B or C will be copied to the solution based on the different combination of move directions (horizontal, vertical and diagonal) in the pseudo-distance matrices.

In order to avoid repetition, we are not providing the detail of the algorithm. This can be found in Lepage (1998) and Lepage and Denoual (2005c). However, we illustrate the solution of an analogy with an example in Figure 3.4. The example solves the analogy *like* : *unlike* :: *known* : x, where x is the variable. The figure shows the possible moves in the two pseudo-distance matrices in parallel. The moves are marked with suffixes in the circled elements of pseudo-distance matrices.

B	=	e	k	i	l	n	u		k	n	o	w	n	=	C
		0	0	0	⓪$_4$	①$_{5,6}$	①$_7$	l	①$_{5,6,7}$	①$_4$	1	1	1		
		0	0	⓪$_3$	1	2	2	i	2	2	②$_3$	2	2		
		0	⓪$_2$	1	2	3	3	k	2	2	2	②$_2$	2		
		⓪$_1$	1	2	3	4	4	e	3	3	3	3	③$_1$		
								||							
								A							

Figure 3.4: Example of pseudo-distance-based analogy solver.

Table 3.2: Solution associated with moves in pseudo-distance matrices

Move	Dir_{AB}	Dir_{AC}	Copy onto x	rule	from string
1	diagonal	diagonal	-e + e + n = n	-A+B+C	C
2	diagonal	diagonal	-k + k + w = w	-A+B+C	C
3	diagonal	diagonal	-i + i + o = o	-A+B+C	C
4	diagonal	diagonal	-l + l + n = n	-A+B+C	C
5	horizontal	horizontal	k	copy from C	C
6	horizontal	no move	n	copy from B	B
7	horizontal	no move	u	copy from B	B

Based on the direction of moves (Dir_{AB} and Dir_{AC}) in the two pseudo-distance matrices, a different character is copied to the solution x. Table 3.2 shows an example of the characters copied to solution x in each move. The solution is constructed in reverse order due to the bottom up traversal in the pseudo-distance matrices. Thus, the reverse of the string copied in x generates the actual solution of the analogy, i.e. $x = unknown$.

3.3 Experiments and Results

We tested our EBMT system using PA for two different tasks. First, a NE transliteration task from English to Hindi. As noted in Section 2.1.1, the PA-based system works well for shorter sentences with a similar structure. Thus, we took the NE transliteration task as a case study which is relevant to MT. Furthermore, it was reported (Hermjakob et al., 2008) that a state-of-the-art SMT system can't handle NEs that are not found in the training parallel text. We choose an NE transliteration task to see the power of a PA-based system as it works well when the length of the input/training-data is shorter. A short description of the task is given below.

Named Entity Transliteration

Named Entity (NE) transliteration is the process of transcribing NEs across languages. For example, in our English-to-Hindi NE transliteration, given a name in English (e.g. *nisha*) we need to transcribe the name into its equivalent in Hindi (e.g. निशा). The main difference between NE transliteration and MT is that NE transliteration deals with the phonetic translation of names while MT involves meaningful translation across languages. However, the approaches used to solve these two tasks in general share a lot of similarities. Both tasks essentially use an amount of training data to learn alignments between the smaller units of the task. In the case of MT, the aligned words and phrase pairs are learned while translating sentences of a language. Similarly, in NE transliteration, aligned characters and/or syllables (smaller units of a NE) are learned for phonetic translation of NEs. Thus, the state-of-the-art PBSMT can be applied successfully to the NE transliteration task e.g., (Haizhou et al., 2009).

Secondly, we tested the PA-based system in two translation tasks, from English-to-Bangla and English-to-Chinese. This was done in order to test the PA-based system on a translation task which has much higher complexity (the sentences are much longer compared to NEs) than a transliteration task. We choose English–Chinese

data from IWSLT09 as the sentences are short and belong to a single domain. As noted in Section 2.5, if the test and training sentences belong to the same domain they are likely to share a larger number of surface words between them. Our in-house English–Bangla patient dialog corpus also belongs to a single domain with relatively shorter sentences. In both corpora, the input test sentences and training examples are essentially homogeneous in nature. Thus, a PA-based EBMT system is anticipated to work well with these corpora. This is due to the fact that to hold an analogy ($A : B :: C : x$), all characters in A must appear in B and C. This essentially indicates that one sentence of the corpus should be a subsentence of two other sentences.

3.3.1 Experiments Conducted

We conducted experiments with three different approaches.

- **SMT:** First, we conducted an experiment to estimate the baseline accuracy of our approach for both the tasks (English-to-Hindi NE transliteration and English-to-Chinese/Bangla translation). We use OpenMaTrEx[3] (Dandapat et al., 2010a), an open-source **SMT** system as a baseline and compared the results with our approach.

- **Analogy-based EBMT (AEBMT):** Five different experiments were conducted based on our EBMT system using PA for all the tasks. We shall call these analogy-based EBMT (AEBMT) experiments. The five different experiments deal with the five different heuristics described in the previous section. Each of these five experiments was also tested with time bounds of one second and three seconds to understand the effect of time while using an analogy-based system.

- **AEBMT+SMT:** Furthermore, we have found that there are cases where AEBMT produces good output but SMT fails and vice versa. In order to fur-

[3]http://openmatrex.org/

ther improve the output quality, we use a combination of AEBMT and SMT. We assume that the translation of a sentence s produced by the AEBMT and SMT systems are $\mathrm{T_{AEBMT}(s)}$ and $\mathrm{T_{SMT}(s)}$, respectively. We back-off to the SMT system when AEBMT fails to produce any output, to mitigate the problem of AEBMT with SMT (AEBMT+SMT). In order to do that, we combine the outputs of both systems in the order $\mathrm{T_{AEBMT}(s)} + \mathrm{T_{SMT}(s)}$, which automatically uses back-off when $\mathrm{T_{AEBMT}(s)} = null$. For example, if $\mathrm{T_{AEBMT}(s)} = \{o_a^1, o_a^1, ..., o_a^n\}$ and $\mathrm{T_{SMT}(s)} = \{o_s^1, o_s^1, ..., o_s^m\}$, then the ordered concatenation of both outputs produces $\mathrm{T_{AEBMT}(s)} + \mathrm{T_{SMT}(s)} = \{o_a^1, o_a^1, ..., o_a^n, o_s^1, o_s^1, ..., o_s^m\}$. When no output is produced by the AEBMT system ($\mathrm{T_{AEBMT}(s)} = null$), the combination holds the output of the SMT system, i.e. $\mathrm{T_{AEBMT}(s)} + \mathrm{T_{SMT}(s)} = \{o_s^1, o_s^1, ..., o_s^m\}$. We consider only the first output to estimate the transliteration accuracy. Thus, we rely on SMT output iff the AEBMT system failed to produce any output.

Thus we have three systems (AEBMT, SMT, AEBMT+SMT) that are tested with five heuristics (H1, H2, H3, H4 and H5) as well as a situation where no heuristics are used.

3.3.2 Data Used for the Experiments

We use three different data sets for our experiments. The first dataset is comprised of the NEWS 2009 English–Hindi transliteration data (Kumaran and Kellner, 2007). The data consists of 10,000 NEs for training and 1,000 names for testing. The same examples {*philippines*} are represented in three different ways: character-level {*p h i l i p p i n e s*}, syllable-level {*phi li ppi ne s*}[4] and word-level {*philippines*}. All the experiments were tested with character- , syllable- and word-level NEs as the example-base.

Our second set of data consists of an English–Chinese corpus from IWSLT09.

[4]The syllabification is based on the NEWS09 NE transliteration data (Kumaran and Kellner, 2007).

The training data consists of 44,164 parallel sentences. We use the IWSLT09 devset[5] as our test set which consists of 489 sentences.

Our third dataset consists of an English–Bangla parallel corpus developed in-house based on the dialogue exchange between a patient and a medical receptionist (cf. Section 2.5.1). The source side of the data is composed of transcribed audio recordings. We manually translated the English corpus into Bangla. Due to the involvement of the aforementioned stages, it is time-consuming and expensive to collect a large amount of medical receptionist dialogue. Thus our training corpus comprises of 380 dialogue turns. A fixed set of 41 sentences disjoint from the training set was used to test the system. Although when dealing with relatively small data sets it is common to use k-fold cross-validation, for our particular experiments using the PA-based approach we had to ensure that all of the vocabulary contained in the test set was fully covered by the training examples, since when using PA techniques, we cannot form valid analogical equations over out-of-vocabulary items. Thus, carrying out this type of cross-validation is not suitable for the PA-based approach when using the English–Bengali medical data set.

3.3.3 Results

We evaluated the NE transliteration task with the NEWS'09 metrics (Li et al., 2009). The accuracy is defined as the ratio of correct transliterations in the first position to the total number of words to be transliterated. This is shown in Equation (3.8).

$$Accuracy(\%) = \frac{\text{Number of correct transliteration in the first position}}{\text{Total number of words to be transliterated}} \times 100 \quad (3.8)$$

In our evaluation, correct transliteration in the first position refers to the most frequent output. The example outputs for two input names in English (*nisha* and *pakur*) are shown in Table 3.3. We consider '**निशा** (21)' for *nisha* which is a correct translation with respect to the reference data and '**पुकार** (11)' for *pakur* which is incor-

[5] *devset* refers to the development dataset, used to tune the parameters of a machine translation model towards the improvement of the models for real test data.

Table 3.3: Example of transliteration. The numbers in bracket are the frequencies of each transliteration candidate as output.

Input NE	Output Transliterations
nisha	निशा (21), नीशा (9), नशी (5), ि◌नशा (4), नइशा (4)
pakur	पुकार (11), पौरक (6), पाकुर (4), पकुर (2), पुकर (2)

rect even though the output at rank third (पाकुर) is correct as per the reference data. Thus the accuracy for the example is 50%.

We used BLEU (Papineni et al., 2002) (cf. Section 2.4.1) and NIST (Doddington, 2002) (cf. Section 2.4.2) for automatic evaluation of our analogy-based systems for translation tasks.

Table 3.4 summarises the final accuracy achieved by different methods varying the allowable running time to transliterate a single name.

Table 3.4: Transliteration accuracies (in %) for English-to-Hindi with different models using different heuristics. RT: Average Running Time.

Heuristics		Character-Level System Accuracy (%)		Syllable-Level System Accuracy (%)		Word-Level System Accuracy (%)	
		SMT=31.8, RT=1.25 seconds		**SMT=36.2**, RT=0.19 seconds		**SMT=8.7**, RT=0.01 seconds	
		AEBMT	AEBMT +SMT	AEBMT	AEBMT +SMT	AEBMT	AEBMT +SMT
Running Time=1s	No	13.7	32.6	14.2	36.5	15.7	15.7
	H1	9.4	32.3	13.0	35.8	11.2	14.1
	H2	22.2	32.5	21.4	32.6	20.6	20.6
	H3	14.1	32.4	15.4	36.2	15.3	15.3
	H4	9.4	32.2	13.0	35.8	11.2	14.1
	H5	**28.1**	**36.0**	**30.2**	**37.1**	**28.7**	**28.7**
Running Time=3s	No	16.6	33.1	17.2	35.1	17.1	17.1
	H1	16.1	33.0	17.1	34.7	17	17
	H2	23.7	31.9	24.1	33.5	23.2	23.2
	H3	18.3	32.6	18.3	34.3	19.3	19.3
	H4	16.0	33.0	17.2	34.8	17.1	17.1
	H5	**28.9**	**35.7**	**30.3**	**37.0**	**29.3**	**29.3**

Note that the SMT baseline accuracies are 31.8%, 36.2% and 8.7%, respectively for the character-, syllable-, and word-level models. The highest accuracies achieved with EBMT using analogies are 28.9%, 30.3% and 29.3%, respectively for character-, syllable- and word-level models with the H5 heuristic and allowing a 3 second run

Table 3.5: Translation scores obtained for English-to-Chinese MT with **AEBMT** system

Heuristics		**SMT(BLEU=14.22, NIST= 3.61)**	
		BLEU (in %)	NIST
Running Time=1s	No	0.00	0.22
	H1	0.64	0.33
	H2	6.11	1.29
	H3	1.10	0.46
	H4	0.00	0.33
	H5	**6.56**	**1.37**
Running Time=3s	No	0.00	0.23
	H1	0.77	0.41
	H2	6.56	1.33
	H3	3.93	0.89
	H4	0.82	0.43
	H5	**6.74**	**1.41**

time. However, when combining SMT with AEBMT (AEBMT+SMT) the highest accuracies obtained are 36.0%, 37.1% and 29.3% with a relative improvement of 13.2%, 2.5% and 236.8%, respectively for the character-, syllable- and word-level models over the baseline (SMT).

In addition, we conducted an experiment with English and Chinese using the IWSLT09 corpus. Table 3.5 summarizes the results obtained using the AEBMT system with different heuristics and with different allowable running times. As for NE transliteration, the AEBMT system has a lower accuracy compared to the baseline SMT system. These low BLEU scores are due to the quite low recall of the AEBMT system. The AEBMT system is unable to produce any translation for a large portion of the test examples. We combine the AEBMT system with SMT in a similar way to the NE transliteration task. Table 3.6 summarizes the accuracy of the combined system (AEBMT+SMT) with the two highest performing heuristics. We found that the combined system (AEBMT+SMT) has a relative improvement of 1.13% and 0.55%, respectively in BLEU and NIST over the baseline SMT system, thereby indicating negligible improvements in fluency and adequacy, respectively.

We also conducted experiments in the direction of Chinese-to-English. Table 3.7 shows the accuracy obtained by the two highest performing heuristics when

Table 3.6: Translation scores obtained for English-to-Chinese MT with **AEBMT+SMT** system

Heuristics		**SMT(BLEU=14.22, NIST= 3.61)**	
		BLEU (in %)	NIST
Running Time=1s	H2	14.38	3.62
	H5	14.33	3.61
Running Time=3s	H2	14.27	3.63
	H5	14.18	3.61

Input	**能 再 看 一下 菜**单 吗 ?
Reference	can we look at the menu again ?
SMT o/p	can i see a menu ?
AEBMT o/p	*would you mind seing menu again ?*
Analogy Solved	Source Analogy (*A* : *B* :: *input* : *D*) **我 能 再 看 下 菜**单 吗 ? : **我 能** 换 **下 座位** 吗 ? :: ***能 再 看 一下 菜 单 吗 ?*** : **能** 换 **一下 座位** 吗 ? Source Analogy (*A´*: *B´*:: *output* : *D´*) can i see the menu again ? : can i change the seats ? :: ***would you mind seing menu again ?*** : would you mind changing seats ?

Figure 3.5: Analogy-based translation example from Chinese-to-English.

translating Chinese into English. Figure 3.5 depicts one example translation with the solved analogical equations while translating Chinese into English.

Table 3.7: Translation scores obtained for Chinese-to-English MT

Heuristics		SMT(BLEU=29.63, NIST= 6.02)			
		AEBMT		AEBMT+SMT	
		BLEU (in %)	NIST	BLEU (in %)	NIST
Running Time=1s	H2	4.49	0.851	27.98	5.87
	H5	4.75	0.898	28.21	5.90
Running Time=3s	H2	4.56	0.865	27.98	5.87
	H5	4.91	0.923	28.14	5.90

In our third experiment, we tested our AEBMT system with our in-house English–Bangla medical receptionist dialogue corpus. As we pointed out earlier, the medical receptionist dialogue corpus is very small but the training and test data are homogeneous in nature. We thought that this scenario would be best suited for PA-based EBMT system. We found that within the allowable running time of 1 second, the

AEBMT system is able to handle all possible analogical equations that can be constructed using the entire training corpus. However, the AEBMT system is able to translate only two sentences from the testset of 41 sentences. This is due to the fact the PA-based system was unable to produce solvable analogical equations from such a small corpus. Due to such low recall, we did not estimate the MT accuracy for this corpus using the AEBMT system. The translations of these two sentences by both AEBMT and SMT are exactly the same, thus the combination of AEBMT with SMT has no effect in the MT accuracy over the baseline SMT system. The solutions of these two sentences are given in Figure 3.6 with the corresponding analogies solved to achieve the output.

Input1	i don't know , i have no idea .
Reference	আমি জানি না, আমার কোন ধারনা নেই।
SMT o/p	আমি জানি না , আমি ঠিক বুঝতে পারছি না ।
AEBMT o/p	*আমি জানি না , আমি ঠিক বুঝতে পারছি না ।*
Analogy Solved	Source Analogy: (*A* : *B* :: *input* : *D*) i don't know , you tell me . : you tell me . :: ***i don't know , i have no idea .*** : i have no idea . Target Analogy: (*A´*: *B´*:: *output* : *D´*) আমি জানি না , আপনি আমাকে বলুন । : আপনি আমাকে বলুন । :: ***আমি জানি না , আমি ঠিক বুঝতে পারছি না ।*** : আমি ঠিক বুঝতে পারছি না ।
Input2	is he a new patient ?
Reference	উনি তিনি কি নুতন রোগী ?
SMT o/p	তিনি কি নুতন রোগী?
AEBMT o/p	তিনি কি নুতন রোগী?
Analogy Solved	Source Analogy: (*A* : *B* :: *input* : *D*) are you a new patient ? : are you an existing patient ? :: ***is he a new patient ?*** : is he an existing patient ? Target Analogy: (*A´*: *B´*:: *output* : *D´*) আপনি কি নুতন রোগী? : আপনি কি পুরনো রোগী ? :: ***তিনি কি নুতন রোগী?*** : তিনি কি পুরনো রোগী ?

Figure 3.6: Translation output for English-to-Bangla system.

3.3.4 Further Study with NE Transliteration

Combination of Heuristics

From our previous experiments (cf. Table 3.4), we found that the use of heuristics generally improves the performance of the AEBMT system. We combined heuristics to investigate their effective usage within the analogy-based EBMT system. We tried combining H4 with H2 and H5 because H4 is based on character constraints while both H2 and H5 are distance-based heuristics. Figure 3.7 shows the performance of the combined heuristics (H2+H4 and H4+H5) for the syllable-level[6] NE transliteration task. We found that the use of H2+H4 improves the performance over H2 and H4 when used in isolation. In contrast, the use of H2+H5 shows an improvement over H2 but fails to improve over H5. However, overall, none of these combinations are able to outperform H5.

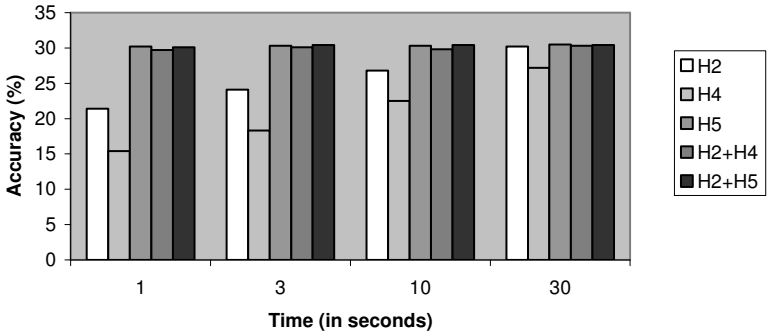


Figure 3.7: The effect of combined heuristics for NE transliteration using AEBMT system.

Furthermore, we observed a similar trend with the combination of heuristics for the syllable-level AEBMT+SMT system. Figure 3.8 shows the effect of different combinations of heuristics for the combined AEBMT+SMT system.

[6]We choose syllable-level as it achieves the highest accuracy, as illustrated in Table 3.4

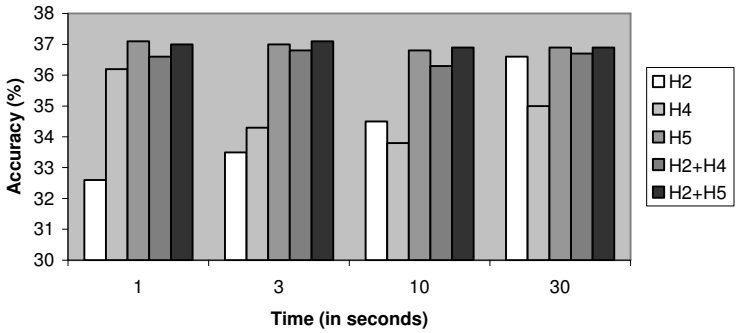


Figure 3.8: The effect of combined heuristics for NE transliteration using the AEBMT+SMT system.

Output-bounded Solution

In our previous experiments, we used a fixed amount of runtime to obtain output translations using AEBMT. Although using a fixed amount of allowable runtime may not produce a sufficient number of solutions to reliably select the most frequent one, we can rely on the output if the number of solutions produced by AEBMT ($|op|$) is greater than some threshold. However, we may encounter test examples which cannot produce a sufficient number of solutions to satisfy the threshold value, thereby creating an infinite loop. Therefore, it is often risky to impose this criterion within the PA-based system.

We use different possible combinations of AEBMT with SMT using different threshold values for $|op|$ to understand the reliability of the most frequent output. In order to do this, we only rely on the AEBMT output when the number of solutions is greater than some threshold ($|op| \geq x$), otherwise we back-off to SMT. Table 3.8 shows the accuracies under different output-bounded combinations of the AEBMT and SMT systems. Here we again found that H5 performs better than all other heuristics. However, we found that for most of the heuristics other than H5, the performance of the combined system increases with the increased $|op|$ threshold. This is because most of these heuristics have a lower accuracy than the SMT system when using $|op| \geq 1$. With the increased thresholds ($|op| \geq 50$ and $|op| \geq 100$), the system uses fewer solutions from AEBMT, and thereby achieves gains in accuracy

with the help of SMT-based output. However, when using H2, we found improved accuracy over the baseline SMT system when using $|op| \geq 50$ and $|op| \geq 100$ and an allowable runtime of 1 second or 3 seconds. The use of H5 always achieves greater accuracy than the baseline SMT system. However, with the increased thresholds ($|op| \geq 50$ and $|op| \geq 100$), H5 has lower accuracy than when $|op| \geq 1$. This essentially signifies that even when the number of output solutions is less, the H5 heuristic is able to produce the correct transliteration (in the most frequent position) in the majority of cases.

Table 3.8: Transliteration accuracies (in %) for English-to-Hindi transliteration using different heuristics under different output-bounded combinations. RT: Average Running Time.

Heuristics		Systems					
		SMT=36.2, RT=0.19 seconds					
		$AEBMT_{|op|\geq1}$ +SMT		$AEBMT_{|op|\geq50}$ +SMT		$AEBMT_{|op|\geq100}$ +SMT	
		Acc(%)	times AEBMT used	Acc(%)	times AEBMT used	Acc(%)	times AEBMT used
Running Time=1s	No	36.5	261	36.2	3	36.2	2
	H1	35.8	227	36.2	3	36.2	2
	H2	32.6	523	36.8	108	36.3	61
	H3	36.2	277	36.2	3	36.2	2
	H4	35.8	227	36.2	3	36.2	2
	H5	**37.1**	**523**	**36.8**	**108**	**36.3**	**61**
Running Time=3s	No	35.1	357	36.2	14	36.2	14
	H1	35.8	345	36.2	12	36.2	6
	H2	33.5	576	36.4	142	36.7	96
	H3	34.2	396	36.5	32	36.2	6
	H4	34.8	344	36.2	12	36.2	6
	H5	**37**	**576**	**36.4**	**142**	**36.7**	**96**

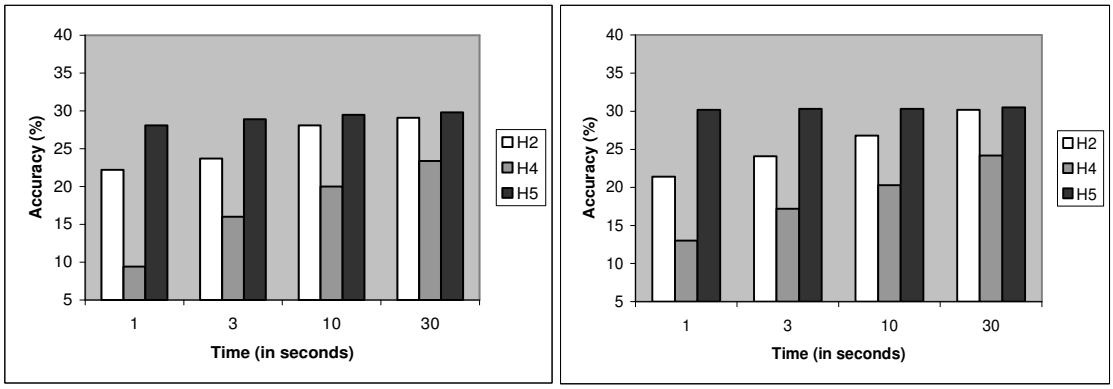
3.3.5 Observations

We found that AEBMT has lower accuracy on its own for both the character- and syllable-level model of the transliteration task. However, the word-level AEBMT models show a huge improvement over the SMT-based models. The claim might be insignificant when transliterating NEs as a task on its own as other models (character- and syllable-level) have higher accuracy. However, in the case of full

text translation, SMT models are trained at the word/phrase level so can only transliterate names that are seen in the corpus. A similar effect has been observed in the case of our word-level NE transliteration experiments. On the contrary, our AEBMT models inherently consider every word/sentence as a string of characters. Thus a significant improvement has been obtained which might be relevant for considering an analogy-based MT system to address unknown words in the standard phrase-based SMT system.

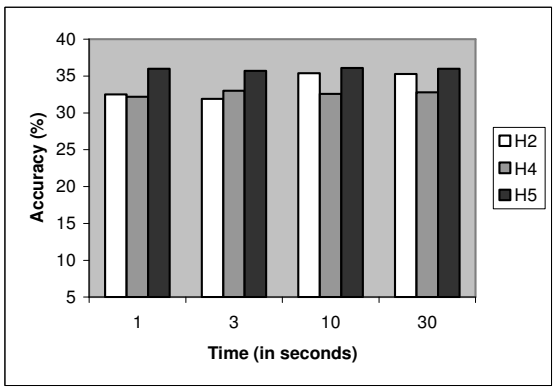
Another significant observation is that AEBMT accuracy increases when a longer time is allowed for the transliteration process. This essentially allows the system to solve more analogical equations to try to produce correct solutions for more NEs. This effect has been observed for all of the heuristics applied in our system when runtime is increased from 1 second to 3 seconds (cf. Table 3.4). Furthermore, we conducted experiments allowing runtime of 10 and 30 seconds , and we found significant improvements with AEBMT for all heuristics other than for H5, but observed no improvement for the combined systems (AEBMT+SMT). The H5 heuristic is able to capture significant amounts of solvable analogies within 3 seconds, so there is no improvement with increased runtime of 10 seconds and 30 seconds. Figures 3.9a and 3.9b show the improvements in accuracy over time, respectively, with character-level and syllable-level AEBMT when employing different heuristics. We found that the performance of the combined AEBMT+SMT system does not vary significantly when allowing longer runtime. However, some exceptions were observed with H2 when allowing 10 seconds and 30 seconds of runtime, respectively, in the character-level and syllable-level experiments. Figures 3.9c and 3.9d show the effect of runtime on the performance of the combined AEBMT+SMT system.

It is interesting to note that the use of heuristics improves the performance of the analogy-based MT for NE transliteration with the exception of H1 and H4 heuristics. This is because some of the valid analogies are filtered out by the risky strategy of heuristics which discount some $\langle A, B \rangle$ pairs due to the significant difference between their length, as in example (31).

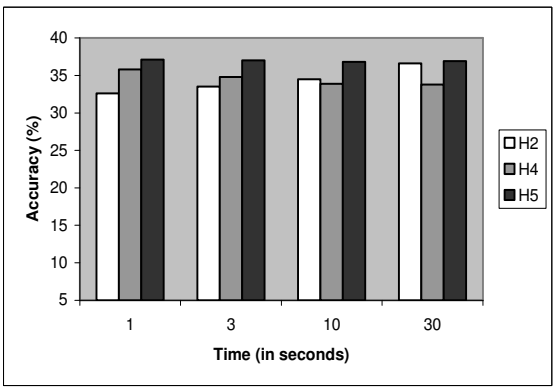


(a) Character-level AEBMT

(b) Syllable-level AEBMT



(c) Character-level AEBMT+SMT



(d) Syllable-level AEBMT+SMT

Figure 3.9: The effect of running time (1 sec, 3 sec, 10 sec and 30 sec) in analogy-based EBMT (AEBMT) and in the combined $\text{EBMT}_{\text{TM}} + \text{SMT}$ system with different heuristics and models.

(31) a. *He dived.* [9 characters]

b. *He dived into the river.* [24 characters]

A combination of AEBMT with SMT (AEBMT+SMT) for NE transliteration, where we are taking back-off for un-transliterated words from the transliteration procedure by SMT, gives an improvement of 13.2%, 2.5% and 236.8%, respectively for character, syllable and word level models compared to the baseline SMT. More precisely, we have seen improvement with AEBMT+SMT in the character-based model with all the heuristics compared to both AEBMT and SMT. However, the syllable-level model shows huge improvement (minimum of 51.9%) with AEBMT+SMT compared to AEBMT but only in two cases (no heuristic and H5) we have found

a small improvement (0.8% and 2.5%, respectively) over SMT although H2 and H3 have better accuracies than when no heuristics are applied in the analogy-based system. This is due to the fact that when a heuristic has better accuracy, in general it is solving more analogical equations. Thus it might be the case that while H2 and H3 are solving more analogical equations, it is producing an incorrect transliteration for some other words for which no back-off can be taken from SMT. However, the H5 heuristic overcomes the situation and shows improvements for all possible combinations.

Figure 3.10 gives a comparison of the total number of NEs transliterated, the number of NEs correctly transliterated irrespective of their rank in the output list and the number of NEs correctly transliterated at the first position. Although, H2 is much better in all aspects over no heuristics, the percentage of names correctly transliterated at top position out of the total NEs transliterated by H2 (30%) is much lower in comparison with no heuristics (42.5%). Thus we have seen in the combined system (AEBMT+SMT), no heuristic has little improvement compared to H2. However, the H5 heuristic overcomes the situation and shows improvement for all possible combinations. More interestingly, the word-based model reflects huge improvement (236.8%) with AEBMT+SMT compared to SMT but has no improvement over the AEBMT model. This signifies that whatever is correctly transliterated by SMT is a subset of the correct transliteration of the AEBMT system.

Regarding our English-to-Chinese experiments, we have seen similar trends as observed in the NE transliteration experiments. We see from Table 3.5 that AEBMT has much lower accuracy on its own compared to the baseline SMT accuracies. It has also been observed that without heuristics (no heuristic), the AEBMT system almost failed to translate any sentence. This is due the fact that within an allowable runtime the AEBMT system is unable to construct a valid analogical equation on both the source and target sides to produce some candidate solution. However, the use of H2 and H5 heuristics improves the translation accuracy compared to the use

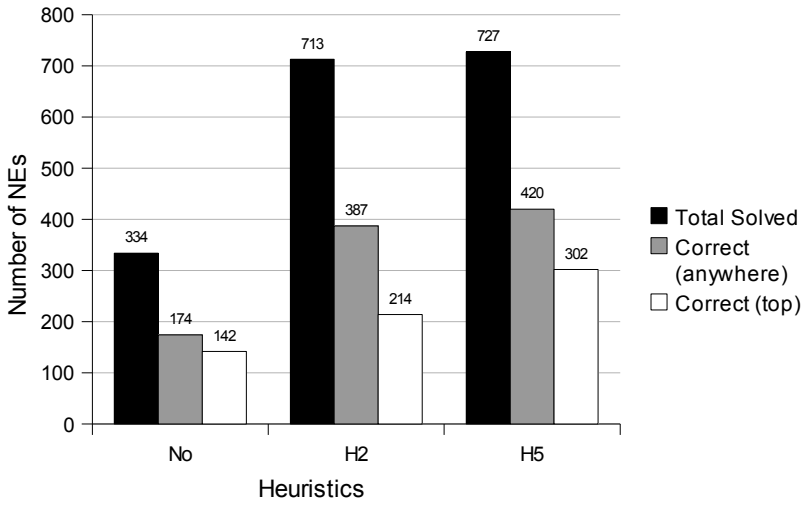


Figure 3.10: Comparison of the total number of NEs transliterated, the total number of correct transliterations in the candidate output set and the correct number of transliterations at rank 1 for the no-heuristic setting, the H2 and H5 settings.

of other heuristics. Here also, we observe that the use of H5 has the best accuracy in the AEBMT system.

The combined (AEBMT+SMT) system shows little improvement over the baseline SMT system for both H2 and H5 heuristics. This essentially reflects that there exist certain sentences which are better translated by the AEBMT system compared to the SMT system resulting in an overall improvement in the BLEU score. In contrast to the NE transliteration experiment, the use of H2 has better accuracy compared to the use of H5 in the AEBMT+SMT system. We have seen in the AEBMT system, the use of H5 translates 179 English sentences when the use of H2 translates 159 sentences, both within 1 second. Thus, the use of H5 has better accuracy on its own as it translates more sentences than H2. However, some of these translations might have a lower score compared to the SMT output resulting in a lower BLEU score for H5 compared to H2 in the combined system. Similar trends have been observed with an allowable runtime of 3 seconds in the AEBMT+SMT system.

Assesment of Error Types

The most common type of error encountered by the AEBMT model is that the correct output is often produced but not always in the first position. We have seen such examples for NE transliteration in Table 3.3 for the input NE *pakur* where the third most frequent output is the correct transliteration. The above phenomenon affects the accuracy of the AEBMT models. As we have seen in Figure 3.10, for H5, only 30.2% of NEs are correctly transliterated with the highest frequency in the output list although a total of 42% of NEs are transliterated correctly irrespective of their position in the output list. A similar trend has been observed during full text translation. Figure 3.11 shows an example of an erroneous translation from Chinese-to-English. The most frequent translation produced by the system is erroneous while the other translations are meaningfully correct, and particularly the third translation exactly matches the reference. The bottom row of the figure shows the set of analogical equations that produces the erroneous and the exact solutions.

The second type of error is spelling variations in the reference data in particular for NE transliteration. There are many cases where the NEs in the target language can be spelled in different ways. For example, the English input NE *edinburgh* can be written as 'एडीनबर्ग'(/edInabarga/) or 'एडिनबर्ग'(/eDinbarga/) in Hindi. The *matra*[7] 'ि'(/i/) becomes 'ी'(/I/). With our system, we are able to produce the latter 'एडिनबर्ग', but the reference translation has 'एडीनबर्ग', thus resulting in an incorrect transliteration. We found 46 (4.6%) such cases where the output differs from the reference due to spelling variation. Capturing these spelling variations could have increased the output accuracy by 4.6% for this particular data set.

Finally, we have seen cases where there is a tie in the top frequency of the output list. We choose one randomly in such cases. The effect is shown in Table 3.9 for the NEs *pratima* and *bhutti*. In the case of *pratima* the correct output as per the reference data is 'प्रतिमा'(/pratimA/) although all the three outputs have the same

[7]Matras are symbols for vowels used when consonants and vowels occur together in Indian languages.

Input	我 的 座位 在 哪里
Reference	Where is my seat ?
SMT o/p	where 's my seat ?
AEBMT o/p with frequency	*where 's mery a telene i cas useat ? (33)* *could you tell me where my seat is ? (2)* *where 's the my seat ? (1)* *where is my seat ? (1)* *where is my seat from here ? (1)*
Analogy Solved	Source Analogy: (*A* : *B* :: *input* : *D*) 电话 在 哪里 ? : 电话 在 哪 ? :: ***我 的 座位 在 哪里 ?*** : 我 的 座位 在 哪 ? Target Analogy: (*A´*: *B´*:: *output* : *D´*) where is there a telephone i can use ? : where is the phone ? :: ***where 's mery a telene i cas useat ?*** : where 's my seat ? Source Analogy: (*A* : *B* :: *input* : *D*) 我 的座位 在 哪 ? : 最 近 的 邮局 在 哪 ? :: ***我 的 座位 在 哪 里 ?*** : 最 近 的 邮局 在 哪里 ? Target Analogy: : (*A´*: *B´*:: *output* : *D´*) where 's my seat ? : where 's the nearest post office ? :: ***where is my seat ?*** : where is the nearest post office ?

Figure 3.11: Erroneous Chinese-to-English translation at rank 1.

frequency of 1. On the other hand, in case of *bhutti*, there are two outputs which have the same frequency of 6 and 'भुट्टी'(/bhuTTI/) is the correct output based on the reference data. However, the top two outputs for *bhutti* are again a change of spelling variation. In such cases, we randomly select one from the top frequencies.

Table 3.9: Example of transliteration with a tie in the highest frequency output.

Input NE	Output Transliterations
pratima	प्रतीमा (1), प्रतिमा (1), प्रतिमै (1)
bhutti	भुट्टि (6), भुट्टी (6), भट्ुटी (2), भट्टई (2)

3.4 Summary

From a very promising start, as reported in Lepage and Denoual (2005a,b,c), some of the drawbacks of the proportional analogy approach have since come to light. Unlike,

other approaches to EBMT, the approach seems to suffer badly when the size of the example-base increased, with both processing times and the number of solutions increasing. It is clear that heuristics must be introduced to reduce the search space, both in identifying likely example pairs $\langle A, B \rangle$, and preventing fruitless attempts to solve equations. Even where equations are solvable, the solution produced may be in need of filtering. While the approach is fraught with difficulties as a stand-alone translation model, its uses for the special case of unknown words, particularly named entities or specialist terms, seems much more promising.

We have not addressed some of the issues that have been explored in the literature while experimenting with PA-based system. Below we discuss the reasons for not exploring two of these issues in this chapter.

1. **Recursive solution:** In step two of the analogy-based EBMT in Section 3.1.2, we look for the translation of C (i.e. C') to form an analogical equation in the target language. However, it might be the case, that our example-base does not have the translation of C. In this circumstance, according to Lepage and Denoual (2005a,b,c), the translation of C needed to be solved recursively. This recursion is briefly discussed in Lepage and Denoual (2005a,c). However, no suggestion has been made to control the recursion so as to prevent the system from selecting the same $\langle A, B \rangle$ pair as an initial candidate and thereby the system gets stuck in a loop. Interestingly, the latter implementation by the same authors (Lepage and Lardilleux, 2007) does not use the recursion stage. The PA-based system works under a time-bound estimation. Thus, instead of using recursion we might try a totally new pair for producing a fruitful C. In this way, we might try more possible candidate pairs $\langle A, B \rangle$ from the example-base within the allowable runtime rather than trying to solve one pair recursively. Keeping this in mind, we have discarded the recursive solution for the translation of any analogical equation.

2. **Data Sparsity:** Our particular work tries to solve analogies by considering

examples as strings of characters. This is apparently an over generalization that might produce some incorrect solution that needs to be filtered. We have seen such instances in the first solution in Figure 3.11. One probable solution to this is to consider each sentence as a string of words to avoid such over generalized candidate solutions. Thus, the analogy will consider words as the smallest unit of a string instead of characters. This leads to the problem of data sparsity. We have already seen that a PA-based system has very low recall. Thus, considering a word-level might produce even lower recall for the PA-based system. Also, we used a PA-based approach for translation using a limited example-base. Data sparsity will be a huge problem if solving analogies and considering words as the smallest units. Hence, we have not experimented with words as the smallest units for analogical equations.

In particular to our experiment, although the PA-based system performs badly with English-to-Bangla MT, we found some improvement with the AEBMT+SMT system over the baseline SMT system for NE transliteration and English-to-Chinese MT tasks. Hence, this approach is unable to find a comprehensive answer to research question RQ1 that focuses on finding EBMT approaches for building better quality MT systems compared to a purely SMT-based system using limited resources. However, the approach shows improvement by combining EBMT systems with state-of-the-art phrase-based SMT systems for two different tasks. This partially answers research question RQ3.

3.4.1 Contributions

The main contribution of this chapter are summarized below:

- We developed the AEBMT system from scratch as we had no access to any open-source PA-based system.
- We developed heuristic (H5) which performs better compared to the other heuristics in the literature.

- We compared all the proposed heuristics under the same experimental set-up to understand their effectiveness.

- We showed that combining AEBMT with SMT is successful for named-entity transliteration and English-to-Chinese MT using IWSLT09 data.

In the next chapter, we will describe our work using a compiled approach to EBMT that can overcome the difficulties (both computation time and low recall) of PA-based technique. The approach precomputes generalized translation templates from example-base which can be further used to translate novel sentences in runtime.

Chapter 4

EBMT using Templates

In the previous chapter, we described EBMT using proportional analogy. We observed that analogy-based EBMT works well for shorter examples (especially with named entities) with small amounts of training data. However, analogy-based EBMT suffers from low recall when translating relatively longer examples. A run-time EBMT approach essentially has this difficulty due to time-bounded solutions which restrict analogy-based EBMT to attempting to solve all possible analogies that can be constructed from an example-base. In contrast, other approaches to EBMT learn rules that can be extracted beforehand from an example-base. EBMT using templates is a flexible method of learning translation templates from an example-base that can overcome the time-bounded solution of analogy-based EBMT. Under this template-based EBMT approach, different translation templates can be incorporated, which cannot be accomplished naturally in an analogy-based approach.

In this chapter, we present our work on a compiled approach to EBMT which essentially learns translation templates during the training stage, based on the description given in Güvenir and Cicekli (1998) and Cicekli and Güvenir (2001). We also present the use of probabilistic information to produce ranked output based on the learned translation templates. Finally, as we did for analogy-based EBMT, we present a combination of template-based EBMT with SMT to improve the performance of the overall system.

The organization of the chapter is as follows: We will first describe the definition of different translation templates, followed by our particular approach to EBMT using translation templates. We then present the experimental setup, the data, and the results followed by our observations from various experiments.

4.1 Translation Templates

A translation template is a generalized translation example pair, where some components (e.g. words, stems, morphemes etc.) are replaced with variables to infer commonality from specific cases. This generalization is done on both the source and target language for a pair of examples. Consider the following two source and target English–Turkish example pairs in (32) from the BTEC corpus.

(32) a. *i have a sharp pain* ⇔ *keskin bir ağrım var*

b. *i have a dull pain* ⇔ *hafif bir ağrım var*

Clearly, the English side of the above two examples share the word sequences *i have a* and *pain* and differ in words *sharp* and *dull*. Similarly on the target-side, the similar part is *bir ağrım var* and differing parts are *keskin* and *hafif*. Based on this observation, the examples in (32) can be generalized as shown in (33).

(33) *i have a* (sharp|dull) *pain* ⇔ (keskin|hafif) *bir ağrım var*

The generalization in (33) represents the source side as *i have a* $(w_1|w_2)$ *pain*, where $(w_1|w_2)$ denotes either the word w_1 or w_2. Similarly, the target-side is generalized into $(t_{w_1}|t_{w_2})$ *bir ağrım var*, where t_{w_i} denotes the translation of the word w_i. The example in (33) can be further universalized in (34) by introducing a single variable that can take any word instead of the set of fixed words.

(34) *i have a* X^S *pain* ⇔ X^T *bir ağrım var*

The variable X^S can range from a single word to a subsentential word sequence. X^T is the translation equivalent of the source segment X^S.

The above example shows a generalization based on the similarity of an example pair. This essentially learns a translation template (34) from the example pairs which can further be used to translate novel sentences with a similar structure. Güvenir and Cicekli (1998) showed that the translation template can be learned automatically from the examples based on similarities and differences within the example pairs. They called these *similarity translation templates* and *difference translation templates*, respectively.

4.1.1 Similarity Translation Templates

The similarity translation templates are learned based on correspondences between two example pairs. These similarities are identified in both source and target language sides of the example pairs and can be of a different granularity. These may include information from the surface word, morphemes, or the syntactic category of the word. Sometimes some semantic information is also used to find a similarity between example pairs to produce translation templates. Consider the pair of sentences in (35) of an English–Turkish example taken from Cicekli and Güvenir (2001).

(35) a. *I will drink orange juice* ⇔ *portakal suyu içeceğim*

b. *I will drink coffee* ⇔ *kahve içeceğim*

In the above example, the similar part in the source (English) side is *I will drink* and the similar part in the target side is *içeceğim*. The remaining dissimilar parts in the source side are *orange juice* and *coffee*. Similarly, the dissimilar parts in the target (Turkish) side are *portakal suyu* and *kahve*. Thus the following subsentential alignments in (36) can be captured from example (35).

(36) a. *I will drink* ⇔ *içeceğim*

b. *coffee* ⇔ *kahve*

c. *orange juice* ⇔ *portakal suyu*

A similarity translation template keeps the similar part and generalizes the differing parts with variables in both source and target side of the example pairs. Example (37) represents the similarity translation template for the example in (35).

(37) *I will drink* $X^S \Leftrightarrow X^T$ *içeceğim*

The subsentential aligned pairs learned in (36) are called *atomic translation templates*. These atomic translation templates do not contain any variable to instantiate during the decoding process. Only one translation template can be produced based on the similarity of two example pairs. Both similarity translation templates and atomic translation templates are used to translate novel sentences.

4.1.2 Difference Translation Template

Translation templates can also be learned from a pair of examples by keeping the differing parts from the example pairs and generalizing over the similar parts. These translation templates are known as difference translation templates. The two difference translation templates from example (35) are shown in (38) where the similar parts (i.e. *I will drink* and *içeceğim*) in each example with variables.

(38) a. X^S *coffee* $\Leftrightarrow$ *kahve* X^T

b. X^S *orange juice* $\Leftrightarrow$ *portakal suyu* X^T

Unlike similarity translation templates, two translation templates can be produced from an example pair when considering the difference. Thus, a total of six translation templates can be produced from the example pairs in (35). These include the three atomic translation templates in (36), one similarity translation template in (37) and two difference translation templates in (38).

Cicekli and Güvenir (2001) proposed an approach to generalize over sequences of words. The underlying assumption is that given two parallel sentence pairs, translation templates can be learned based on the similarities in both the source and target sides. The same applies to the differing parts between two parallel sentences. Generalization in this approach consists of replacing the similar or differing sequences

with variables and producing a set of translation templates (including *atomic translation templates* containing no variables as in (36)). These translation templates are further used to assist in translating new input sentences.

Translation templates essentially reduce the data-sparseness problem by generalizing some of the word sequences.[1] Gough and Way (2004) demonstrated that a set of automatically derived generalized templates can improve both coverage and translation quality. Thus, the approach is anticipated to answer research questions RQ1 (focuses on exploiting EBMT approaches in resource-poor settings) and research question RQ3 (concentrates on effective combination of EBMT and SMT to handle data-sparsity problem). This motivates us to use this approach to overcome the data-sparsity problem of phrase-based SMT.

4.2 Our Approach

Translation templates are used to extend the example-base in order to reduce data-sparseness. This suggests that translation accuracy can improve with a training set of fewer examples if templates are used in addition to the surface-level source–target sentence/phrase equivalents. With this in mind, we have developed our generalized translation-template-based EBMT system based on the description given in Cicekli and Güvenir (2001).

Like Cicekli and Güvenir (2001), we have developed two separate components within our approach, namely *learning* and *decoding*. The learning component first infers translation templates from the example-base. The decoding component translates new sentences using the translation rules produced in the learning phase. In addition to the work done by Cicekli and Güvenir (2001), we introduce the concept of translation scores to rank the output during decoding based on the probabilities of the learned translation templates.

[1]It is worth noting that similar experiments can be conducted in a hierarchical phrase-based SMT framework (Chiang et al., 2005).

4.2.1 Learning Translation Templates

The learning algorithm infers translation templates based on the similarities and differences between two example pairs (e_1, e_2) from a bilingual example-base. Each translation pair contains the source(S)-target(T) translation equivalences. Formally, $e_i : s_i \Leftrightarrow t_i$, where $s_i \in S$ and $t_i \in T$.

We find the similarities and differences between example pairs in the surface-level words. The similarity between two examples of a language refers to the non-empty sequence of common words in both sentences. The difference between two examples of a language refers to a pair of sequences (d_1, d_2), where d_1 and d_2 are subsequences in the first and second example, respectively, and d_1 and d_2 do not contain any common item.

Based on the similarities and differences between two example pairs (e_1, e_2), we first estimate the *match sequence* ($M_{1,2}$), as shown in (4.1).

$$M_{1,2} : sim_0^S, d_0^S, sim_1^S, ..., d_{n-1}^S, sim_n^S \Leftrightarrow sim_0^T, d_0^T, sim_1^T, ..., d_{m-1}^T, sim_m^T \qquad (4.1)$$

sim_i^S and sim_i^T refers to the similarity between two examples, respectively, in source and target language. Correspondingly, d_i^S and d_i^T denotes a difference pairing between two examples in the source and target language, respectively. In order to learn translation templates, one similarity on each side of the match sequence must be non-empty. In addition, there must be a difference sequence between two non-empty similarity sequences. However, sim_0^S, sim_n^S, sim_0^T or sim_m^T can be empty. The detailed formulation of the match sequence can be found in Cicekli and Güvenir (2001). However, none of their papers describe the algorithm to produce the match sequence.

For this reason we developed our own algorithm for finding a match sequence between two example pairs (e_1, e_2). We used an edit-distance trace algorithm to find the match sequence. Our approach is shown in Algorithm 1.

Given two sentences of one language the algorithm find the similarities and dis-

Algorithm 1 sequence(ED, s_1, s_2, row, col)

In: Edit-distance matrix between s_1 and s_2 ED,
first example s_1,
second example s_2,
row = length of s_1
col = length of s_2
Out: Similarity and difference sequence seq

while i>0 and j>0 **do**
  i=row; j=col
  if $ED_{i,j} = ED_{i-1,j-1}$ and $s_1[i] = s_2[j]$ **then**
    $seq = s_1[i].seq$
    sequence($ED, s_1, s_2, row-1, col-1$)
  else if $ED_{i,j} = ED_{i-1,j-1} + 1$ and $s_1[i] \neq s_2[j]$ **then**
    $seq = (s_1[i]|s_2[j]).seq$
    sequence($ED, s_1, s_2, row-1, col-1$)
  else if $ED_{i,j} = ED_{i-1,j} + 1$ **then**
    $seq = (s_1[i]|-).seq$
    sequence($ED, s_1, s_2, row-1, col$)
  else
    $seq = (-|s_2[j]).seq$
    sequence($ED, s_1, s_2, row, col-1$)
  end if
end while
if $i > 0$ **then**
  $seq = (s_1[1...i]|-).seq$
end if
if $j > 0$ **then**
  $seq = (-|s_2[1...j]).seq$
end if

similarities at the level of surface words. For example, consider two examples of a language, $s_1 = w_1^s\, w_2^s\, w_3^s\, w_4^s\, w_5^s$ and $s_2 = w_1^s\, w_2^s\, w_3^s\, w_6^s$. These examples essentially represent the source-side sentences of example (35, p.79):

(39) a. s_1 = I(w_1^s) will(w_2^s) drink(w_3^s) orange(w_4^s) juice(w_5^s)

b. s_2 = I(w_1^s) will(w_2^s) drink(w_3^s) coffee(w_6^s)

The first three words are common between the two examples and the last two words of the first example are different from the last word of the second example. Figure 4.1 shows the matching between s_1 and s_2 produced by Algorithm 1.

After obtaining the sequence in Figure 4.1, we produce the match sequence by concatenating adjacent similarity and difference sequences. This produces the source-side match sequence $M_{1,2}^s = w_1^s\, w_2^s\, w_3^s(w_4^s\, w_5^s|w_6^s)$ between s_1 and s_2, where

$$
\begin{array}{lcccccc}
s_1 & = & w_1^s & w_2^s & w_3^s & w_4^s & w_5^s \\
 & & | & | & | & & \\
s_2 & = & w_1^s & w_2^s & w_3^s & - & w_6^s
\end{array}
$$

Figure 4.1: Example of matching based on edit-distance trace.

$sim_0^S = w_1^s\, w_2^s\, w_3^s$ and the difference pair (d_0^S) from the two examples is $(w_4^s\, w_5^s | w_6^s)$. Correspondingly, based on example (35), the target-side match sequence between $t_1 = w_1^t w_2^t w_3^t$ and $t_2 = w_4^t w_3^t$ is $M_{1,2}^t = (w_1^t w_2^t | w_4^t) w_3^t$. The overall match sequence between the example pairs is shown in Equation (4.2).

$$M_{1,2} : w_1^s\, w_2^s\, w_3^s\, (w_4^s\, w_5^s | w_6^s) \Leftrightarrow (w_1^t\, w_2^t | w_4^t)\, w_3^t \tag{4.2}$$

Based on Equation (4.2), the example pairs in (35) can be represented as (4.3):

$$M_{1,2} : I\ will\ drink\ (orange\ juice | coffee) \Leftrightarrow (portakal\ suyu | kahve)\ \text{içeceğim} \tag{4.3}$$

Inferring Similarity Templates

We adopt the algorithm described in Cicekli and Güvenir (2001, p.62) to infer the similarity translation templates from the match sequence. The outline of the algorithm is as follows:

(i) If the match sequence contains one different item on both source and target side then these differing items are a translation of each other. For example, consider the match sequence $sim_0^S, d_0^S, sim_1^S \Leftrightarrow sim_0^T, d_0^T, sim_1^T$, which has a single difference between the source and target language. Then d_0^S is a translation equivalent of d_0^T. A similarity translation template (as in (4.4)) is inferred by replacing the difference sequences with variables.

$$sim_0^S\, X_0^S\, sim_1^S \Leftrightarrow sim_0^T\, X_0^T\, sim_1^T \tag{4.4}$$

The differing part in the source (d_i^S) and target sides (d_i^T) of the match sequence is a pair of substrings of the sentences used to form the match sequence. For example, in the match sequence (4.2), the differing parts on the source and target sides are ($w_4^s\, w_5^s, w_6^s$) and ($w_1^t\, w_2^t, w_4^t$), respectively. This can be formally expressed in (4.5).

$$\begin{aligned} d_0^S &\equiv (d_{0,1}^S, d_{0,2}^S) \\ d_0^T &\equiv (d_{0,1}^T, d_{0,2}^T) \end{aligned} \tag{4.5}$$

where, $d_{i,j}^S$ and $d_{i,j}^T$ are the j-th component of the i-th differing element in the source and target sides, respectively. These constituents are a translation of each other. The atomic translations in (4.6) are inferred from these differing constituents.

$$\begin{aligned} d_{0,1}^S &\Leftrightarrow d_{0,1}^T \\ d_{0,2}^S &\Leftrightarrow d_{0,2}^T \end{aligned} \tag{4.6}$$

(ii) If there are an equal number (n) of differing subsequences on both sides of the match sequence, but greater than one, then prior knowledge is used to infer the translation templates. Previously learned templates are used as prior knowledge for learning new templates. If $(n-1)$ of these differing sequences are observed previously, then a new similarity template is inferred replacing the unobserved difference sequences with variables. For example, if we have a match sequence $sim_0^S, d_0^S, sim_1^S, d_1^S, sim_2^S, d_2^S \Leftrightarrow sim_0^T, d_0^T, sim_1^T, d_1^T, sim_2^T, d_2^T$, and if we have observed that $d_0^S \equiv d_0^T$ and $d_2^S \equiv d_2^T$, then we can infer the similarity translation template in (4.7).

$$sim_0^S\, X_0^S\, sim_1^S\, X_1^S\, sim_2^S\, X_2^S \Leftrightarrow sim_0^T\, X_0^T\, sim_1^T\, X_1^T\, sim_2^T\, X_2^T \tag{4.7}$$

Each differing sequence is replaced by a corresponding variable. This also infers

new atomic translation templates ($d^S_{1,1} \Leftrightarrow d^T_{1,1}$ and $d^S_{1,2} \Leftrightarrow d^T_{1,2}$) based on the translation equivalent for the previously unmatched difference sequences (d^S_1 and d^T_1).

The process is performed iteratively until no new translation templates are found.

Inferring Difference Templates

Difference translation templates are learned in a similar way to the learning similarity translation templates. Here also, we use the approach described by Cicekli and Güvenir (2001, p.64). The outline of the algorithm is as follows:

(i) If there exists only one single non-empty similarity on both the source and target side of the match sequence, then these similar constituents are the translation of each other. Translation templates are inferred by replacing the similar parts with variables and keeping the different parts as new translation templates. For example, considering the match sequence in Equation (4.2), we can replace the similar source- and target-side parts ($w^s_1\, w^s_2\, w^s_3$ and w^t_3) with the variables X^S_0 and X^T_0 to infer two translation templates in (4.8).

$$\begin{aligned} X^S_0\, w^s_4\, w^s_5 &\Leftrightarrow w^t_1\, w^t_2\, X^T_0 \\ X^S_0\, w^s_6 &\Leftrightarrow w^t_4\, X^T_0 \end{aligned} \tag{4.8}$$

The similar part in source and target side is a translation equivalent and produces the atomic translation template $w^s_1\, w^s_2\, w^s_3 \Leftrightarrow w^t_3$.

(ii) If both the source and target part of the match sequence have an equal number ($n \geq 1$) of similarity sequences and ($n-1$) of them have already been observed from previously learned templates, then we can infer difference translation templates. Consider two similarity sequences (sim^S_0, sim^T_0) and (sim^S_1, sim^T_1), which have already been checked in the match sequence $sim^S_0, d^S_0, sim^S_1, d^S_1, sim^S_2 \Leftrightarrow sim^T_0, d^T_0, sim^T_1, d^T_1, sim^T_2$. The remaining unchecked

similarity sequence is (sim_2^S, sim_2^T). Thus, two translation templates are learned (as in (4.9)), replacing the unchecked similarity sequences with variables and keeping the differing parts (($d_{0,1}^S, d_{0,2}^S$) and ($d_{0,1}^T, d_{0,2}^T$)) in both source and target sides.

$$\begin{aligned} X_0^S\, d_{0,1}^S\, X_1^S\, d_{1,1}^S\, X_2^S &\Leftrightarrow X_0^T\, d_{0,1}^T\, X_1^T\, d_{1,1}^T\, X_2^T \\ X_0^S\, d_{0,2}^S\, X_1^S\, d_{1,2}^S\, X_2^S &\Leftrightarrow X_0^T\, d_{0,2}^T\, X_1^T\, d_{1,2}^T\, X_2^T \end{aligned} \tag{4.9}$$

New atomic translation templates are inferred based on the unchecked similarity sequence in both the source and target language. This single unchecked similarity sequence is a translation equivalent in two languages. The atomic translation template learned for the aforementioned match sequence is $sim_2^S \equiv sim_2^T$.

The difference translation template learning process is also performed in an iterative way until no new template is learned in a particular iteration. Applying both template learning processes to the example in (35) we obtain the translation templates in examples (36, p.79), (37, p.80) and (38, p.80).

In our approach, we enhance the existing algorithm by assigning a probability to each translation template (including the atomic translation templates). This probability is essential in helping us to produce a translation score during the decoding phase. After learning the templates we assign a probabilistic score to each translation template ($T_i : s_i \rightarrow t_i$) using the counts in ($i$) as in Equation (4.10):

$$p_i(t_i|s_i) = \frac{count(s_i \rightarrow t_i)}{count(s_i)} \tag{4.10}$$

Thus our translation templates are in the form of $(T_i : s_i \rightarrow t_i)\ :: p_i$.

Time Complexity of the Learning Process

The template learning algorithm works iteratively. In each iteration, we examine all possible example pairs to estimate a match sequence. The number of possible

example pairs is n^2 ($\sum_{i=1}^{n} i$), where n is the number of sentences in the example-base. For each example pair we need to find the match sequence. Finding a match sequence involves computing an edit-distance matrix for both the source and target language of an example pair. In general, edit-distance computation between strings of characters has quadratic time complexity with respect to the length of the sentence (Wagner and Fischer, 1974). Thus, the time complexity of finding a match sequence from an example pair is $O(m^2)$, where m is the average length of the examples in words. Accordingly, the time complexity for finding a match sequence is $O(n^2m^2)$. After obtaining the match sequences, we obtain the translation templates based on the similarities to and differences from the match sequence. Theoretically, the maximum number of possible iterations is $(n-2)$. Thus, the learning template has a worst case time complexity of $O(n^3m^2)$. The average sentence length of a corpus does not vary significantly with the size of the corpus, and the value of m does not increase with the size of the training data. Therefore, in practice the run time complexity of the learning algorithm is $O(n^3)$.

4.2.2 Translation Using Templates

In the decoding phase, the translation templates learned in the previous section are used directly to translate new sentences. The recursive decoding algorithm described in Cicekli and Güvenir (2001, p.71) produces multiple translations, one for each translation template matching the input sentence. In our approach, we enhance the existing algorithm by supplying an associated translation score (q) with every output produced. Figure 4.2 represents the block diagram of our decoding process.

The template matching procedure returns all possible templates that match with an untranslated segment of a sentence. Each matched translation template rule is then applied to the input and an associated translation score is computed. The translation score is computed based on the probability of the applied translation template (p) and the similarity of the translation template with the input (w). After applying all possible translation templates to the input, the fully translated

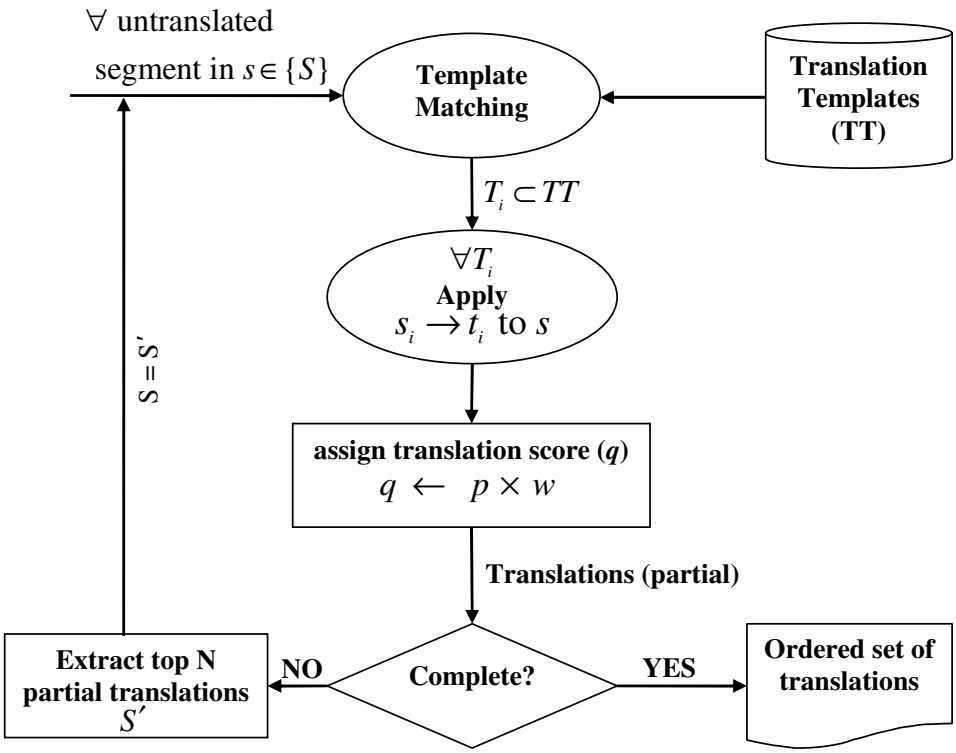


Figure 4.2: Decoding architecture.

sentences are placed in an ordered set of output translations. A fixed set of remaining partially translated sentences are iteratively translated until the partially translated set is empty or no further translation template can be applied. Initially, for a new test sentence, the whole sentence is one untranslated segment. There may be multiple untranslated segments depending on the number of variables in the translation template that has been applied to the input sentence. The decoding process is applied to all untranslated segments sequentially. Our particular decoding algorithm is based on beam search which potentially reduces the computation of the search. At each stage we consider a fixed set of partial translations (N) and repeat the process. The number N is essentially the size of the beam. The detail of our approach is shown in Algorithm 2.

Lines 5 and 6 of the algorithm estimate the set of translation templates that match an untranslated segment. In line 5, *untranslated*(Y) returns the set of un-

Algorithm 2 decoding(X, T, M, N)

In: source sentence X,
translation template set T,
number of top-ranking translations M,
number of live hypotheses during decoding N
Out: set of the M best translations and their score H

```
1: {S is the set of partial translations containing pairs (partial translation, score)}
2: S ← (X, 1) {initialise set to contain the source sentence and score=1}
3: repeat
4:    S' ← ∅
5:    for all (Y, q) ∈ S and Z ∈ untranslated(Y) do
6:       for all s → t ∈ T such that s matches Z do
7:          Y' = substitute(Y, Z, s → t)
8:          w = numSurfaceWords(s)/length(Y)
9:          q ← q × p × w
10:         S' ← S' ∪ {(Y', q')}
11:      end for
12:   end for
13:   for all (Y', q') ∈ S' do
14:      if untranslated(Y') ≠ ∅ then
15:         S' ← S' − (Y', q') {remove complete translation for further processing}
16:         H ← top(H ∪ {(Y', q')}, M) {add to ordered set of completed translations H}
17:      end if
18:      S ← top(S', N) {only N partial translations are considered in the next iteration}
19:   end for
20: until S ≠ ∅
```

translated substrings in the partial translation Y. In line 7, the function *substitute*(Y,Z,$s \rightarrow t$) generates a new partial translation Y', where untranslated segment Z is instantiated by $s \rightarrow t$ so that those parts of Z matched by variables in s are copied to the positions of their corresponding variables in t. The similarity weight factor (w) of each translation template ($T_i : s_i \rightarrow t_i$) is computed during the runtime decoding process, as in line 8 of the algorithm. The factor represents the ratio of the surface words in the source part of the translation template (s_i) to the length of the untranslated segment (Y). The two factors (p and w) are multiplied to assign a translation score (q) to each output translation in the potential translation set (line 9 of the algorithm). After applying the translation rules, we remove the fully translated output (in line 15) and add it to the ordered set of completed translations (H) in line 16 of the algorithm. In line 18, we extract the top N partial translations and repeat the process.

4.3 Experiments and Results

We evaluated our translation-template-based EBMT system on two MT tasks. In our first task, we chose English-to-Bangla translation using medical receptionist corpus described in Section 2.5, which did not show any promising result with a PA-based system in the previous chapter. In the second task, we tested our template-based EBMT system for translating English into Turkish. We chose English–Turkish data from the IWSLT09 shared task. Both the English–Bangla medical receptionist corpus and the IWSLT09 corpus comprised shorter sentences that are from a single domain. The average length of the source and target sentences (in words) in the English–Bangla corpus are 8.5 and 8.3, respectively. The average length of the source and target sentences in the IWSLT 09 corpus are 9.5 and 7.0 words, respectively. We chose these two corpora due to their homogeneity as both training and test examples from these small closed-domain corpora have a significant similarity in surface words. The translation templates are extracted based on the similarities and differences in the surface form of a pair of examples. Thus, it was anticipated that a significant amount of translation templates can be inferred from these data sets as they share a lot of words between example sentences.

4.3.1 Experiments Conducted

As with the PA-based system in the previous chapter, we conducted three different experiments for both language pairs to evaluate our translation-template-based EBMT system.

- **SMT:** Our first experiment is to estimate the baseline accuracy of the translation task. We estimate this baseline accuracy using an SMT system. We use OpenMaTrEx (Dandapat et al., 2010a), an open-source **SMT** system as the baseline and compare the results with our approach.

- **GEBMT:** We conduct our second experiment using our template-based EBMT system. The experiment was based on the translation templates learned and

the decoder presented in Sections 4.2.1 and 4.2.2. We shall refer to this system as **GEBMT** (generalized translation-template-based EBMT). In this experiment, out of many possible translations produced by the GEBMT system, we chose the best candidate during evaluation. The best candidate is selected using the translation score (q) produced by the GEBMT system.

- **GEBMT+SMT:** In the third experiment, we combine the SMT system with the GEBMT system to improve the translation score. We found that there are cases where the GEBMT system produces correct output but the SMT system fails and vice-versa. In order to further improve the translation accuracy, we used a combination of GEBMT and SMT. We use the translation score (q) to combine GEBMT with SMT. We assume the translations of an input sentence s produced by GEBMT and SMT are $\mathrm{T}_{\mathrm{GEBMT}}(s)$ and $\mathrm{T}_{\mathrm{SMT}}(s)$, respectively. We also have the translation score (q) for each output produced by the GEBMT system. During combination, we rely on the GEBMT system if the value of the translation score (q) is greater than a particular threshold. If the value of q is greater than a particular threshold we rely on the output $T_{GEBMT}(s)$; otherwise we take the output from $\mathrm{T}_{\mathrm{SMT}}(s)$. We conducted experiments with the threshold for q varying from 0.3 to 0.9 (threshold range was empirically selected) to see the relative effect. We shall refer to this system as $\mathbf{GEBMT_{score>x} + SMT}$ where x refers to the particular threshold used to rely on GEBMT output. This experiment will estimate the effect of using the GEBMT system for some sentences.

This gives three different experiments (SMT, GEBMT and $\mathrm{GEBMT}_{\mathrm{score>x}}$ + SMT) for two different translation tasks.

4.3.2 Data Used for Experiments

We use the same English–Bangla data (described in Section 3.3.2) in our systems for translating English into Bangla. The data consists of 380 parallel sentences

from medical receptionist dialogue exchange. The test data consists of 41 sentences disjoint from the training set.

To evaluate the English-to-Turkish system, we used IWSLT09 data. The IWSLT09 data consists of 19,972 sentences from the basic traveller expression corpus (BTEC). We used only two small sets (1242 sentences and 2484 sentences) of data to learn translation templates. Note that due to the large time complexity of the learning algorithm for inferring translation rules in the GEBMT approach, we conducted our GEBMT experiments with a smaller subset of the whole IWSLT09 English–Turkish training data. We used 414 sentences from the IWSLT09 development set as the test set of our English-to-Turkish experiment.

Note, the amount of data used to test this approach in the literature was also very small.[2] Güvenir and Cicekli (1998) and Cicekli and Güvenir (2001) used a small example base of 747 sentences to learn the translation templates. Öz and Cicekli (1998) reported learning translation templates using 488 examples. A maximum of 4,152 training examples were used for learning translation templates in Cicekli (2005). Similarly, we used a small set of training examples to see the effect of using GEBMT for translating homogeneous data using limited resource (relates to research question RQ1). Our main goal with this work is to ivestigate the effectiveness of the GEBMT technique (described by Cicekli and Güvenir (2001)) in a resource-poor setting and to explore the possibility of combining the same with SMT-based model.

Table 4.1 provides the number of translation template rules inferred from the example-base used in our experiment.

Table 4.1: Number of translation rules inferred using different data sets.

Data	Number of templates	
	with variables	atomic
English-to-Bangla (380 sentences)	1928	1070
English-to-Turkish (1242 sentences)	5777	4232
English-to-Turkish (2184 sentences)	15189	9636

[2]Brown (2001) used a simplified variant of the approach described by Cicekli and Güvenir (2001). The run complexity of each iteration in the induction step remains $O(n^2)$. However, the approach effectively used 20k sentence pairs (1.1 million tokens) for French-to-English translation.

Table 4.2: System accuracies obtained by different GEBMT models for English-to-Bangla MT. The subscript $score > x$ denotes the value of the translation score (q).

System	BLEU (in %)	NIST
Training Data: 380 sentences		
SMT	33.69	4.61
GEBMT	29.11	4.49
$\text{GEBMT}_{\text{score}\geq 0.3}$ + SMT	34.81	4.83

4.3.3 Results

We evaluated the resulting translation against the provided reference translation sets in terms of two automatic evaluation metrics - BLEU and NIST.

The results for English-to-Bangla translation for the three different experiments are presented in Table 4.2. The BLEU score obtained with the baseline SMT system for English-to-Bangla translation is 33.69%. The GEBMT system on its own has a lower BLEU score of 29.11% compared to baseline SMT. However, when combining the GEBMT system with SMT, the BLEU score achieved by the combined system($\text{GEBMT}_{\text{score}\geq 0.3}$ +SMT) is 34.81%. The combined system selected translations from the GEBMT system when the translation score was greater than or equal to 0.3. For English-to-Bangla translation, the combined system shows a relative improvement of 3.3% over the baseline SMT score when the value of the translation score (q) is greater than or equal to 0.3. This high BLEU score is due to the property of the English–Bangla test sentences (disjoint from training data) which are very similar sentences to the example-base.

We measure statistical significance to estimate the reliability of the improvements. Statistical significance tests were performed using paired-bootstrap resampling (Koehn, 2004).[3] The improvement of the combined system over the baseline SMT is statistically significant (with a reliability of 97%).

A similar trend has been observed with NIST scores. The GEBMT system has a lower NIST score when used on its own compared to the NIST score ob-

[3]http://www.ark.cs.cmu.edu/MT/

Table 4.3: System accuracies using different GEBMT models for English-to-Turkish MT. The subscript $score > x$ denotes the value of the translation score (q).

System	BLEU (in %)	NIST
Training Data: 1242 sentences		
SMT	7.63	2.89
GEBMT	6.80	2.78
$\text{GEBMT}_{\text{score}\geq 0.3} + \text{SMT}$	7.96	2.98
Training Data: 2484 sentences		
SMT	10.72	3.51
GEBMT	7.21	3.07
$\text{GEBMT}_{\text{score}\geq 0.9} + \text{SMT}$	10.83	3.52
$\text{GEBMT}_{\text{score}\geq 0.8} + \text{SMT}$	10.99	3.53
$\text{GEBMT}_{\text{score}\geq 0.7} + \text{SMT}$	10.76	3.53
$\text{GEBMT}_{\text{score}\geq 0.6} + \text{SMT}$	10.55	3.52

tained by the baseline SMT system — 4.49 and 4.61, respectively. The combined $\text{GEBMT}_{\text{score}\geq 0.3} + \text{SMT}$ system shows a 4.77% relative improvement compared to the individual systems.

Table 4.3 shows the resulting translation scores obtained by the three different systems for English-to-Turkish. Like the English-to-Bangla translation, we observe a similar trend in translation accuracy when translating English into Turkish. The first three rows in Table 4.3 show the translation score when 1242 sentences were used to infer the translation templates. The BLEU score obtained with the SMT and GEBMT systems are 7.63% and 6.80%, respectively. This shows that GEBMT has a lower accuracy compared to the baseline SMT system for English-to-Turkish translation. Combining the two systems shows improvement over the individual systems. The BLEU score obtained by the combined system is 7.96% with 1242 training sentences. This has been achieved when SMT output is augmented by the GEBMT output that has a translation score greater than or equal to 0.3. This shows a relative BLEU point improvement of 4.3% with the combined system when $q \geq 0.3$ compared to the baseline SMT system. However, this improvement is only significant for a small training data size (1242 sentences). Upon doubling the data (2484 sentences), no improvements were observed for $q < 0.7$. Under the circumstances, the highest improvement (2.5%) was achieved when $q \geq 0.8$. Turkish

Table 4.4: System accuracy obtained with different translation score parameters in the English-to-Turkish GEBMT system.

Parameter Used	Translation Score (q)	BLEU
Training Data: 2484 sentences		
w	$q = q \times w$	6.50
p	$q = q \times p$	6.29
p,w	$q = q \times p \times w$	7.21

is a morphologically very rich language. The use of small amounts of training data for such a morphologically rich language results in general low BLEU scores for all the above experiments.

None of the improvements in Table 4.3 are statistically significant. The reliability of the improvement is 88% when $q \geq 0.8$. Smaller values of q (< 0.7) show less reliable ($< 70\%$) improvement of the translation accuracy.

A similar effect was observed with the NIST evaluation of the English-to-Turkish translation. We found a relative NIST score improvement of 3.57% over the baseline SMT when $q \geq 0.3$ for small training data (1242 sentences). With the increased training data (2184 sentences), we found a relative NIST score improvement of 0.5% over the baseline SMT when $q \geq 0.7$. We found similar improvements using the two MT evaluation metrics for both English-to-Bangla and English-to-Turkish translations. These signify that the improvement using the GEBMT system has a similar effect on both the fluency and adequacy of the translations.

Additionally, we conducted an experiment to understand the effect of the two parameters (probability of a translation template p and similarity weight factor w) used to compute the translation score (q) during decoding (step 9 of Algorithm 2). We use these two parameters individually and together to compute the translation score (q) to estimate their effect in translation. Table 4.4 shows the accuracy obtained with different translation score factors for English-to-Turkish. The use of both p and w together improves the translation score compared to the individual uses of p and w.

4.3.4 Observations

We found that the GEBMT system works well for certain sentences when a small amount of homogeneous data is used to learn the translation templates. We have seen in Table 4.3 that the combined system has a 4.3% relative improvement compared to the baseline SMT when 1242 English–Turkish sentences are used to learn translation templates. Also, the combined system relies on GEBMT output where $q \geq 0.3$. In contrast, with 2484 English–Turkish training examples, the relative BLEU point improvement with the combined system over the baseline SMT is 2.5%, relying on a high value of q (≥ 0.8). This is due to the fact that a small corpus might generate appropriate translation templates if some similar examples exist in the example-base. In contrast, the SMT system may produce an incorrect solution due to a lack of evidence to estimate the probabilities.

A higher value of translation score (q) signifies that fewer sentences are translated using the GEBMT approach in the combined system. Table 4.5 shows the percentage of sentences selected using the GEBMT approach in the combined system. The table also shows the BLEU score comparison for the selected sentences for the SMT and GEBMT approach. It has been observed that with 2484 sentences only a small amount of test sentences (6.7% when $q \geq 0.8$) are translated using the GEBMT approach. The sentences that are translated using GEBMT are generally shorter sentences which result in a high BLEU score for those particular cases. The average length of the 40 sentences (when $q \geq 0.7$) translated using GEBMT is 5.3 words but the entire set of test sentences has the average length of 6.7 words.

Another significant observation is that GEBMT is often unable to translate all the words which are translated by SMT. In order to translate a word sequences using GEBMT, the surface level word needs to appear in a translation template (either in the templates with variables or in the atomic templates). Table 4.6 shows this effect. Three words (*have them reissued*) remain untranslated in the output produced by the GEBMT system. However, the word order of the GEBMT output better matches reference translations compared to the SMT output. In contrast,

Table 4.5: System accuracies obtained by different translation scores (q) in English-to-Turkish GEBMT system.

q	times/percentage GEBMT used	BLEU (%)	
		SMT	GEBMT
English-to-Bangla (Training: 380 sentences; Test: 41 sentences)			
≥ 0.3	13 (31.7%)	47.46	55.92
English-to-Turkish (Training: 1242 sentences; Test: 414 sentences)			
≥ 0.3	178 (42.9%)	11.07	13.17
English-to-Turkish (Training: 2484 sentences; Test: 414 sentences)			
≥ 0.9	16 (3.9%)	42.09	44.9
≥ 0.8	28 (6.7%)	32.08	39.67
≥ 0.7	40 (9.6%)	30.31	31.78
≥ 0.6	51 (12.3%)	29.32	28.67

with SMT, only one word (*reissued*) is untranslated. This essentially produces a better n-gram match for SMT against the reference translation. This results in a low BLEU score for GEBMT compared to SMT although the GEBMT translation looks more fluent (other than the untranslated sequence) compared to the SMT output.

Table 4.6: Example translation using GEBMT and SMT systems.

Source	how long does it take to have them reissued ?
Reference	*onları tekrar çıkarttırmak ne kadar sürer ?*
SMT	*gitmek ne kadar sürer* reissued *onları var ?*
GEBMT	have them reissued *ne kadar sürer ?*

Improvement Using SMT Phrases

Based on the aforementioned observation relating to the example given in Table 4.6, we conduct an experiment that uses phrases from the SMT phrase table as additional atomic translation templates in order to mitigate the issue of untranslated words in the GEBMT system. This is an alternative way of backing off from the SMT system to the GEBMT system. We used all phrase pairs from the SMT system as additional atomic translation templates along with the atomic translation templates learned by the GEBMT system. We recompute the probabilistic score (p_i, p.87) of each atomic translation templates ($T_i : s_i \rightarrow t_i$) based on their frequency in the

GEBMT approach and in the SMT phrase table in order to maintain the probability constraints. We shall refer to this system as GEBMT-PT.

Table 4.7 compares the translation accuracy of the GEBMT and GEBMT-PT systems for English-to-Turkish translation.

Table 4.7: System accuracies obtained using different GEBMT and GEBMT-PT models for English-to-Turkish MT. The subscript $score > x$ denotes the value of the translation score (q).

System	BLEU (in %)	NIST
Training Data: 1242 sentences		
SMT	7.63	2.89
GEBMT	6.80	2.78
GEBMT-PT	**7.10**	**2.81**
$\text{GEBMT}_{\text{score}\geq 0.3}$ + SMT	7.96	2.98
$\text{GEBMT-PT}_{\text{score}\geq 0.3}$ + SMT	**8.19**	**3.01**
Training Data: 2484 sentences		
SMT	10.72	3.51
GEBMT	7.21	3.07
GEBMT-PT	**7.36**	**3.13**
$\text{GEBMT}_{\text{score}\geq 0.9}$ + SMT	10.83	3.52
$\text{GEBMT}_{\text{score}\geq 0.8}$ + SMT	10.99	3.53
$\text{GEBMT}_{\text{score}\geq 0.7}$ + SMT	10.76	3.53
$\text{GEBMT-PT}_{\text{score}\geq 0.9}$ + SMT	**10.85**	**3.53**
$\text{GEBMT-PT}_{\text{score}\geq 0.8}$ + SMT	**11.06**	**3.55**
$\text{GEBMT-PT}_{\text{score}\geq 0.7}$ + SMT	**10.81**	**3.55**

We found that incorporating the additional atomic translation templates improves system accuracy over the GEBMT system. However, the performance of the GEBMT-PT system on its own remains lower compared to the baseline SMT system. The combination of GEBMT-PT with SMT shows improved translation scores when compared to the individual systems. The GEBMT-PT+SMT combination also has better scores than GEBMT+SMT across different threshold values of the translation score.

4.4 Summary

Like the PA-based systems described in the previous chapter, the GEBMT system has shown a similar trend in MT performance, i.e. the performance of the approach on its own is quite poor compared to the baseline SMT system, but it shows improvement when combined with SMT. The standalone GEBMT system is not successful enough to positively answer research question RQ1. However, like the PA-based system, the GEBMT system provides an affirmative answers to research question RQ3 for two different MT tasks showing under certain conditions that we can effectively combine EBMT systems with phrase-based SMT systems. In some of our experiments conducted in this chapter, the improvements of the combined system is quite low and statistically less significant. However, the experiments shows that there are sentences that are better translated by an EBMT approach compared to the SMT-based system. This observation leads us to our research on Chapter 5 and 6 that focuses on both finding a suitable EBMT technique for translating homogeneous data and its effective use for certain sentences to produce the best of EBMT and SMT.

4.4.1 Contributions

Our main contributions to this approach are as follows:

- We introduced two parameters (similarity weight factor and probability of a translation template) to rank the output translation. We developed a decoding strategy using these parameters to produce ranked output translation for a GEBMT system.

- We improved MT accuracy by combining GEBMT with SMT based on translation score. We used two parameters to judge the confidence of a translation produced by GEBMT approach. Based on certain confidence thresholds of the translation score, we achieve better translation quality combining GEBMT and SMT systems.

The moderate success of both the PA-based approach and GEBMT system prompted us to attempt a novel approach of integrating subsentential TMs into an EBMT system, in the next chapter. This EBMT system is developed using the concepts of TM technology and is anticipated to work well on homogeneous data in a resource-poor setting.

Chapter 5

EBMT Using a Subsentential Translation Memory

The results presented in Chapter 3 and Chapter 4 demonstrate that the two different approaches to EBMT (using proportional analogy and translation templates) serve to give a small improvement in translation quality for some of the test sentences. We also found, in most cases the improvements are not statistically significant. We also observed that both approaches suffer from considerable time complexity issues. In addition, both methods demand at least two similar examples from each side of the example-base to produce the translation of a new sentence. In particular, the proportional-analogy-based system requires two example pairs that will cover all the characters of an input test sentence. The compiled approach in Chapter 4 needs two examples to learn a translation template that can be applied to a novel input sentence. It is not often the case that two examples similar to the input sentence are present in a small corpus.

In this chapter, we present a novel approach to EBMT that primarily relies on having only one example pair in the example-base. The approach integrates a subsentential translation memory (TM) into an EBMT system for alignment and recombination. We then present a hybrid SMT-EBMT system using this approach that gives a significant improvement over both SMT and EBMT baseline systems.

The EBMT system is combined with the SMT system based on some underlying features for effective hybridization of the pair of systems.

The chapter is organized as follows. The next section presents the motivation of our particular work. Then we describe the process of automatically building a subsentential TM using SMT technology. In the next section, we describe the detail of our EBMT framework using TM. Subsequently, in Section 5.4, we present the experimental setup, the data and the results obtained with our EBMT system. We show the improvement by combining an SMT-based system and our EBMT system in the following section. Finally, we present our observations with analysis of errors and summarize in Section 5.8 with some avenues for the immediate future work addressed in the next chapter.

5.1 Motivation

The state-of-the-art phrase-based SMT model generally requires a significant amount of training data. Developing such large corpora for a new language pair is costly and time-consuming as noted in Section 2.5 when developing the patient dialogue corpus for English–Bangla. However, small domain-specific parallel corpora are available for many languages for particular usage, e.g. IWSLT corpora. Despite the difficulty of developing a specialized parallel corpora for a new language pair (cf. Section 2.5.1), it is, however, possible to develop a small amount of parallel data for a particular domain in a short period of time and at low cost (Lewis, 2010). These corpora are often homogeneous in nature. In a homogeneous domain-specific corpus, examples are quite close in nature. For example, while IWSLT09 training data is quite small ($\approx$ 20k sentences for English–Turkish), we found the corpus is comprised of very similar domain-specific sentences, as illustrated in (40) and (41). The portions in italics are the only differences between (a) and (b) in the above examples.

(40) a. I'd like to *see that camera* on the *shelf* .

b. I'd like to *have it parted* on the *left* .

(41) a. Have you ever *seen a* Japanese *movie* ?

b. Have you ever *tried* Japanese *food* ?

While using a domain-specific homogeneous corpus, it is likely that the input test sentences also belong to the same domain. We (Dandapat et al., 2010c, 2011) also observed that some sentences in the test set share a large number of surface words with some examples from the example-base. Each of the examples (42), (43) and (44) show the test sentence and the corresponding similar example from the example-bases taken from IWSLT09 (English–Turkish), EMEA (English–French) and our English–Bangla corpus, respectively. The sentences in (a) and (b) represent the test sentence and a similar sentence from the example-base, where the portion in italics denotes the differing parts between them.

(42) a. Does the *tour bus* have a *restroom* ?

b. Does the *room* have a *bath* ?

(43) a. Use in adult patients *with kidney disease but not receiving dialysis* the usual starting dose is *50* iu / kg .

b. Use in adult patients *in an autologous predonation programme* the usual starting dose is *600* iu / kg .

(44) a. I need a medical for my *insurance company* .

b. I need a medical for my *new job* .

The above examples show that the test sentences have a lot in common with a single example from the example-base. In that way, the translation of the sentences in (42a), (43a) and (44a) may share the translation of the common parts in (42b), (43b) and (44b), respectively. Thus it might be helpful to reuse the translation of the common part while translating a new sentence. The above observation leads us to reuse some parts of the sentence which are common to the closest sentence in the example-base in an EBMT system.

In general, an EBMT system can be built with fewer examples (Somers, 2003, p.12) compared to the amount of training data used by an SMT system. Homogeneous domain-specific parallel corpora for many resource-poor languages tend to be able to provide such an example-base. Keeping this in mind, we plan to develop a novel EBMT system that can create a skeleton translation based on the closely-matched example from the example-base. The remaining unmatched subsentential portion (between the input and the closely-matched sentences) can be further translated using other parallel resources.

5.2 Building a Subsentential Translation Memory

As noted in Chapter 2 (Section 2.3), a TM usually contains translation units (TU) linked at the sentence, phrasal and word level. TUs can be derived manually or automatically (e.g. using the marker-hypothesis (Groves and Way, 2006)). Usually, TUs are linguistically motivated translation units. In our work, we explore a different route, as manual construction of high-quality TMs is time consuming and expensive. Furthermore, only considering linguistically motivated TUs may limit the matching potential of a TM. Because of this, we used SMT technology to automatically create the subsentential part of our TM at the phrase (i.e. no longer necessarily linguistically motivated) and word level. Based on Moses word alignment and phrase table construction, we build the additional TM for further use within an EBMT approach.

Moses uses GIZA++ (Och and Ney, 2003) to learn the initial word alignment file based on IBM Model 4 (Brown et al., 1993). GIZA++ learns the word alignment in both source (e) to target (t) and reverse. The final word alignments are taken from the intersection of the bidirectional run of GIZA++. Additional alignments are extracted based on the union of the bidirectional run using the grow-diag-final heuristic (Koehn, 2010, p.112). Finally, these phrases are extracted into a phrase translation table and five probabilities are estimated for each aligned phrase pair.

Figure 5.1 shows the learned phrase pairs with the associated probabilities for a source English phrase *a hotel* to its different target equivalent in Turkish. We add entries to the TM based on the aligned phrase pairs from the Moses phrase table using the following two scores:

(i) Direct phrase translation probabilities: $p(t|e)$

(ii) Direct lexical weight: $lex(t|e)$

We chose $p(t|e)$ and $lex(t|e)$ as we wished to only consider the most probable target equivalent (t) for a given source (e). The reverse probabilities ($p(e|t)$ and $lex(e|t)$), strictly speaking, do not directly model the most probable target equivalent (t) for a given source (e).

English (e)	**Turkish (t)**	**p(*e*|*t*)**	**lex(*e*|*t*)**	**p(*t*|*e*)**	**lex(*t*|*e*)**	**penalty**
a hotel	bir otel	0.95	0.505436	0.826087	0.12843	2.718
a hotel	bir otelde	0.166667	0.294511	0.0869565	0.073134	2.718
a hotel	otel mi	0.5	0.0521575	0.0434783	0.0066215	2.718
a hotel	otel	0.0128205	0.0521575	0.0434783	0.223603	2.718

Figure 5.1: Moses phrase equivalents with associated probabilities.

Table 5.1 shows some of the English-to-Turkish translation units in the TM. Note that the entries in the TM (including those in Table 5.1) may contain incorrect source–target equivalents due to unreliable word/phrase alignment produced by Moses.

Firstly, we add entries to the TM based on the aligned phrase pairs from the Moses phrase table. A source phrase may have multiple target equivalents. We keep all target equivalents in a sorted order based on the phrase translation probability $p(t|e)$ and the lexical probability $lex(t|e)$. These two probabilities are highlighted in Figure 5.1. First we sort the target phrases based on $p(t|e)$. If there exists a tie in $p(t|e)$ among target phrases, we use $lex(t|e)$ to rank the possible target equivalents. It has been observed that more than one target phrase sometimes has the equal $p(t|e)$. For example, the two Turkish target equivalents (*otel mi* and *otel*) have the

same $p(t|e)$ for the source English phrase *a hotel*. In order to rank such cases, we use $lex(t|e)$ to avoid the conflict. The final ranked target phrases for the source English phrase *a hotel* are shown in the third row of the Table 5.1.

Table 5.1: Source–target translation equivalents in TM

Source(English)	**Target(Turkish)**
Example entries in TM from Moses phrase table	
i don't like it	{“sevmedim”,“bunu sevmedim”}
i can't sleep well.	{“iyi uyuyamıyorum .”}
a hotel	{“bir otel”,“bir otelde”,“otel”,“otel mi”}
load this camera	{“bu kamerayı yükler”}
Example entries in TM from Moses word-aligned file	
coffees	{“kahve”}
fair	{“fuar”,“bayanımı”,“ortalama”}
helps	{“vücudun”,“yardım”,“eder”}
playground	{“alanı”,“oyun”}

Secondly, in addition to the phrase table, additional entries in the subsentential TM are extracted from the source–target lexical table. We also keep the multiple target equivalents for a source word in a sorted order. This essentially adds source- and target-language equivalent word pairs into the TM. Moses builds a source–target lexical translation table based on the GIZA++ word alignment with associated probability $w(t|e)$. Figure 5.2 shows the lexical equivalents learned by GIZA++ with associated probabilities. We rank the target translation for a given source word based on this lexical probability $w(t|e)$. These lexical translation pairs are also kept in our subsentential TM. The sixth row in the Table 5.1 depicts the sorted lexical translation equivalent in the TM for the English source word *fair*.

English (e)	**Turkish (t)**	***w(t|e)***
fair	faur	0.50
fair	bayanımı	0.25
fair	ortalama	0.25

Figure 5.2: Moses lexical equivalents with associated probabilities.

We keep all the target equivalents for a word/phrase to identify the matched

segment between source and target language in the matching procedure (see section 5.3.2). But during recombination we only consider the most probable target equivalent (see section 5.3.3). The ranked list of possible translations in the TM helps us to choose the most probable target equivalent for a source word.

We find four Turkish target equivalents ('bir otel','bir otelde', 'otel', 'otel mi') for the English source phrase 'a hotel' in Table 5.1. The phrase *bir otel* (a/one hotel+nominative) is the most probable translation based on the Moses phrase table. The second most probable Turkish phrase *bir otelde* refers to 'a/one hotel+locative'. The third target equivalent *otel* refers to 'hotel+nominative'. This is because Turkish noun phrases may not always have an article. The fourth target Turkish equivalent *otel mi* refers to 'hotel+question'. In the example shown in Figure 5.2, the English word 'fair' also has three target Turkish equivalents ('fuar', 'bayanımı','ortalama') extracted from the Moses word-aligned file. The target equivalent *fuar* is used as a noun (e.g. book *fair*), and has the highest probability based on Moses word alignment. The second target equivalent *bayanımı* refers to *my lady*. The third Turkish equivalent *ortalama* is used as an adjective which denotes something moderately large (e.g. a *fair* income).

5.3 Approach

Like most EBMT systems, our particular approach comprises three stages: *matching*, *alignment* and *recombination*.

5.3.1 Matching

The first step in an EBMT system is to find source-language examples that closely match the input sentences. In particular, in our approach, we find *the closest sentence* (s_c) from the example-base for the input sentence (s) to be translated, as in Equation (5.1).

$$s_c = \arg\max_{s_i} \text{score}(s, s_i) \tag{5.1}$$

We used a word-based edit distance metric (Wagner and Fischer, 1974) to find the closest matching sentence from the example-base ($\{s_i\}_1^N$) based on Equation (5.2).

$$\text{score}(s, s_i) = 1 - \frac{\text{ED}(s, s_i)}{\max(|s|, |s_i|)} \tag{5.2}$$

where $|x|$ denotes the length (in words) of a sentence, and $\text{ED}(x, y)$ refers to the word-based edit distance between x and y.

Based on the above fuzzy scoring criteria, we are able to choose the closest match (s_c) for the input sentence (s) to be translated.

We take two running examples to describe the work-flow of our EBMT approach. These two examples are indicative of some of the different possible operations in the later stage due to the difference in matching segments. For example, for the input sentences in (45a) and (46a) from the IWSLT09 test data, the corresponding closest fuzzy-matched sentences from the example-base (IWSLT09 training data) are given in (45b) and (46b).

(45) a. s: i'd like a present for my mother.

b. s_c: i'd like a shampoo for my greasy hair.

(46) a. s: take two tablets after every meal.

b. s_c: please take two tablets after each meal.

Then we consider the associated translations (t_c) in (47) and (48) of the closest matching source sentence in (45b) and (46b), to build a skeleton for the translations of the input sentences (45a) and (46a).

(47) t_c: yağlı saçlar için bir şampuan istiyorum .
[GREASY HAIR FOR ONE SHAMPOO I'D-LIKE]

(48) t_c: lütfen her yemekten sonra iki tablet alın .
[PLEASE EACH MEAL AFTER TWO TABLET TAKE]

We will use some segments of the associated skeleton translations (47) and (48) to produce the new translation for the input sentences (45a) and (46a) in the alignment and recombination steps. Note that we find the closest matching sentence at runtime from the whole example-base using the edit-distance-based fuzzy match score. Thus the time complexity matching step of our EBMT system is $O(nm^2)$, where n denotes the size of the example-base and m denotes average length (in words) of a sentence.

5.3.2 Alignment

After matching and retrieving an example with its associated translation, the next step is to extract the non-matching fragments from that translation. In order to do that, we align the three sentences: the input (s), the closest source-side match (s_c), and its target equivalent (t_c).

First, we mark the mismatch portion between s and s_c while computing the edit distance in Equation (5.2). We use the edit-distance trace algorithm (as described in section 4.2.1) to find matched and non-matched segments between s and s_c. Given the two sentences (s and s_c), the algorithm finds the minimum possible number of operations (substitutions, additions and deletions) required to change the closest match s_c into the input sentence s. For example, consider the input sentence $s = w_1\, w_2\, w_3\, w_4\, w_5\, w_6\, w_7\, w_8$ and $s_c = w'_1\, w'_3\, w_4\, w_5\, w_7\, w_8\, w'_9\, w'_{10}$. Figure 5.3 shows the matched and non-matched sequence between s and s_c using edit-distance trace.

$$
\begin{array}{lcccccccccc}
s & = & w_1 & w_2 & w_3 & w_4 & w_5 & w_6 & w_7 & w_8 & - & - \\
 & & | & & & | & | & & | & | & & \\
s_c & = & w_1 & - & w'_3 & w_4 & w_5 & - & w_7 & w_8 & w'_9 & w'_{10}
\end{array}
$$

$$\Downarrow$$

$$
\begin{array}{lcccccc}
s & = & \underline{w_1} & \boxed{w_2\, w_3} & \underline{w_4\, w_5} & \boxed{w_6} & \underline{w_7\, w_8} & \boxed{null} \\
 & & | & \downarrow & | & \downarrow & | & \downarrow \\
s_c & = & \underline{w_1} & \boxed{w'_3} & \underline{w_4\, w_5} & \boxed{null} & \underline{w_7\, w_8} & \boxed{w'_9\, w'_{10}}
\end{array}
$$

Figure 5.3: Extraction of matched and non-matched segments between s and s_c.

First we identify the edit operations required to convert s_c into s as shown in the upper half of the figure. The consecutive matched words and consecutive non-matched words are combined into a single segment. This is shown in the bottom half of the figure, where matched segments are marked with underlines and unmatched segments are marked with boxes. The edit operations are marked with vertical arrows corresponding to the non-matched segments to convert s_c into s. The three operations indicate that w'_3 needs to substitute with $w_2\, w_3$, w_6 needs to be added immediately after w_5 in s_c, and $w'_9\, w'_{10}$ need to be deleted from s_c. This is shown in (49a) and (49b) with angled brackets. The character and the following numbers in angled brackets indicates the edit operation ('s' indicates substitution) and the index of the mismatched segments. In the second example in (50a) and (50b), where 'd#' within angled brackets indicates the translation of the corresponding segment that need to be deleted from the final output. Note that the swapped order of substitutions in t_c is obtained by the alignment process between s_c and t_c using subsentential TM as described below the examples.

(49) a. s: i 'd like a <s#0:*present*> for <s#1:*my mother*> .

b. s_c: i 'd like a <s#0:*shampoo*> for <s#1:*greasy hair*> .

c. t_c: <s#1:*yağlı saçlar*> için bir <s#0:*şampuan*> istiyorum .

(50) a. s: take two tablets after <s#0:*every*> meal .

b. s_c: <d#:*please*> take two tablets after <s#0:*each*> meal.

c. t_c: <d#:lütfen> <s#0:her> yemekten sonra iki tablet alın .

We align each non-matched segment in s_c with its associated translation t_c using the TM and GIZA++ alignment. The alignment process for the non-matched segment is as follows:

- First, we rely on the subsentential TM to find the target equivalent segment in t_c for a non-matched segment in s_c. We only use the portion of TM that

has been constructed from the phrase table.[1] First we look for the target equivalent in t_c for the entire non-matched segment in s_c. We mark the target equivalent in t_c for a non-matched segment in s_c. It is often the case that the TM does not have a target equivalent for an entire non-matched segment. However, the TM may have a target equivalent for some subsegment of a particular non-matched segment. We find the longest possible segment from the non-matched segment in s_c that has a matching target equivalent in t_c based on the source–target equivalents in the TM. We continue the process recursively until no further segments of the non-matched segment in s_c can be matched with t_c using the TM.

- The use of TM may not find target correspondences for all the words of a non-matched segment in s_c. In the second step, we use the GIZA++ word alignment information to align remaining words from the non-matched segment in s_c with its equivalents in t_c. GIZA++ essentially produces the alignments between s_c and t_c in both directions. We use the target-to-source direction where each source word is listed with its reference to aligned target words. Example (51) shows the alignment for the English–Turkish sentence pair in (49b) and (49c). The numbers in the brackets on the source side indicate the position of aligned target words. We mark the target equivalents for all remaining words of a non-matched segment based on the GIZA++ alignment.

(51) a. s_c: NULL ( ) i ( ) 'd ( ) like (6) a (4) shampoo (5) for (3) greasy (1 2) hair ( ) . (7)

b. t_c: yağlı saçlar için bir şampuan istiyorum .

Based on the source–target aligned pair from the TM and GIZA++, we mark the mismatched segment in the t_c as in (49c) and (50c). The portions marked

[1]The source–target equivalent in the phrase table is more reliable as it has been constructed based on the intersection of the bidirectional alignment of GIZA++. In contrast, the lexical alignment is based on all possible GIZA++ alignments which is much more noisy. Thus, during alignment we rely on those TM entries that have been constructed from the Moses phrase table.

with angled brackets in (49c) and (50c) are aligned with the mismatched portions in (49b) and (50b), respectively. Here also, the first character and the following number within an angled bracket in t_c indicates the mapping between the segments with s_c.

With the help of the above matching method, in the recombination step, we replace/delete the segments within the angled brackets in t_c keeping the remaining matched fragments unchanged.

5.3.3 Recombination

The final step of this EBMT approach is recombination. We add or substitute segments from the input sentence (s) with the skeleton translation equivalent (t_c). We also delete some segments from t_c that have no correspondence in s. From (49), we need to replace the two segments in (49c) (*yağlı saçlar* (greasy hair) and *şampuan* (shampoo)) with the two corresponding source segments in (49a) (*my mother* and *present*) to produce a target equivalent. Thus, keeping the mapping, we produce the skeleton target equivalent in (52):

(52) <s#1:*my mother*> için bir <s#0:*present*> istiyorum .

From (50), we need to delete one segment in (50c) *lütfen* which is a Turkish translation equivalent of the English word *please* in s_c. We also need to substitute the segment *her* with its corresponding source segment *every* in (50a) to produce the target equivalent. Thus, we produce the skeleton translation in (53).

(53) <s#0:every> yemekten sonra iki tablet alın .

If there are some extra segments in s which do not have any mapping in t_c, then we add the new segments from s into the target equivalent t_c. Thus we produce the target equivalents in (52) and (53) after adding/deleting/substituting segments from the input sentences to be translated (s) with the skeleton translation (t_c). Then, the untranslated segments in (52) and (53) are translated using our subsentential TM. The detail of the algorithm is given in Algorithm 3.

Algorithm 3 recombination(X, TM)

In: source segment X,
subsentential translation memory TM
Out: translation of source segment X

1: mark all words of X as untranslated (untranslatedPortions(X) $\leftarrow \{X\}$)
2: **repeat**
3: U = untranslatedPortions(X)
4: x = longest subsegment in untranslatedPortions(X) such that $(x, t_x) \in$ TM;
5: substitute($X, x \rightarrow t_x$) {substitute x with its target equivalent t_x in X}
6: remove x from untranslatedPortions(X)
7: **until** (untranslatedPortions(X) = $\emptyset$)
8: return X

Replacing the untranslated segments in (52) and (53) with the corresponding translations obtained using TM, we derive the output translations shown in (54) and (55), respectively, of the original input sentences.

(54) <*annem*> için bir <*hediye*> istiyorum .

(55) <*her*> yemekten sonra iki tablet alın .

Note that unknown words are left untranslated, which is the case for most MT techniques. Incorrect translations may be expected due to incorrect word/phrase alignments.

5.4 Experiments

We conduct three different experiments for three different language pairs (English-to-Bangla, English-to-Turkish and English-to-French) to set the baseline and to understand the performance of our EBMT system for each language pair.

5.4.1 Experimental Setup

First we conduct two experiments to estimate the baseline accuracy of the MT systems.

- **OpenMaTrEx**: Our first baseline is the performance of an SMT-based system. We use OpenMatrEx (Dandapat et al., 2010a) to estimate the baseline phrase-based SMT accuracy and to compare results with our approach.

- **TM**: We conduct a second baseline experiment based on the matching step (cf. section 5.3.1) of our EBMT system. This is essentially an output extracted based on TM matching. We obtain the closest target-side equivalent (the skeleton sentence t_c as in (47, p.103)) and consider this as the baseline output for the input sentence (as in (45a, p.109)) to be translated. This is essentially reflects the translation accuracy of the TM match. We produce the final translation based on this TM match (initial skeleton translation) thus we will consider this as the baseline accuracy for our EBMT system using TM.

In addition, we conduct an experiment with our EBMT system using subsentential TM.

- **EBMT$_{\mathbf{TM}}$**: After obtaining the skeleton translation through matching and alignment (section 5.3), in the recombination step we use the subsentential TM to translate any unmatched segments as described in Algorithm 3. We call this EBMT$_{\mathrm{TM}}$.

5.4.2 Data Sources

We used three data sets for all our experiments. The three data sets represent three language pairs of different size and type. In the first dataset, we used our in-house English–Bangla medical receptionist dialogue corpus (described in section 2.6.1). The training data consists of 380 parallel sentences from a medical receptionist dialogue exchange. The test set is comprised of a disjoint set of 41 dialogue turns. Note that the dialogue corpus is homogeneous in nature with short sentences. The average length (in words) of the sentences in the source- and target-side training data are 8.5 and 8.27, respectively.

In the second dataset, we used the same English–Turkish corpus from IWSLT09 as described in section 2.6.2. The IWSLT09 training data consists of 19,972 parallel sentences. We used the IWSLT09 development set as our test set which consists of 414 sentences. The IWSLT09 data also belongs to a particular domain (it is a

subset of the C-STAR[2] project's Basic Traveller Expression Corpus). The corpus also consists of shorter length sentences with an average length (in words) of 9.5 for the source side and 6.9 for the target side.

Our third dataset consists of an English–French corpus from the European Medicines Agency (EMEA)[3] (Tiedemann and Nygaard, 2009). The training data consists of 250,806 unique parallel sentences.[4] As a test set we used a set of 10,000 randomly drawn sentences disjoint from the training data. These data also represent a particular domain (medicine) but with a longer sentence length compared to the English–Bangla and English–Turkish data. The average length (in words) of the sentences in the source- and target-side training data are 18.8 and 22.61, respectively. This is a moderate sized corpus in terms of the amount of data generally used to train an MT system.

5.4.3 Immediate Results and Observations

For consistency with previous experiments we used BLEU (Papineni et al., 2002) and NIST (Doddington, 2002) for the automatic evaluation of our experiments. Table 5.2 and Table 5.3 show the BLEU score obtained by the three different experiments (described in section 5.4.2) for the three language pairs.

Table 5.2: Baseline BLEU scores (%) of the two systems and the scores for EBMT_{TM} system.

System	Language pairs		
	English-to-Bangla	English-to-Turkish	English-to-French
SMT	39.32	**23.59**	**55.04**
TM	50.38	15.60	40.23
EBMT_{TM}	**57.56**	20.08	48.31

Note that the baseline BLEU score for SMT for English-to-Bangla translation is 39.32 but the baseline TM match gives a BLEU score of 50.38. This absolute improvement of 11.06 BLEU points motivated us to use the skeleton sentence (t_c)

[2]Consortium for Speech Translation Advanced Research. `http://www.c-star.org/`

[3]http://opus.lingfil.uu.se/EMEA.php

[4]A large number of duplicate sentences exists in the original corpus (comprised of approximately 1M sentences). We remove duplicates and consider sentences with unique translation equivalents.

Table 5.3: Baseline NIST scores of the two systems and the scores for $EBMT_{TM}$ system.

System	Language pairs		
	English-to-Bangla	English-to-Turkish	English-to-French
SMT	4.84	**4.85**	**11.01**
TM	5.32	3.34	7.98
$EBMT_{TM}$	**6.00**	4.41	9.72

and make further changes to some fragments in the skeleton sentence with respect to the original sentence (s) to be translated. We have achieved 57.56 BLEU score with our $EBMT_{TM}$ system — an absolute improvement of 7.09 BLEU score over the baseline TM match. The high BLEU score is due to the nature of the data used for this experiment. The training and test examples are homogeneous in nature and comprised of domain specific sentences. Also, the baseline TM match has a better score than the baseline SMT system due to this homogeneity. The improvement of the $EBMT_{TM}$ system is statistically significant (with a reliability of 99%) using boot strap resampling (Koehn, 2004).

The situation differs in the case of English-to-Turkish and English-to-French translations using a moderate sized corpus. We found in both English-to-Turkish and English-to-French translation, the baseline SMT system has better BLEU scores (23.59 and 55.04 BLEU points, respectively) compared to the baseline TM match (15.60 and 40.31 BLEU points, respectively) and our $EBMT_{TM}$ system (20.08 and 48.31 BLEU points, respectively). However, the $EBMT_{TM}$ has an absolute improvement of 4.48 and 8.08 BLEU points over the TM-based matching, respectively, for English-to-Turkish and English-to-French systems.

We find a similar trend using the NIST evaluation metric. In the case of English-to-Bangla translation, the $EBMT_{TM}$ system gets a better score compared to the two baselines (SMT and TM). On the other hand, in both English-to-Turkish and English-to-French, the baseline SMT system gets a better score compared to the $EBMT_{TM}$ system. However, the $EBMT_{TM}$ system shows improvement over the baseline TM match for both the language pairs.

Though the EBMT_{TM} system has lower translation accuracy compared to the baseline SMT system, there are considerable amount of sentences for which the EBMT_{TM} system produces better translations.

5.5 Improvement

The EBMT_{TM} system has low translation accuracy on its own compared to the baseline SMT for both English-to-Turkish and English-to-French experiments. However, we observed that there are cases where EBMT_{TM} produces better quality translation compared to the SMT-based approach and vice-versa (Dandapat et al., 2011). Thus we plan to use a combination of both EBMT_{TM} and SMT for effective hybridization of the pair of systems by choosing the best approach for each input to produce a better quality translation system.

5.5.1 System Combination

During system combination we use features to decide whether to rely on the output produced by the EBMT_{TM} system or to rely on the SMT-based output. We use the following two features for combining both the systems.

- **FMS**: We use *fuzzy match score* (FMS) (as in Equation (5.2)) as one of our features in order to trigger the output of the combined system from the output produce by the EBMT_{TM} system. This feature essentially indicates the nearness of the closest-matched sentence (s_c) for a given test sentence (s). As a higher FMS value between s and s_c indicates a greater percentage match between the surface words, it is also likely that when the FMS is high, a fewer number of changes need to be made to the skeleton translation (t_c) to produce the translation of s.

- **EqUs**: This feature refers to the equal number of unmatched segments (EqUs) between s, s_c and t_c. This is a binary valued feature. If there is an equal

number of non-matched segments among s, s_c and t_c, then this feature is set to 1, otherwise it is set to 0. This generally indicates only substitution operations need to be performed between s and s_c and we are able to find target correspondences for all non-matched segments in s_c. This feature can be useful due to the fact that if only substitutions are required to convert s_c to s, it is likely that they have the same grammatical structure, especially when the length of the mismatched segment is short (e.g. a single word or a phrase as in Example (49, p.111)). In contrast, addition and deletion in alignment indicates changes to the grammatical structure of the sentence.

We combine EBMT_{TM} and SMT based on the above features. We assume that the translation of an input sentence s produced by EBMT_{TM} and SMT systems are $\text{T}_{\text{EBMT}}(s)$ and $\text{T}_{\text{SMT}}(s)$, respectively. If the value of the FMS is greater than some threshold and EqUs exists between s, s_c and t_c, we rely on the output $\text{T}_{\text{EBMT}}(s)$; otherwise we take the output from $\text{T}_{\text{SMT}}(s)$. We refer to this system as $\textbf{EBMT}_{\textbf{TM}} + \textbf{SMT}$.

5.5.2 Experiments and Results Using the Combined System

We conducted different experiments with the combined system ($\text{EBMT}_{\text{TM}} + \text{SMT}$) using different feature combinations and varying the FMS threshold. We tested the $\text{EBMT}_{\text{TM}} + \text{SMT}$ system for English-to-Turkish and English-to-French translation where the EBMT_{TM} system has a lower score than the baseline SMT system.[5]

Table 5.4 shows the accuracies obtained with the combined system using different feature combinations for English-to-Turkish translation. We found that though EBMT_{TM} has a lower accuracy compared to the baseline SMT, combining it with SMT has a positive effect. We found that the combined system performs better (highest relative improvement of 3.48% in BLEU and 1.03% in NIST with an overall

[5]We do not use the combined system for the English-to-Bangla experiment as the performance of the baseline EBMT_{TM} system is well above the baseline SMT system and our focus is on improving EBMT system.

Table 5.4: English-to-Turkish MT system results for the $\text{EBMT}_{\text{TM}} + \text{SMT}$ system with different combining factors. The second column indicates the number (and percentage) of sentences selected from the EBMT_{TM} system during combination.

System: $\text{EBMT}_{\text{TM}} + \text{SMT}$			
Condition	**No. of times EBMT_{TM} used**	**BLEU (in %)**	**NIST**
Baseline SMT: BLEU=23.59% and NIST=4.85			
FMS>0.85	35 (8.5%)	24.22	4.89
FMS>0.80	114 (27.5%)	23.99	4.84
FMS>0.70	197 (47.6%)	22.74	4.73
FMS>0.80 || (FMS>0.70 & EqUS)	165 (40.0%)	23.87	4.83
FMS>0.85 & EqUS	24 (5.8%)	**24.41**	**4.90**
FMS>0.80 & EqUS	76 (18.4%)	24.19	4.88
FMS>0.70 & EqUS	127 (30.7%)	24.08	4.87

score of 24.41 and 4.9, respectively, for BLEU and NIST) compared to the baseline SMT approach. We found that if an input has a high FMS with the example-base, then the EBMT_{TM} system does better compared to SMT. We found that a FMS over 0.8 showed an improvement over the SMT-based approach with our current experimental setup. Improvements are statistically significant (reliability of 99%), but only for a very high FMS (>0.85). However, FMS might not be the only factor for triggering EBMT_{TM}. We consider EqUS as another factor. Though an FMS over 0.7 shows no improvement in overall system accuracy, inclusion of the EqUS feature along with FMS shows improvement. Thus, the EBMT_{TM} approach is more effective if the number of non-matched segments correspond in s, s_c and t_c.

Table 5.5 shows that combining EBMT_{TM} with SMT also shows improvement over the baseline SMT system for English–French data set. Here also, we found that the combined system has the highest relative improvement of 4.99% in BLEU points and 3.18% in NIST over the baseline SMT approach. The improvements of $\text{EBMT}_{\text{TM}} + \text{SMT}$ over the baseline SMT are statistically significant (reliability 99%) using bootstrap resampling. The highest accuracy has been achieved with the feature FMS>0.85. Here, we found that the combined system relied on the output of the EBMT_{TM} system for a large number of sentences (one third of the test sentences with highest improvement). This is due to the fact, that the English–

Table 5.5: English-to-French MT system results for the combined $EBMT_{TM}$ + SMT system with different combining factors.

System: $EBMT_{TM}$ + SMT			
Condition	**No. of times $EBMT_{TM}$ used**	**BLEU (in %)**	**NIST**
Baseline SMT: BLEU=55.04% and NIST=11.01			
FMS>0.85	3323 (33.2%)	**57.79**	**11.36**
FMS>0.80	4300 (43.0%)	57.55	11.31
FMS>0.70	5283 (52.8%)	57.05	11.24
FMS>0.60	6148 (61.5%)	56.25	11.1
FMS>0.50	6148 (61.5%)	54.98	10.89
FMS>0.80 || (FMS>0.70 & EqUS)	4707 (47.1%)	57.46	11.31
FMS>0.85 & EqUS	2358 (23.6%)	57.24	11.29
FMS>0.80 & EqUS	2953 (29.5%)	57.16	11.28
FMS>0.70 & EqUS	3360 (33.6%)	57.08	11.26
FMS>0.60 & EqUS	3664 (36.6%)	56.92	11.24

French EMEA corpus is built using translation memory and results in a large number of test sentences getting a high FMS.

5.6 Manual Evaluation

In addition to the above automatic evaluations, we performed a manual evaluation of the MT output for all the language pairs to understand the translation quality from a human perspective. While manually evaluating the MT systems, we assign values from two five-point scales representing *fluency* and *adequacy* (Ma and Cieri, 2006). The five-point scale of adequacy indicates how much meaning is conveyed in the hypothesis translation in connection to the reference translation. The five-point scale of fluency indicates the closeness of the hypothesis translation to natural text. These two scales are explained in Table 5.6.

In order to test the reliability of our manual evaluation, we measure the inter-annotator agreement (IA). IA is a good indicator of the reliability of manual evaluation by different human evaluators (Dandapat et al., 2009). We used Fleiss' kappa measure (Fleiss, 1971) for assessing the reliability of agreement between different human evaluators. Values of kappa can range from -1.0 to 1.0, with -1.0 indicating

Table 5.6: Human MT evaluation scales

Fluency	Adequacy (meaning expressed)
5=Flawless Output	5=All
4=Good Output	4=Most
3=Non-native Output	3=Much
2=Disfluent Output	2=Little
1=Incomprehensible	1=None

perfect disagreement, and 1.0 denotes perfect agreement. Conventionally, a kappa score of <0.2 is considered poor agreement, 0.21–0.4 fair, 0.41–0.6 good, 0.61–0.8 strong, and more than 0.8 near-complete agreement (Landis and Koch, 1977).

First, we performed a manual evaluation of all 41 sentences of the English-to-Bangla MT output. Using the evaluation scale in Table 5.6, four different native Bangla speakers[6] were asked to score each translation produced by the different MT systems. Table 5.7 shows the average fluency and adequacy of the two MT approaches (SMT and EBMT_{TM}) for English-to-Bangla translation. The IA between

Table 5.7: Average fluency and adequacy of the English-to-Bangla MT system on a scale of 1-5 (as in Table 5.6).

System	**Fluency**	**Adequacy**
SMT	3.00	3.16
EBMT_{TM}	3.50	3.70

the 4 evaluators for fluency and adequacy of the SMT output are, respectively 0.45 and 0.41 for English-to-Bangla MT output. We found a higher IA for both fluency and adequacy (0.51 and 0.46, respectively) for the output of the EBMT_{TM} system compared to the SMT output. All these IA scores indicate reasonably good agreement between the human-annotators thereby assuring the reliability of the manual evaluation process.

In order to acquire a deep insight into the EBMT_{TM} system output, we conducted a manual analysis[7] of a subset of the EBMT_{TM} system's output against the baseline

[6]All evaluators for English-to-Bangla translations have good English skills and a strong educational background, having achieved at a least post-graduate degree. The evaluation work was done voluntarily.

[7]The evaluators were agnostic of the systems (EBMT_{TM} and baseline SMT) producing the translations, they were comparing.

Table 5.8: Manual inspection of reasons for improvement in English-to-Bangla translation.

N=41 test sentences				
	$EBMT_{TM}$ improves over baseline SMT (N=24)		Baseline SMT improves over $EBMT_{TM}$ (N=7)	
Reason	#	%	#	%
Better word order	8	25.8%	2	6.5%
Better phrase/word selection	14	45.1%	4	13%
Better verb translation	2	6.5%	1	3.3%

Table 5.9: Average fluency and adequacy of the English-to-Turkish MT systems on a scale of 1-5 (cf. Table 5.6). n=number of sentences evaluated under a particular feature value.

System	**Feature**	**n**	**Fluency**	**Adequacy**
Overall performance of the systems				
SMT	-	100	3.31	3.4
$EBMT_{TM}$	-	100	3.27	2.96
$EBMT_{TM}$ + SMT	FMS>0.85	100	3.34	3.42
$EBMT_{TM}$ + SMT	FMS>0.8	100	**3.42**	**3.53**
Performance of the systems for sentences with high FMS				
SMT	FMS>0.85	8	4.25	4.38
$EBMT_{TM}$	FMS>0.85	8	4.63	4.63
SMT	FMS>0.8	32	3.91	4.03
$EBMT_{TM}$	FMS>0.8	32	4.25	4.23

SMT translations. We asked the evaluators to manually compare the $EBMT_{TM}$ and baseline SMT output, with the aim of finding an explanation as to why $EBMT_{TM}$ improved over the baseline SMT and vice versa. We tried to classify the reason for improvement into a few predefined classes.

We manually inspected all of the 41 test sentences of the English-to-Bangla experiment to inspect the reason for improvement. We found that both systems produce the same output for 10 test sentences. For the remaining 31 sentences, $EBMT_{TM}$ system shows an improvement for 24 sentences over the baseline SMT and the baseline SMT system shows an improvement for 7 sentences over the $EBMT_{TM}$ system. Table 5.8 exhibits the reason for improved translations of the $EBMT_{TM}$ over the baseline SMT system and vice versa. Some example sentences for improved translations are illustrated in Table 5.13 (p.127) and Table 5.14 (p.128).

We manually evaluated the English-to-Turkish MT output for 100 sentences

(randomly chosen from our testset) by 3 evaluators. This evaluation was performed with the help of DCU Translation Service.[8] We paid the native Turkish speakers (with good educational background and English skills) to perform the task. Table 5.9 shows the average fluency and adequacy of three different approaches (SMT, EBMT_{TM} and $\text{EBMT}_{\text{TM}} + \text{SMT}$) for English-to-Turkish MT output. We found that the human evaluation corelates with the automatic evaluation (cf. Table 5.4). We measured Pearson's correlation coefficient (Soper et al., 1917) to estimate the correlation between automatic and human evaluations. We found a high correlation (r) between BLEU and fluency ($r = 0.71$), and between NIST and adequacy ($r = 0.91$). We found that EBMT_{TM} on its own has lower fluency (3.27) and adequacy (2.96) compared to the baseline SMT system. However, the combined system ($\text{EBMT}_{\text{TM}} + \text{SMT}$) improves with respect to both fluency and adequacy over the baseline SMT system. The fluency of the $\text{EBMT}_{\text{TM}} + \text{SMT}$ (with FMS$>$ 0.8) and SMT systems are 3.42 and 3.3, respectively. The adequacy for these two systems are 3.53 and 3.40, respectively. Furthermore, we compared the fluency and adequacy of SMT and EBMT_{TM} systems for those sentences with high fuzzy match scores (FMS). This shows a larger improvement in fluency and adequacy for EBMT_{TM} system over the baseline SMT as shown in the last 4 lines of Table 5.9. The average IA between the 3 evaluators for fluency and adequacy are 0.50 and 0.56, respectively.

We also manually investigated the output in order to discover why EBMT_{TM} does better than SMT and vice versa. We studied the sentences with high FMS (> 0.85 and > 0.8) from the manually evaluated 100 sentences. Table 5.10 shows the reasons for the improvements for English-to-Turkish MT. The examples for this improved translations of one system over the other are included in Table 5.13 and Table 5.14. Note that both EBMT_{TM} and SMT systems produce the same output translation for 7 sentences out of 32 sentences with FMS$>$0.8.

Finally, we also manually evaluated 100 random test sentences from our English–French testset using 3 evaluators.[9] Table 5.11 shows the average fluency and ade-

[8] http://dculs.dcu.ie/

[9] This evaluation was performed in-house as a voluntary work. All the evaluators for this task

Table 5.10: Reasons for improvement in English-to-Turkish translation.

N=8 test sentences with FMS>0.85				
	$EBMT_{TM}$ improves over baseline SMT (N=5)		Baseline SMT improves over $EBMT_{TM}$ (N=3)	
Reason	#	%	#	%
Better word order	1	12.5%	1	12.5%
Better phrase/word selection	5	62.5%	2	25.0%
Better verb translation	2	25.0%	0	0.0%
N=32 test sentences with FMS>0.8				
	$EBMT_{TM}$ improves over baseline SMT (N=17)		Baseline SMT improves over $EBMT_{TM}$ (N=8)	
Reason	#	%	#	%
Better word order	8	25.0%	1	3.1%
Better phrase/word selection	9	28.1%	7	21.8%
Better verb translation	5	15.6%	4	12.5%
Fewer OOV words	3	9.4%	0	0.0%

quacy of the different MT outputs. As with our English-to-Turkish system, we found that for English-to-French, the $EBMT_{TM}$ system has a lower score on its own compared to the baseline SMT system. However, the combined $EBMT_{TM}$ + SMT system shows improvements for both fluency and adequacy over the baseline SMT system, co-relating with the automatic evaluation scores (cf. Table 5.5). The Pearson's correlation coefficient between BLEU and fluency is 0.90, and between NIST and adequacy is 0.89. The evaluation shows a small improvement by the $EBMT_{TM}$ + SMT system over the baseline SMT system in both fluency (4.3 to 4.47) and adequacy (4.25 to 4.45). This is due to the fact that a large number of sentences produce the same translation output by both systems (shown in Table 5.12). Furthermore, while looking into those sentences with a high FMS, we found larger improvements by $EBMT_{TM}$ system in both fluency and adequacy over the baseline SMT system. This is shown in the bottom half of Table 5.11. The average IA between the 3 evaluators for fluency and adequacy are 0.53 and 0.55, respectively.

For a large number of sentences, we found that both $EBMT_{TM}$ and baseline SMT systems produce equivalent translations. Out of 31 sentences (with FMS>0.85), we found 22 sentences receive the same translations by both systems. Similarly, both

are native French speakers having good education background with knowledge of MT.

Table 5.11: Average fluency and adequacy of the English-to-French MT systems on a scale of 1-5 (cf. Table 5.6). n=number of sentences evaluated under a particular feature value.

System	**Feature**	**n**	**Fluency**	**Adequacy**
Overall performance of the systems				
SMT	-	100	4.3	4.25
EBMT_{TM}	-	100	4.17	3.96
EBMT_{TM} + SMT	FMS>0.85	100	4.45	4.44
EBMT_{TM} + SMT	FMS>0.8	100	**4.47**	**4.45**
Performance of the systems for sentences with high FMS				
SMT	FMS>0.85	31	4.31	4.22
EBMT_{TM}	FMS>0.85	31	4.58	4.61
SMT	FMS>0.8	47	4.37	4.31
EBMT_{TM}	FMS>0.8	47	4.62	4.62

EBMT_{TM} and baseline SMT produce same translations for 31 sentences out of 47 sentences having FMS>0.8. Table 5.12 shows the reasons for improvement by one system over another (cf. Table 5.13 and Table 5.14) for the remaining sentences where different translations are produced by the systems.

Table 5.12: Reasons for improvement in English-to-French translation.

N=31 test sentences with FMS>0.85				
	EBMT_{TM} improves over baseline SMT (N=7)		Baseline SMT improves over EBMT_{TM} (N=2)	
Reason	#	%	#	%
Better word order	4	12.9%	0	0.0%
Better phrase/word selection	5	16.1%	2	3.2%
Better verb translation	1	3.2%	0	0.0%
N=47 test sentences with FMS>0.8				
	EBMT_{TM} improves over baseline SMT (N=11)		Baseline SMT improves over EBMT_{TM} (N=5)	
Reason	#	%	#	%
Better word order	5	10.6%	1	2.1%
Better phrase/word selection	5	10.6%	4	8.5%
Better verb translation	2	4.2%	0	0.0%
Less OOV words	2	4.2%	0	0.0%
Other Reasons	0	0.0%	1	2.1%

Table 5.13: Examples of improved translations by EBMT_{TM} system over the baseline SMT system for different reasons.

Reason	English-to-Bangla	English-to-Turkish	English-to-French
Better word order	**Input:** do you not have anything later ? **Reference:** আপানাদের এর পরে কিছু আছে কি ? **SMT:** আপনি কি না anything পরে আছে ? **EBMT_{TM}:** আপানাদের এর পরে কিছু আছে কি ?	**Input:** where is the bus stop for city hall ? **Reference:** belediye binasına gitmek için otobüs durağı nerede ? **SMT:** otobüs durağı nerede şehir hall için ? **EBMT_{TM}:** şehir hall için otobüs durağı nerede ?	**Input:** no dosage adjustment is required in elderly patients based on age alone . **Reference:** pas d ' adaptation posologique sur la seule notion de l ' âge . **SMT:** aucune adaptation posologique n ' est nécessaire chez les patients âgés en fonction de l ' âge seul . **EBMT_{TM}:** chez les sujets âgés : pas d ' adaptation posologique sur la seule notion de l ' âge .
Better phrase/word selection	**Input:** can i make an appointment for my dad ? **Reference:** আমি কি আমার বাবার জন্য একটা অ্যাপয়ন্টমেন্ট করতে পারি ? **SMT:** আমি কি আমার জন্য একটা অ্যাপয়ন্টমেন্ট করতে dad ? **EBMT_{TM}:** আমি কি আমার বাবার জন্য একটা অ্যাপয়ন্টমেন্ট করতে পারি ?	**Input:** my air conditioner is n't working . **Reference:** klimam çalışmıyor . **SMT:** benim hava kremine çalışmıyor . **EBMT_{TM}:** hava kremim düzgün çalışmıyor .	**Input:** pack size of one , in either blistered or non-blistered packaging . single use only . **Reference:** boîte de une , avec ou sans plaquette thermoformée usage unique exclusivement . **SMT:** boîte de 1 , soit sous plaquette ou non-blistered primaire . usage unique seulement . **EBMT_{TM}:** boîte de un . dans conditionnee sous plaquette ou sécurisée emballage . usage unique exclusivement .
Better verb translation	**Input:** when can you bring him in ? **Reference:** আপনি কখন ওনাকে আনতে পারবেন ? **SMT:** আপনি কখন তাকে নিয়ে আসতে পারি ? **EBMT_{TM}:** আপনি কখন তাকে আনতে পারবেন ?	**Input:** is there someplace around here to buy something to drink ? **Reference:** bu civarlarda içecek bir şeyler satın alabileceğim bir yer var mı ? **SMT:** buralarda bir yer var mı almak için bir şeyler içmek ister misin ? **EBMT_{TM}:** içecek bir şeyler alabilecek bir buralarda var mı ?	**Input:** always take invirase / ritonavir exactly as your doctor has told you . **Reference:** respectez toujours la posologie d ' invirase / ritonavir indiquée par votre médecin . **SMT:** toujours prendre invirase / ritonavir exactement comme votre médecin vous l ' a indiqué . **EBMT_{TM}:** respectez toujours la posologie de invirase / ritonavir indiquée par votre médecin .

Table 5.14: Examples of improved translations of the baseline SMT system over the EBMT_{TM} system for different reasons.

Reason	English-to-Bangla	English-to-Turkish	English-to-French
Better word order	**Input:** is it possible to book an appointment later this week ? **Reference:** এই সপ্তাহের শেষের দিকে একটা অ্যাপয়ন্টমেন্ট বক করা সমভব কি ? **SMT:** একটা অ্যাপয়ন্টমেন্ট বুক্ করা সম্ভব এই সপ্তাহে পরে ? **EBMT_{TM}:** পরে এই সপ্তাহে ? একটা অ্যাপয়ন্টমেন্ট বুক্ করা সম্ভব কি ।	**Input:** may i try on this cotton sweater ? **Reference:** bu pamuk süveteri üzerimde deneyebilir miyim ? **SMT:** bu kazak pamuk deneyebilir miyim ? **EBMT_{TM}:** bunu deneyebilir miyim pamuk süveter ?	**Input:** do not remove the device from the package . **Reference:** ne pas retirer le dispositif de l ' emballage . **SMT:** ne retirez pas le stylo de son emballage . **EBMT_{TM}:** ne pas retirer l dispositif de son emballage .
Better phrase/word selection	**Input:** can i cancel my appointment , i'm feeling much better . **Reference:** আমি কি অযপয়নটমেনটটা আমার বাতিল করতে পারি , আমি বেশ ভালো আছি । **SMT:**আমি কি আমার অ্যাপয়ন্টমেন্টটা বাতিল করতে পারি , আমি বেশ ভালো আছি , । **EBMT_{TM}:** আমি কি আমার অ্যাপয়ন্টমেন্টটা বাতিল করতে পারি , আমি করছিলেন আসংখ্য বেশ ভালো আছি ।	**Input:** where is the most famous department store ? **Reference:** en ünlü alışveriş merkezi nerede ? **SMT:** en ünlü alışveriş merkezi nerede ? **EBMT_{TM}:** en ünlü bakanlık haznesi nerede ?	**Input:** in these cases take immediate contact to a doctor . **Reference:** dans ce cas , prendre immédiatement contact avec un médecin . **SMT:** dans ces cas , prenez immédiatement contacter immédiatement un médecin . **EBMT_{TM}:** dans ces cas prendre conditionnements contact à un médecin .
Better verb translation	**Input:** dr finn doesn't work on friday mornings . dr thomas can see you . **Reference:** ডাঃ ফিন শুক্রবার সাকলে কাজ করেন না । ডাঃ থমাস আপনাকে দেখতে পারেন । **SMT:** ডাঃ ফিন কাজ করেন না শুক্রবার সকালটা । ডাঃ thomas আপনাকে দেখতে পারেন । **EBMT_{TM}:** ডাঃ ফিন কাজ করেন না শুক্রবার ও । ডাঃ thomas কি দেখা হবে ।	**Input:** i recommend this wine . **Reference:** bu şarabı tavsiye ederim . **SMT:** bu şarap tavsiye ederim . **EBMT_{TM}:** bunu şarap önerebilir .	**Input:** the duration of treatment should be restricted to the period that corresponds to allergenic exposure . **Reference:** la durée du traitement devra être limitée à la période d ' exposition allergénique . **SMT:** la durée du traitement doit être limité à la période correspondant à une exposition allergenic . **EBMT_{TM}:** la durée du traitement sera la limité à la période correspondant à une exposition allergenic .

5.7 Observations

We found that the EBMT_{TM} system shows a higher accuracy across all metrics (cf. Table 5.2 and Table 5.3) compared to the baseline SMT system for English-to-Bangla translation. This is due to the fact that the English–Bangla training and test examples are homogeneous in nature. Due to the homogeneity a large segment of the input sentences to be translated can be matched with the example-base. This helps to retain the word order in the target translations which would otherwise affect the BLEU score. In contrast, the SMT-based system essentially does not use these matched segments as a whole instead the SMT decoder prefers the most probable translation.

In contrast, with English-to-Turkish and English-to-French experiments, we found that EBMT_{TM} shows a lower accuracy on its own compared to the baseline SMT system. We used moderate sized corpora for these two experiments. For this reason, the SMT system has more evidence to estimate the probability distributions used in the decoding process. In contrast, our system mainly relies on the one sentence that closely matched the input test sentence. Thus, the effect of increased data size is less compared to the SMT system. Our approach is more effective with a small homogeneous corpus. In order to see this effect, we used different size training data for the English-to-Turkish experiments by choosing the closely matched sentences from the whole corpus. Figure 5.4 depicts the BLEU scores obtained with different data sizes. We found that the EBMT_{TM} system on its own has higher BLEU scores than the baseline SMT approach when the amount of training data is less than 5000 sentences. This is due to the fact that the use of a very small amount of training data does not produce a reliable phrase translation model. However, with increased data sizes, SMT performs better compared to the EBMT_{TM} system. However, there remain some sentences which are better translated by the EBMT_{TM} approach compared to SMT, although the overall document translation score is higher with the SMT. Thus, we combined both systems based on different features.

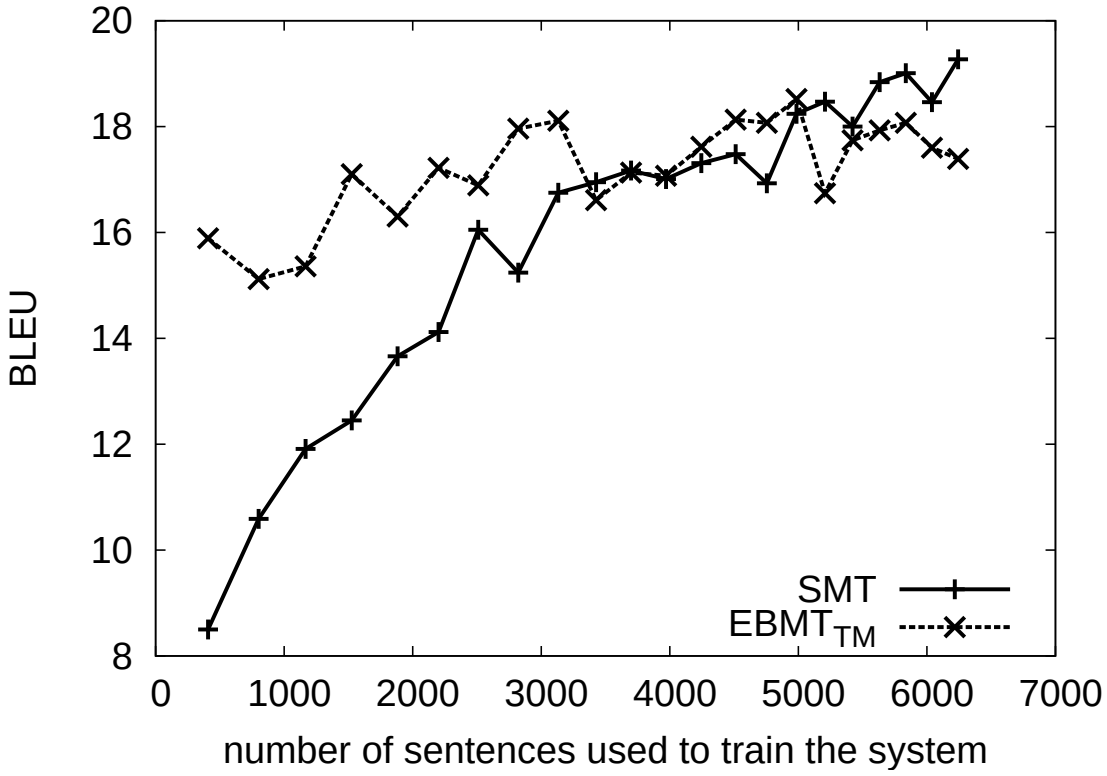
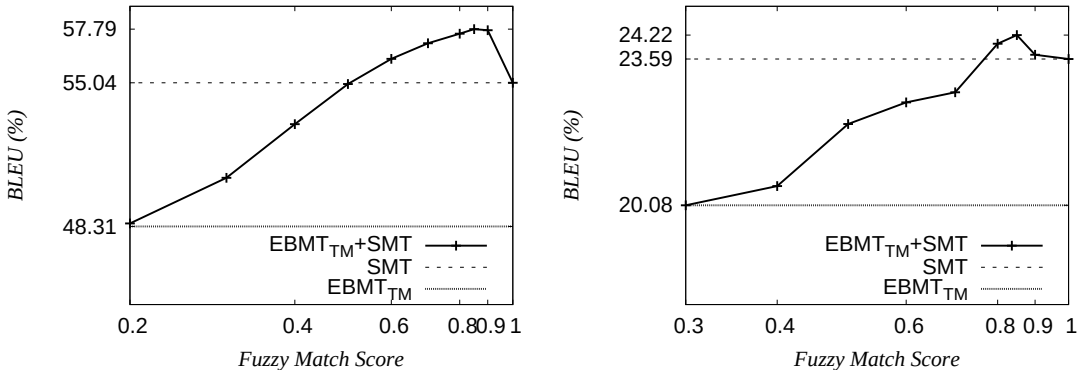


Figure 5.4: BLEU score obtained by two different systems with different data sizes for English-to-Turkish translation.

We found that the combined system ($\text{EBMT}_{\text{TM}} + \text{SMT}$) performs better for the English-to-Turkish and English-to-French experiments compared to the baseline SMT approach. Figure 5.5 shows the effect in the translation quality when different FMS thresholds were used to combine the two systems.

We found that if an input has a high FMS with the example-base, then the EBMT_{TM} system does better than SMT. In particular for the English-to-Turkish experiments, we found that a FMS of over 0.8 shows an improvement over SMT. As the testset is disjoint from the training set, no sentences have a FMS of 1.0. Thus, the $\text{EBMT}_{\text{TM}} + \text{SMT}$ and SMT systems have the same translation score when the FMS equals 1.0, as shown in Figure 5.5. We found more gains using the combined $\text{EBMT}_{\text{TM}} + \text{SMT}$ system in translation score for English-to-French compared to the English-to-Turkish translation. This is due to the fact that the English-to-French system has a relatively larger percentage of sentences with a high FMS. For example,



(a) English-to-French (b) English-to-Turkish

Figure 5.5: Effect of FMS in the combined $\text{EBMT}_{\text{TM}} + \text{SMT}$ system.

the number of test sentences with matches from the example-base of FMS>0.85, respectively, for English-to-Turkish and English-to-French systems are 35 (8.5%) and 3323 (33.23%).

However, FMS may not be the only factor for triggering the EBMT_{TM} system (Marcu, 2001b), we also consider the EqUs factor. Though an FMS over 0.7 shows no improvement in overall system accuracy for English-to-Turkish translation, inclusion of the EqUs along with FMS does show improvement. Thus the EBMT_{TM} approach is more effective if the number of non-matched segments in the source and the target is equal. With our English-to-French experiments, we have found that the combined system does better compared to SMT when the FMS is over 0.6. However, the inclusion of EqUs does not have much impact in the translation score. This feature shows a small drop in BLEU score when FMS is over 0.8 but shows improvement with lower FMS scores (0.5 to 0.7).

5.7.1 Assessment of Error Types

Errors are propagated due to the incorrect selection of source–target equivalences in the phrase table and lexical table which are used as the TUs in our TM. This results in some incorrect alignments in the matching step of our EBMT system. For example, in the sentence shown in (56) from English-to-Bangla translation, the matching module gives the following alignment:

(56) a. s: which doctor <s#0:*do*> you <s#1:*usually*> see ?

b. s_c: which doctor <s#0:*would*> you <s#1:*like to*> see ?

c. t_c: আপনি কোন ডাক্তারকে দেখাতে <s#1: চান> ?

Apani kona DAktArake dekhAte chAna ?

YOU WHICH DOCTOR-accusative TO SEE-causative WANT?

In the above example, the word 'would' does not have any alignment in t_c. The three target equivalents in the TM: হবে (/habe/ [is-3Fr]), বলব (/balaba/ [say-1Fr]) and কি (/ki/ [what]) of the word 'would' do not match with any of the words in t_c. Also, the system suffers when there is a mismatch either in the verb or in the subject of the sentence. This is because the inflection on the verbs depends on the morphological attributes of the subject.

Example (57) shows similar alignment errors for English-to-Turkish translation.

(57) a. s: i have a terrible <*headache*> .

b. s_c: i have a terrible <*cough*> .

c. t_c: berbat bir öksürüğüm var .

In the above example, the word 'cough' does not have any alignment in t_c. Neither of the two target equivalents of the word 'cough' in the TM (*öksürük* (cough) and *öksürük tedavisi için* (for cough treatment)) match any of the words in t_c. The word aligner fails to align any word with 'cough' between the sentences s_c and t_c. Furthermore, the system suffers when there is a mismatch either in the verb or in the subject of the sentence. This is because in Turkish the inflection on the verb depends on the morphological attributes of the subject.

The second type of error is propagated during the recombination step. Consider the English-to-Bangla translation example in (58). We have successfully matched the segments between s_c and t_c.

(58) a. s: i'll call you back <s#0:*in a few minutes*> .

b. s_c: i'll call you back <s#0:*within half and hour*> .

c. t_c: আমি <s#0: আধ ঘন্টা থেকে এক ঘন্টার ভিতরে> আবার কল করব।

Ami Adha ghanTA theke eka ghanTAra bhitare AbAra kala karaba.

I HALF AN HOUR TO ONE HOUR WITHIN AGAIN CALL-future.

However, in the recombination step, we need to generate the translation for the segment 'in a few minutes'. We have found that the portion 'a few minutes' has a translation equivalent 'কয়েক মিনিট দেরিতে (/kaYeka miniTa derite/)' in the TM. Thus, we still need to translate the word 'in' to generate the target equivalent for the whole segment. For the word 'in', the TM has a separate entry with three target equivalents: নিয়ে (/niYe/), নিয়ে আসতে (/niYe Asate/), and আসুন (/Asuna/). Picking the most probable target equivalent for 'in' and combining with the target equivalent of 'a few minutes', we generate 'নিয়ে কয়েক মিনিট দেরিতে (niYe kaYeka miniTa derite)'. This is not a fluent target equivalent because we don't need to translate the word 'in' separately as this has been already been captured in the inflection (তে – te[locative]) of the final word of the target equivalent of 'a few minutes'.

Similar errors are present across languages. Consider the English-to-Turkish translation example in (59).

(59) a. s: i want something <s#0:*with shorter sleeves*> .

b. s_c: i want something <s#0:*to cure headache*> .

c. t_c: <s#0:*baş ağrısını geçiren*> bir şey istiyorum .

In the recombination step, we need to generate the translation for the segment 'with shorter sleeves'. We are unable to find the whole segment in the TM, and moreover none of the bigrams are present in the TM. Thus, we translate each word of the segment one by one which results in an erroneous translation 'birlikte boydan kollu'. The most likely translation of the words 'with' and 'shorter' are 'birlikte' and 'boydan', respectively, in the TM. However, this causes an error in this context as

'boydan' is an incorrect translation for 'shorter', and 'with' is translated to –lu in 'kollu'.

Another common type of error occurs due to the wrong morpho-syntactic alignment and recombination. The effect can be seen in English-to-Turkish translation example in (60):

(60) a. s: do you have a japanese <s#0:*guidebook*> ?

b. s_c: do you have a japanese <s#0:*magazine*> ?

c. t_c: japonca bir <s#0:*derginiz*> var mı ?

The word 'magazine' is matched with 'derginiz' (dergi 'magazine' + possessive ending) but a valid match should point out only the 'dergi' part. The effect is clear when 'guidebook' is translated to 'rehber kitab', the required suffix is missing in the output. Thus, due to the rich morphology of Turkish, many morphosyntactic suffix assignment errors are generated.

5.7.2 Time Complexity

Our particular EBMT_{TM} approach is a type of runtime EBMT. Thus running time is a big concern to make the system scalable with larger example-bases. The alignment step is the most time consuming step in our approach which finds the closest example from the example-base for a given input sentence using edit-distance-based fuzzy match score. The worst-case time complexity of the matching step is $O(nm^2)$, where n denotes the size of the example-base and m is the average length (in words) of a sentence. The worst-case time complexity of both the alignment and the recombination step is $O(m^2)$, where m is the average number of words in a sentence. Thus the total time complexity is $O(nm^2)$.

We measure the real-time taken by our EBMT_{TM} approach for the three different language pairs used in our experiment in a 3GHz Core 2 Duo machine with 4GB RAM. We also estimate the decoding time of the SMT approach. Table 5.15 shows the average running time to translate one sentence using two different systems.

Table 5.15: Average running time (in *seconds*) of the two different systems.

System	Language pairs		
	English-to-Bangla	English-to-Turkish	English-to-French
SMT	0.19	0.34	1.87
EBMT_{TM}	0.01	0.72	13.6

Note that with larger amount of data (English-to-French experiments), the EBMT_{TM} system has a large time complexity. The majority of this time is to compute the edit distance with a large example-base to find the closest match sentence. We will address the issue of time complexity reduction in Chapter 6.

5.8 Summary

The experiments show that the EBMT_{TM} approach works better compared to the SMT-based system when available resources are limited. A combination of EBMT_{TM} and SMT achieves higher scores than the individual systems. Integration of a subsentential TM with the EBMT framework improves the translation quality in our experiments. Our English-to-Bangla experiment shows that EBMT_{TM} has a better accuracy on its own than the baseline SMT system which answers research question RQ1. This effect is also illustrated in the English-to-Turkish experiments when the amount of training data is less than 5000 sentences. The approach uses an auxiliary subsentential TM to translate some of the unmatched portions of the EBMT system. Finally, with a larger amount of training data (English–French), a combination of EBMT_{TM} and SMT has better translation quality than the individual systems which answers research question RQ3. Thus, the approach satisfactorily answers research questions RQ1, RQ2 and RQ3 (cf. Chapter 1).

We have achieved promising results using the EBMT_{TM} system with a moderate size closed-domain corpus. We have seen that the system works well for certain sentences especially those with higher FMS-based similarity to the example-base. Based on these observations, we assume that a similar trend can be found when a larger amounts of training data are available for the language pair. However,

the matching step of the EBMT_{TM} is a time consuming process with a runtime complexity of $O(nm^2)$. This will drastically decrease the throughput of our runtime EBMT_{TM} system using a larger example-base. In order to handle this situation, we address the issue of scalability of the EBMT_{TM} approach in the next chapter.

5.8.1 Contribution

Our main contributions regarding this work are as follows:

- The development of an end-to-end EBMT_{TM} system. This is a novel approach to developing an EBMT system using TMs derived by SMT methods.
- Finding different features (FMS and EqUS) for effectively combining EBMT_{TM} with a state-of-the-art SMT system.

In the next chapter, we explore different methods to make the EBMT_{TM} system scalable at runtime. We use a heuristic-based approach and an information retrieval-based technique to source a potential set of suitable candidate sentences for EBMT matching.

Chapter 6

EBMT_{TM}: Improving Scalability

In Chapter 5 we demonstrated a novel EBMT system using subsentential translation memory. The results presented in that chapter demonstrated clearly how effectively the EBMT_{TM} system can be used for translating homogeneous data in a resource-poor setting. In addition, we also demonstrated how the performance of an SMT system could be improved using the EBMT_{TM} system when translating sentences with greater similarity to the example-base. Our EBMT_{TM} system is a runtime EBMT approach that uses a time-consuming edit-distance-based measure to find closely matched sentences from the example-base. We have seen that the approach suffers from the significant time complexity issues of a runtime approach even with a moderate sized example-base.

In this chapter, we address the issue of scalability of our runtime EBMT_{TM} approach. First, we use a heuristic-based approach which is often useful to avoid some of the computation. Furthermore, we used an IR-based indexing and retrieval technique to speed up the time-consuming matching procedure of the EBMT system.

The organization of the chapter is as follows. We first describe the motivation for using different approaches to improve the scalability of the system. We then describe two different approaches we are using to improve the runtime performance of the EBMT system. Then we present the results with our observations from different experiments conducted in this chapter.

6.1 Motivation

Translation quality and speed are two important concerns when developing an MT system. While translation quality is important in all application areas of MT, translation speed has a role to play in real time applications, e.g. online chat translations. The main motivation for scalability is to improve both the quality and speed of the EBMT system when using a large example-base. The matching procedure in an EBMT system finds the example (or a set of examples) which most closely match the source-language string to be translated. All matching processes necessarily involve a distance or similarity measure. The most widely used distance measure in EBMT matching is Levenshtein distance (Levenshtein, 1965; Wagner and Fischer, 1974) which has quadratic time complexity. This is quite time-consuming even when a moderate amount of training examples are used for the matching procedure. However, Ukkonen (1983) gave an algorithm for computing edit-distance with the worst-case time complexity $O(md)$, where m is the length of the string and d is its edit-distance. This is effective when $m \gg d$. We use word-based edit-distance, so m is shorter in length.

Runtime EBMT approaches generally do not include any training stage, which has the advantage of not having to depend on time-consuming preprocessing. On the other hand, their runtime complexity can be considerable. This is due to the time-consuming matching stage taking place at runtime. In our EBMT_{TM} system, we find the closest matching sentences at runtime from the whole example-base for a given input sentence using the edit-distance matching score. In the previous chapter, we showed that the matching step of the EBMT_{TM} system is a time-consuming process with a runtime complexity of $O(nm^2)$, where n denotes the size of the example-base and m denotes the average length (in words) of a sentence. Due to a significant runtime complexity, the EBMT_{TM} system can only handle a moderate size example-base in the matching stage. However, it is important to handle a large example-base for scalability and to improve the quality of an MT system.

It is often the case that we do not need to compute the time consuming edit-distance score for all the examples in the example-base. In order to tackle this problem, we propose the use of heuristics. They also help to avoid some of the unnecessary computations. Heuristics can be used to extract a potential set of candidate sentences from the example-base that are likely to contain the closest matching sentence to the input sentence to be translated.

6.2 Approach

We adopt two approaches for finding the closest matching sentences efficiently in order to make the system scalable. First we use an heuristic-based solution. Secondly, we use an IR-based indexing technique to speed up the time-consuming matching procedure of the EBMT_{TM} system.

6.2.1 Grouping

In order to discard some of the edit-distance computation, we rely on the hypothesis that the input sentence (s) and its closest match sentence (s_c) from the example-base are likely to have a similar sentence length. We use the following heuristic to reduce the effort wasted on computing edit-distances with some of the example sentences which are unlikely to be a close match sentence for an input test sentence:

The input sentence (s) and its corresponding closest match sentence (s_c) from an example-base should have comparable sentence lengths.

In order to do that, we divide the example-base into *bins* based on sentence length. It is anticipated that the sentence from the example-base that most closely matches the input sentence will fall into the group which has comparable length to the length of the input sentence. First, we divide the example-base E into different bins based on their word-level length $E = \bigcup_{i=1}^{l} E_i$ and $E_i \bigcap E_j = \emptyset$ for all $i \neq j$ where $0 \leq i, j \leq l$. E_i denotes the set of sentences with length i and l is the maximum length of a sentence in E. In order to find the closest match for a test sentence (s

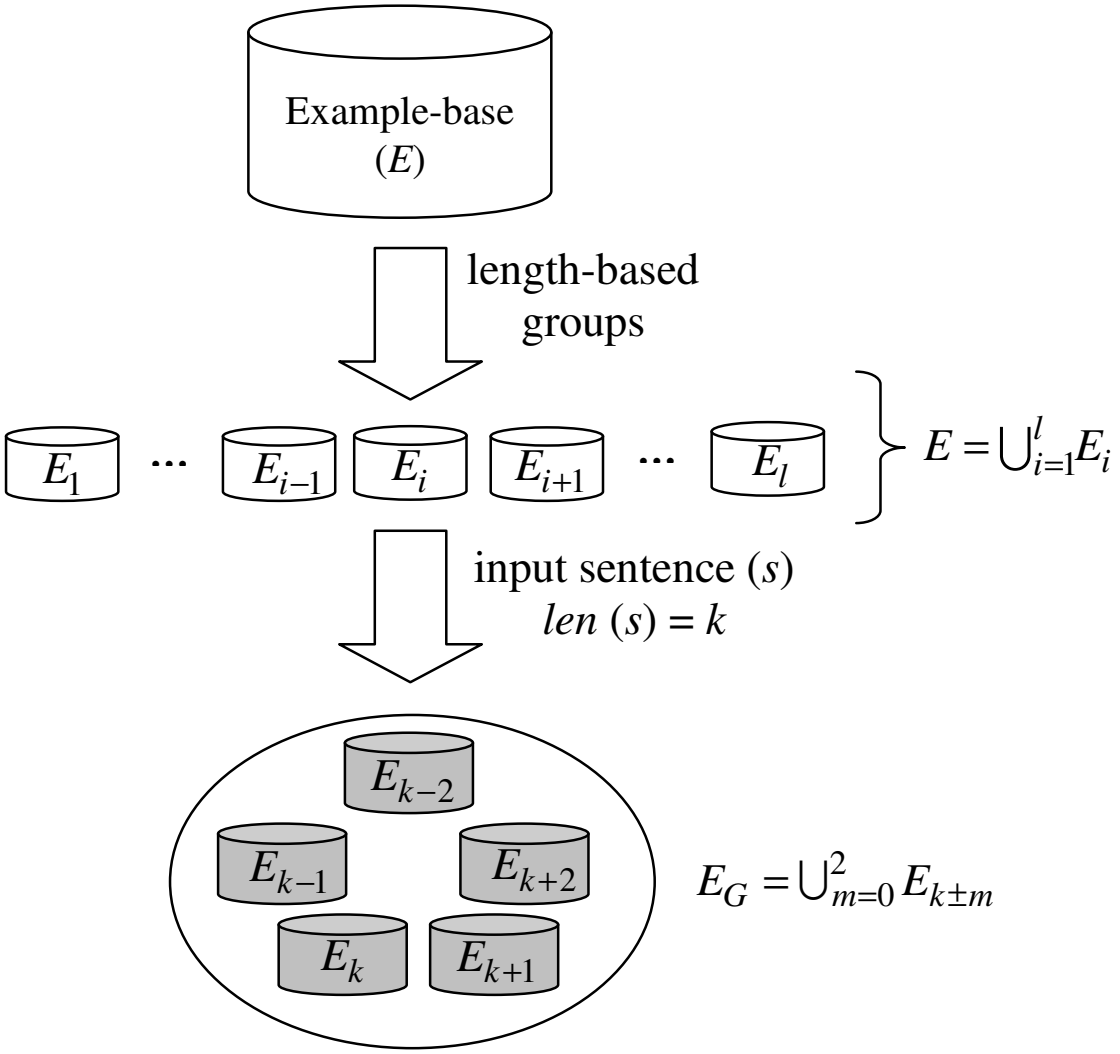


Figure 6.1: Length-based selection of potential set of candidate examples to find the closest match.

of length k), we only consider examples $E_G = \bigcup_{m=0}^{x} E_{k\pm m}$, where x indicates the maximum window size. This is shown in Figure 6.1. In our experiment, we consider the value of m from 0 to 2. Furthermore, we find the closest-match sentence s_c from E_G for a given test sentence s using the edit-distance measure. E_G has fewer sentences compared to E which will effectively reduce the time of the matching procedure.

6.2.2 Indexing

Our second approach to addressing time complexity is to use indexing. Search engine indexing is an effective way of storing data for fast and accurate retrieval of information (Manning et al., 2008b). During retrieval a set of documents is extracted based on their similarity to the input query. We use this concept to efficiently retrieve a potential set of suitable candidate sentences from the example-base for finding the closest matching sentence. We index the entire source-side example-base using an open-source IR-engine SMART[1] and retrieve the potential set of candidate sentences (likely to contain the closest match sentence) from the example base. Unigrams extracted from the sentences of the example-base are indexed using a language model (LM) and complete sentences are considered as retrievable units. In LM-based retrieval we assume that a given query is generated from a unigram document language model. The application of a LM retrieval model in our case returns a sorted list of sentences from the example-base ordered by the estimated probabilities of generating the given input sentence.

Figure 6.2 provides the detailed architecture of the proposed work flow of the IR-engine integrated EBMT_{TM} system. In order to improve the run-time performance, we integrate the SMART retrieval engine with the matching procedure of our EBMT_{TM} system. To do this, we index the source side of the example-base using the SMART IR-engine to retrieve the candidate close-matching sentences. The input sentence s is considered to be the query to the IR-engine. The retrieval engine estimates a potential set of candidate close-matching sentences from the example-base E for a test sentence s. Based on the retrieved candidate examples, we extract a set of source–target example pairs for the given query. We assume that the closest source-side match s_c of the input sentence s can take the value from $E_{IR}(s)$, where $E_{IR}(s)$ is the potential set of close-matching sentences computed by the LM-based retrieval engine. We have used the top 50 candidate sentences from $\text{E}_{\text{IR}}(\text{s})$ in our

[1]SMART stands for System for Mechanical Analysis and Retrieval of Text. An open source information retrieval system from Cornell University. `ftp://ftp.cs.cornell.edu/pub/smart/`

current experimental setup.[2] Since the IR engine tries to retrieve the document (sentences from E) for a given query (input) sentence, it is likely to retrieve the closest match sentence s_c in the set $E_{IR}(s)$. Due to a much reduced set of possibilities, this approach is anticipated to improve the run-time performance of the EBMT system without hampering system accuracy.

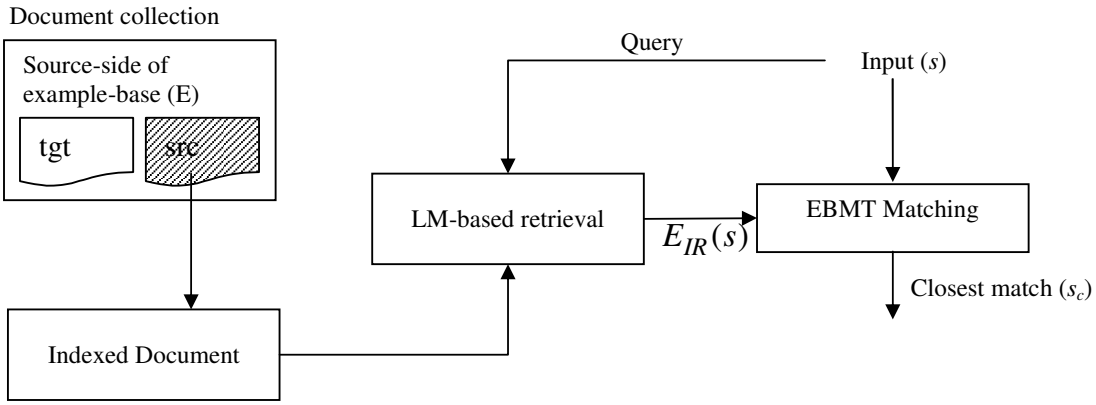


Figure 6.2: Detailed workflow of the IR-engine integrated EBMT_{TM} system.

6.2.3 IR Engine

In this section we describe the working principle of the retrieval model of the SMART IR-engine, namely the process of finding candidate close matching sentences.

Let D be the document collection containing a finite number of points $\{d_1, d_2, ..., d_n\}$, each referring to an actual source-side sentence in the example-base (E). q is an input query containing a finite number of points $\{t_1, t_2, ..., t_m\}$, each referring to an actual word in the input test sentence (s). Our retrieval method is based on the LM approach proposed by Hiemstra (2001). In an LM-based IR model, a document $d \in D$ is ranked by a linear combination of estimated probabilities $P(t_i|d)$ of generating a query term t_i from the document d, and $P(t_i)$ of generating a query

[2]We assume that the set of the top-50 IR-based retrieved candidates is large enough to contain the baseline edit-distance-based closest matching example. However, a smaller or a larger set of the retrieved candidate may hold the optimal solution. We plan to explore the trade-off between the set of candidates $E_{IR}(s)$ and translation scores in future work with a varying size of $E_{IR}(s)$.

term from the collection. The document is modeled to choose $q = t_1, t_2, ..., t_m$ as a sequence of independent words as proposed by Hiemstra (2001).

$$P(q|d) = P(d) \prod_{i=1}^{m} \lambda_i P(t_i|d) + (1 - \lambda_i) P(t_i) \tag{6.1}$$

$$logP(q|d) = logP(d) + \sum_{i=1}^{m} log(1 + \frac{\lambda_i}{1 - \lambda_i} \frac{P(t_i|d)}{P(t_i)}) \tag{6.2}$$

λ_i is the Jelineck-Mercer smoothing parameter (Jelinek and Mercer, 1980). $P(q|d)$ is the prior probability of the relevance of a document d. The term weighting equation in (6.2) can be derived from Equation (6.1) by dividing both sides by $(1 - \lambda_i)P(t_i)$[3] and taking the logarithm on both sides so as to convert a product into an addition. This transformation also ensures that the computed similarities between a document and a given query is always positive. We index each query vector q as $q_k = tf(t_k)$ and each document vector d as $d_i = log(1 + \frac{\lambda_i}{1-\lambda_i} \frac{P(t_i|d)}{P(t_i)})$, so that their dot product $d.q$ gives the likelihood of generating q from d and hence can be used as a similarity score to rank all the documents.

In Equation (6.1), λ_i is the probability of choosing the ith query term from the document d, whereas $(1 - \lambda_i)$ is the probability of choosing the term from the collection. In our particular experiment, we will assign more weight to a document (a sentence from the example-base) that has more terms matching the query (the input test sentence). Hiemstra (2001) suggests that a high value of λ_i is indicative of an implicit conjunction of query terms, i.e. it supports coordination-level ranking. Since our objective is to retrieve sentences with maximum term overlap to the query sentence, we use a high value of λ_i to enforce an implicit conjunction of query terms.

Hiemstra (2001) found that the performance of an IR system does not vary significantly over small discrete ranges within the choice of parameter λ_i. As per the requirement of maximum term overlap, we choose λ_i in the high range close to 1. More precisely, for all our IR experiments, we used the setting of $\lambda_i = \lambda = 0.9, \forall i$ which was chosen empirically.

[3] $(1 - \lambda_i)P(t_i)$ is a collection-level statistics and does not depend on d.

Time Complexity

The LM-based retrieval uses an inverted indexed list to extract the candidate document from the document collection. The inverted list contains a list of references to documents for each word. In particular, in our task, a list is maintained that contains a mapping for all the unique words and their associated sentence of occurrence. For a given query (input sentence), we need to search the associated list for all the words in the input query in order to retrieve the candidate set of document (sentences). The worst case runtime of the retrieval component is $O(\sum_{\forall w_i} s_i)$, where w_i is a word in the input sentence and s_i is the number of sentences in the example-base that contain w_i. This can be of the maximum of $O(nm)$, where n is the number of documents (sentences in the example-base) and m is the number of words in the input query. This is possible if and only if each individual word in the input string occurs in every sentence of the example-base. The very fact that a query term occurs in every sentence of an example-base, is highly unlikely because both the query and the sentences in the example-base are natural language sentences. The only exceptions are stop-words (e.g articles and prepositions) which occur in a large number of sentences in the example-base. Thus, finding the potential set of candidate sentences is much faster ($O(\sum_{\forall w_i} s_i)$) than traditional edit-distance-based retrieval ($O(nm^2)$) on the full example-base.

6.3 Experiments

We conduct two different experiments to test the scalability of our EBMT_{TM} system.

- $\textbf{EBMT}_{\textbf{TM}} + \boldsymbol{group}_i$: First, we conduct an experiment using the sentence-length-based grouping as described in Section 6.2.1. We refer to this system with $+group_i$, where i indicates the window size while comparing the length of the input sentence with the bins. In our experiment, we consider the value of i from 0 to 2 for finding the closest match sentence. This indicates that we are considering those bins which have at most a length difference of 2

words between an example sentence and the input sentence to be translated. Furthermore, we conducted an experiment using this group-based heuristic in our combined SMT-EBMT system ($\text{EBMT}_{\text{TM}} + \text{SMT}$). We refer to this system as $\mathbf{EBMT_{TM}} + \mathbf{SMT} + group_i$.

- $\mathbf{EBMT_{TM}} + \boldsymbol{index}$: We conduct the second experiment based on the LM-based indexing technique (Section 6.2.2) to retrieve a potential set of candidate sentences from the indexed example-base. We refer to this system with the suffix $+\boldsymbol{index}$. We also conduct an experiment using the IR-based retrieval of closest match sentences with the combined ($\text{EBMT}_{\text{TM}} + \text{SMT}$) systems. We refer to this system as $\mathbf{EBMT_{TM}} + \mathbf{SMT} + index$.

Note that the baseline score for these experiments is the accuracy obtained by the EBMT_{TM} system described in the previous chapter. The EBMT_{TM} system finds the closest match sentence by computing fuzzy match scores for all the sentences in the example-base. The main goal of our current experiments is to improve the running time of the EBMT_{TM} system without affecting the accuracy. Thus, we consider the accuracy reported with the EBMT_{TM} system in the previous chapter as the baseline for both running-time and system accuracy.

6.3.1 Data Used for Experiments

We used three different data sets for our experiments. Two of these data sets are those we used in the previous chapter in order to compare our scalability results with the baseline EBMT_{TM} system.

- The first data set is the IWSLT09 English–Turkish data consisting of 19,972 training examples and a disjoint test set of 414 sentences from IWSLT09 development set. Note that the average length of the sentences in the source- and target-side training data are 9.5 and 6.9 words respectively.

- The second data set is the English–French data from the EMEA corpus. The English–French training data consists of 250,806 unique parallel sentences. As

a test set we use a set of 10,000 randomly drawn sentences disjoint from the training corpus. As noted in the previous chapter, this corpus represents a particular domain with relatively longer sentences compared to the English–Turkish data. The average length of the sentences in the source- and target-side training data are 18.8 and 22.61 respectively.

These two data sets represent a small and moderate-sized example-base for two different languages. We conduct all our scalability experiments with these data sets and compare the results with baseline SMT and EBMT_{TM} systems.

Our third and larger data set, is from the JRC-acquis (Steinberger et al., 2006)[4] multilingual English–French parallel corpus. This corpus also belongs to a single domain, containing European Union legal documents. Note, that this corpus is automatically crawled from websites and automatically aligned using HunAlign (Varga et al., 2005). The corpus consists of 753,323 parallel examples for training. We used a set of randomly drawn 2,000 disjoint sentences for testing. This corpus comprises of relatively longer sentences compared to the IWSLT09 English–Turkish and English–French EMEA data sets. The average length (in words) of the sentences on the source- and target-side training data are 23.84 and 25.67 respectively.

6.3.2 Results

We measure both the translation time and accuracy with the two approaches described in Section 6.2 to improve the scalability of the EBMT system on 3 different data sets. All the experiments were performed on a 3GHz Core2 Duo machine with 4GB RAM.

Table 6.1 shows the running time of two systems ($\text{EBMT}_{\text{TM}} + \text{SMT} + \textit{group}_i$ and $\text{EBMT}_{\text{TM}} + \text{SMT} + \textit{index}$) and compares the runtimes with two baseline systems (SMT and EBMT_{TM}) for the moderate-sized data sets. Note that the runtime for the $\text{EBMT}_{\text{TM}} + \text{SMT} + \textit{index}$ system includes the retrieval time along with the

[4]http://optima.jrc.it/Acquis/

Table 6.1: Average running time (in *seconds*) of different systems with English–Turkish IWSLT09 and English–French EMEA data sets.

System	Data set	
	English-to-Turkish IWSLT09	English-to-French EMEA
SMT	0.34	1.86
EBMT_{TM}	0.72	13.60
$\text{EBMT}_{\text{TM}} + group_0$	0.08	0.37
$\text{EBMT}_{\text{TM}} + group_1$	0.23	1.09
$\text{EBMT}_{\text{TM}} + group_2$	0.36	1.81
$\text{EBMT}_{\text{TM}} + index$	0.014	0.029

translation time of the three stages of the EBMT_{TM} system. However, the indexing time is not included here as it is a one-time preprocessing of the example-base. The time taken to index the source English sentences is 3 and 24 seconds, respectively, for English–Turkish IWSLT09 and English–French EMEA data sets. The indexing time for the source English sentences of the English–French JRC-acquis data is 165 seconds.

The above table shows that both the grouping and indexing methodologies proved successful for system scalability. Note that the SMT decoder takes on average 0.34 seconds and 1.86 seconds, respectively, to translate each English sentence for the English–Turkish and English–French test sets. In contrast, the baseline EBMT_{TM} system takes a longer average translation time per sentence of 0.72 seconds and 13.6 seconds respectively. The fastest translation time was 0.014 seconds and 0.029 seconds per sentence when using indexing, respectively, for the English–Turkish IWSLT09 and English–French EMEA data sets.

We also need to estimate the accuracy while combining group-based and index-based techniques with the baseline system (EBMT_{TM}) to understand their relative performance. We present the system accuracy of the EBMT-SMT combined systems (EBMT_{TM} +SMT+$group_i$ and $\text{EBMT}_{\text{TM}} + \text{SMT} + index$) using the grouping and indexing techniques. This is due to the fact that the combined system has better accuracy than the individual system. The baseline for these experiments is $\text{EBMT}_{\text{TM}} + \text{SMT}$ when no indexing/grouping is applied and finding the closest

Table 6.2: BLEU scores for the three different systems for English-to-Turkish and English-to-French under different conditions. i denotes the number of bins considered during grouping.

Condition	System				
	EBMT_{TM} +SMT	EBMT_{TM} +SMT + $group_i$			EBMT_{TM} +SMT + *index*
		i=0	i=±1	i=±2	
English-to-Turkish (IWSLT09)					
FMS>0.85	24.22	24.18	24.18	24.23	24.24
FMS>0.80 || (FMS>0.70 & EqUS)	23.87	23.34	23.90	24.40	24.37
FMS>0.85 & EqUS	**24.41**	**24.17**	**24.38**	**24.34**	**24.39**
English-to-French (EMEA)					
FMS>0.85	**57.79**	**56.47**	**57.48**	**57.76**	**57.92**
FMS>0.80 || (FMS>0.70 & EqUS)	57.46	55.69	57.07	57.33	57.56
FMS>0.85 & EqUS	57.24	56.48	57.23	57.29	57.32

sentences fully relies on fuzzy-matched-based selection.

Table 6.2 provides the system accuracy scores using the grouping and indexing techniques for the combined system with the highest performing features for two different data sets (English–Turkish IWSLT09 and English–French EMEA). We report the translation quality under three conditions. Similar trends have been observed for other conditions.

The results shows that the translation accuracy remains unchanged or sometimes increases with the use of indexing. A similar effect has been observed with the grouping heuristic when a considerable number of bins (i=±2) were used for finding the closest matching sentence. Though the use of the grouping heuristic (i=±2) does not affect the system accuracy, the use of a large number of bins does not improve the running time either (cf. Table 6.1).

English-to-French translation using JRC-acquis corpora

Based on the success obtained by the LM-based retrieval to improve the scalability of our EBMT_{TM} system, we conducted an additional experiment using the larger English–French JRC-acquis data. Based on the results obtained using moderate-sized data, we used the third data set only with EBMT_{TM} + SMT + *index* experiments and compare the results with the baseline SMT system. This is because with the third data set we want to ensure that the system is capable of handling a large

Table 6.3: System accuracies of the EBMT_{TM} + SMT + *index* system with different combining factors using English–French JRC-acquis data.

System: EBMT_{TM} + SMT			
Condition	**times** EBMT_{TM} **used**	**BLEU (in %)**	**NIST**
Baseline SMT: BLEU=57.97% and NIST=11.12			
FMS>0.85	395 (19.8%)	**59.57**	**11.27**
FMS>0.80 || (FMS>0.70 & EqUS)	571 (28.6%)	59.56	11.27
FMS>0.85 & EqUS	226 (11.3%)	58.70	11.19

example-base using LM-based retrieval. Due to the significant time complexity of the baseline EBMT_{TM} system we need an alternative way (EBMT_{TM} + SMT + *index*) to handle larger example-bases in our translation framework. Thus, we conduct this experiment to show that the alternative LM-based retrieval technique makes the system scalable without affecting translation quality. In addition, we also wanted to show that while using a large dataset with the help of the LM-based retrieval the system might produce a better translation for certain sentences compared to the baseline SMT approach.

The average time taken to translate each sentence using the EBMT_{TM} + SMT + *index* system is 5.89 seconds (using the constraint FMS > 0.85), where the baseline SMT system takes 7.11 seconds. Table 6.3 shows the accuracy of the EBMT_{TM} + SMT + *index* system under different conditions. We found that the combined system (EBMT_{TM} + SMT + *index*) using the indexing technique shows an improvement (1.6 absolute BLEU points) over the baseline SMT system. The improvement with the combined system over the baseline SMT system is statistically significant[5] (reliability of 98%). A similar trend has been observed with the NIST evaluation metric with an improvement of 0.15 absolute points over the baseline SMT system.

[5]Statistical significance tests were performed using paired-bootstrap resampling (Koehn, 2004).

6.4 Observations and Discussions

We have seen in the previous chapter that the use of our EBMT_{TM} approach is effective in terms of translation quality. However, we found that like other runtime EBMT approaches, the EBMT_{TM} system also has a considerable runtime complexity. In order to translate one sentence from English into Turkish using an example-base of 19,972 sentence pairs, the basic EBMT_{TM} system takes on average 0.72 seconds. The situation changes when using the large example-base (≈250k sentence pairs) for English into French translation. Here, we found that the EBMT_{TM} system takes an average of 13.6 seconds to translate one source English sentence into French. This is a significant amount of time for one sentence by any standard for a runtime approach. However, both the grouping and indexing of examples reduce the time complexity of the approach effectively.

The time reduction with grouping depends on the number of bins considered to find the closest sentence during the matching stage. Systems with a lower number of bins take less time but cause more of a drop in translation quality. The average time taken to translate an English sentence to Turkish takes 0.08 and 0.36 seconds, respectively, when using one ($i = 0$) and five ($i = \pm 2$) nearest bins. The use of a single bin causes a drop of 0.24 absolute BLEU points (highlighted in Table 6.2) but the translation quality remains the same with the use of five bins. The effect is more prominent with the English-to-French system in Table 6.2 that uses a comparatively large example-base. We found a drop of 1.32 absolute BLEU points while considering a single bin whose length is equal to the length of the test sentence. This configurations takes an average of 0.37 seconds to translate one English sentence into French. However, the BLEU score barely changes (a drop of 0.03 absolute BLEU points) when considering 5 nearest bins (± 2) to find the closest match for a given test sentence. Nevertheless, there is not much of a reduction in translation quality but it increases the average translation time to 1.81 seconds for the translation of an English sentence into French. Thus, the group-based method is not effective enough

to balance system accuracy and translation time with a large example-base.

Incorporation of the index-based retrieval technique into the matching stage of the EBMT_{TM} system has the highest efficiency gains in runtime. The average time taken to translate an English sentence into Turkish is 0.014 seconds. Translating each English sentence into French takes an average of only 0.029 seconds. However, the use of IR-based retrieval introduces the preprocessing indexing stage within the framework. This preprocessing stage of indexing the corpus takes only a small amount of time. As noted earlier, the time taken to index the source side of the English–Turkish IWSLT09 (≈20k sentences), English–French EMEA corpus (≈250k sentences) and English–French JRC-acquis corpus (≈750k sentences) are 3, 24 and 165 seconds, respectively. Thus, the IR-based approach only involves a time efficient preprocessing stage. It is also interesting to note that with indexing, the BLEU score remained the same or even increased. This is due to the fact that, compared to FMS-based matching, a different closest matching sentence (s_c) is selected for some of the input sentences while using index-based retrieval, thus resulting in a different translation outcome. Figure 6.3 compares the number of times the $\text{EBMT}_{\text{TM}} + \text{SMT} + \textit{index}$ is used in the hybrid system and the number of times both the $\text{EBMT}_{\text{TM}} + \text{SMT} + \textit{index}$ and $\text{EBMT}_{\text{TM}} + \text{SMT}$ system select the same closest matching sentences for English-to-Turkish translation.

The use of index-based candidate selection for EBMT matching shows effective improvement in time taken, and BLEU scores remained the same or increased. Due to the selection of a different closest-matching sentence s_c, sometimes the system produces a better quality translation which increases the system level BLEU score. Table 6.4 shows such examples for English-to-Turkish and English-to-French (using EMEA corpus) translation where an index-based technique produced a better translation than the baseline ($\text{EBMT}_{\text{TM}} + \text{SMT}$) system. In the English-to-Turkish translation example, both the baseline EBMT_{TM} and the index-based $\text{EBMT}_{\text{TM}} + \text{SMT}+$ *index* find the closest match (s_c) that has a single word difference with the input (s). However, due to the different skeleton translation (s_t) corresponding to

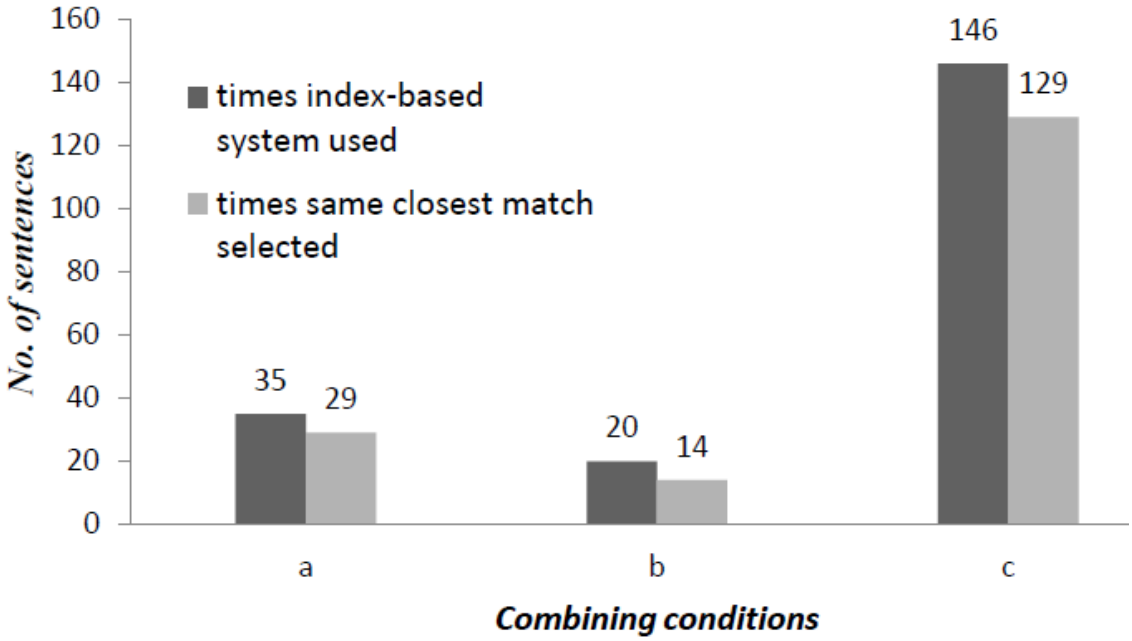


Figure 6.3: Number of times $\text{EBMT}_{\text{TM}} + \text{SMT} + \textit{index}$ used in the hybrid system and the number of times the same closest-matching sentences are selected by the systems. a=FMS>0.85, b=FMS>0.85 & EqUS and c=FMS>0.80 OR (FMS>0.70 & EqUS).

a different s_c, the resulting outputs of the systems sometimes differ considerably. In Table 6.4, the translation produced by the baseline EBMT_{TM} system has no word common with the reference translation while the translation produced by the $\text{EBMT}_{\text{TM}} + \text{SMT} + \textit{index}$ system has two words in common with the reference translation. This is why the system-level BLEU score sometimes increases with the index-based system compared to the baseline EBMT_{TM} system.

6.5 Summary

In this chapter, we have addressed the issue of scalability of our EBMT_{TM} approach. Our baseline EBMT_{TM} system is a runtime approach which has high time complexity when using a large example-base. We have proposed two different solutions to improve the scalability of the system. We have seen from our experiments that the solution based on the grouping heuristic effectively improves the running time with

Table 6.4: The effect of indexing in selection of s_c and in final translation.

	English-to-Turkish	**English-to-French**
Input(s):	where can i buy accessories	zeffix belongs to a group of medicines called antivirals .
Ref:	**nereden aksesuar alabilirim**	**zeffix appartient à une classe de médicaments appelés antiviraux .**
	Baseline EBMT$_{\mathbf{TM}}$ system	
s_c:	where can i buy *plates*	*simulect* belongs to a group of medicines called *immunosuppressants* .
s_t:	*tabak* almak istiyorum	*simulect* fait parti d ' une classe de médicaments appelés *immunosuppresseurs* .
Output:	**aksesuarı almak istiyorum**	**zeffix fait parti d ' une classe de médicaments appelés antiviraux .**
	EBMT$_{\mathbf{TM}}$ + *index* system	
s_c:	where can i buy *stockings*	*diacomit* belongs to a group of medicines called *antiepileptics* .
s_t:	nereden *çorap* satın alabilirim	*diacomit* appartient à un groupe de médicaments appelés *antiépileptiques* .
Output:	**nereden aksesuarı satın alabilirim**	**zeffix appartient à un groupe de médicaments appelés antiviraux .**

a relatively small-sized example-base (e.g. English–Turkish). Also the grouping-heuristic does not hamper the translation accuracy when more number of bins are explored to find the closest match sentence (e.g. five bins with our English-to-French experiment). However, considering fewer bins for a large example-base affects the translation accuracy while there is an improvement in runtime over the baseline system. Thus the grouping heuristic is not an effective solution to balance translation quality and throughput of the system.

In the second solution, we used an IR technique to find the closest match sentence from the example-base. Other systems have used inverted indices and suffix array variants to support retrieving examples in runtime. Brown (1996) indexed the source-language sentences in the Pangloss EBMT system. Suffix arrays provide an efficient data structure for accessing an arbitrary sequence of strings within a large corpus (Yamamoto and Church, 2001). The search algorithm has a worst-case runtime complexity of $O(m \log n)$, where n is the number of tokens in the index and m is the length of the phrase being looked up. The concept of suffix array-based data structures is becoming popular in the area of MT as evidenced by the work

of Brown (2004), Callison-Burch et al. (2005), Zhang and Vogel (2005) and Lopez (2008). Cunei (Phillips, 2012) uses an extension of the traditional suffix array to include position information to support retrieving translations at runtime.

In contrast to the search problem in these approaches, in our particular work, we need to model distance-based approximate string matching (i.e. fuzzy match score (Sikes, 2007)) to retrieve the closest possible match from the example-base. This can be done using the traditional suffix array-based data structure. However, approximate string matching using suffix arrays has a worst-case runtime complexity of $O((ms)^{d+1} + M)$, where m is the length of the input string (in words), s is the vocabulary size of the example-base, d is the edit distance and M is the number of matches. This is much higher than the runtime complexity of the index-based retrieval ($O(\sum_{\forall w_i} s_i)$, p.144) of the closest possible match from the example-base.

We found that the integration of an LM-based approach retrieval substantially improves runtime without affecting translation accuracy. We have tested our system with moderate and large example-bases. The IR-based solution always shows significant improvement in runtime. Interestingly, the IR-based solution sometimes shows a small improvement in translation quality over the baseline EBMT_{TM} system due to the selection of different closest-matching sentences to produce the skeleton translation. Thus, the approach satisfactorily answers research question RQ4 that addresses the issue of scaling up the EBMT_{TM} system to larger amounts of training data.

6.5.1 Contribution

The main contribution of the work described in this chapter is the integration of IR-based indexing and retrieval step in the flow of our EBMT_{TM} system to make the system scalable at runtime. A significant amount of work has been done in EBMT and in IR system development under separate threads. There is also work (Hildebrand et al., 2005) that links IR-based technology with SMT in the area of translation model adaptation to produce better quality translations. However, no work

has been done combining these two technologies to improve the efficiency in the matching phase of a runtime EBMT system. We investigate the effective use of integrating IR technology in a runtime EBMT system to avoid the drawback of time consuming edit distance calculation and yields the scope of integrating IR-based retrieval technique in a CAT system to find closely matching examples from a TM database.

Chapter 7

Conclusion

In this thesis, we have explored the effects of using different EBMT methods to overcome some of the difficulties encountered with SMT when translating homogeneous data in a resource-poor setting. The experiments in this thesis show that the EBMT approaches work better when compared to the SMT-based system for certain sentences, particularly when the amount of available resources is limited.

First, we adopted two alternative approaches (a pure and a compiled approach) to EBMT to tackle some of the problems of SMT. Both approaches have shown difficulties when used in standalone systems to produce good quality MT output. We also presented different ways to improve the output quality by combining the EBMT approaches with the SMT system which we have shown to be successful in most of the experiments we have conducted. Furthermore, we have developed a novel EBMT system using subsentential TM. Integration of subsentential TM with EBMT shows an improvement when the amount of available resources is limited. In addition, this integrated approach has the highest improvement in translation quality when combined with an SMT system and can effectively handle large amounts of training examples.

At the start of this thesis, in Chapter 2, we reviewed SMT and EBMT, the two paradigms of interest in our work and outlined suggestions to mitigate the problems of SMT using EBMT. We observed the strengths of different EBMT systems and

considered the possibility of producing reliable translations with limited amounts of homogeneous data. We also discussed the TM paradigm and existing research that attempts to integrate MT and TM to produce automatic high quality end-to-end translations. Based on this observation, we decided to investigate different EBMT approaches to develop a reasonably good quality MT system based on limited amounts of data.

In Chapter 3, we implemented a proportional-analogy-based EBMT system using the approach of Lepage and Denoual (2005a) who found that it performed very well on data from the IWSLT04 competition. Looking into the nature of the IWSLT04 data (short sentences from the homogeneous BTEC corpus), we anticipated that the PA-based approach would be effective for translating homogeneous data with limited resources. However, the approach performs badly in some of our experiments (English-to-Bangla). We found that the PA-based approach suffers from low recall compared to SMT, since the PA-based approach is unable to find any solution in many cases. We implemented different heuristics from the literature and proposed an additional novel heuristic to improve recall. Finally, we showed that a combination of both EBMT and SMT can achieve reasonably good improvements over the individual systems for the NE transliteration task and for the English-to-Chinese MT task.

In Chapter 4, we explored a generalized translation-template-based EBMT technique (Cicekli and Güvenir, 2001) and the system has shown a similar trend to the PA-based system in terms of MT quality. The performance of the approach on its own is quite low compared to the baseline SMT system, but the approach shows marginal improvements when combined with SMT.

In our third system, we showed a novel strategy of integrating TM into an EBMT system (EBMT_{TM}) in Chapter 5. This system has shown quite promising results for all the experiments conducted in this thesis. Marcu (2001a) showed that adding a TM into an SMT system improved translation quality. In his paper, he further anticipated that the use of similar techniques for EBMT systems might lead to improvements in translation quality for homogeneous data. This expectation has

been successfully corroborated in our experiments. The effect of this approach is far greater when the input data is homogeneous to the existing example-base (e.g. the English-to-Bangla experiment in Section 5.4) and when resources are limited (e.g. the English-to-Turkish experiment in Section 5.7). We have also shown that the approach works well for a moderately sized corpus (the English-to-French experiments in Section 5.5.2) for certain sentences. We showed that a feature-based combination of the EBMT_{TM} approach with SMT has a higher score than the individual baseline systems. In addition, we provided evidence that we can indeed mitigate some of the problems of SMT through the use of EBMT techniques.

In Chapter 6, we extend our work to improve the scalability of the EBMT_{TM} system. The basic EBMT_{TM} system presented in Chapter 5 is a runtime approach which has high time complexity (due to use of the time-consuming edit-distance measure) when using a larger example-base. We investigated two alternative approaches (a heuristic-based and an IR-based approach) to tackle this problem. We found that the integration of IR-based indexing and retrieval substantially improves runtime performance without affecting BLEU score.

Now we revisit the research questions we proposed in Chapter 1:

(RQ1) Can we exploit EBMT approaches to build better quality MT systems compared to purely SMT-based systems when working with limited resources?

(RQ2) Can we use a TM technology within an EBMT system for translating homogeneous data?

(RQ3) How effectively can we combine EBMT systems with state-of-the-art phrase-based SMT systems to handle the particular data sparsity in SMT?

(RQ4) If the EBMT/TM-based approach successfully works with limited homogeneous data, can we scale up the basic system to larger amounts of training data?

Initially, we used proportional analogy and a generalized translation-template-based approach to tackle **RQ1**. The performance of these two approaches as standalone systems is quite low when compared to the baseline SMT system. Hence, these two approaches are unable to find a comprehensive answer to the the research question RQ1. However, the combination of the proportional analogy approach with SMT partially answers research question RQ3 for two different tasks (NE transliteration and English-to-Chinese MT). Like the analogy-based approach, the translation-template-based approach answers research question RQ3 for two different MT tasks, showing that under certain conditions we can effectively combine an EBMT system with a phrase-based SMT system to handle the data sparsity problem of SMT.

We integrated a subsentential TM into an EBMT system in response to **RQ2** in Chapter 5. In addition to the user's TM, we used SMT to construct supplementary subsentential translation units in the TM. We used the TM in the alignment and recombination stages of the EBMT system. This approach on its own has shown promising results when the amount of resource is limited. Hence, this newly developed system based on RQ2 also successfully addresses the research question **RQ1**.

We found that the proportional analogy and generalized translation-template-based approaches had moderate success in answering **RQ3**. However, the EBMT_{TM} system has successfully answered RQ3. The EBMT_{TM} method was successfully combined with the state-of-the-art SMT system using two different features (FMS and EqUS). The integration of the two approaches gave an improvement in both automatic and human evaluation scores. We combined the EBMT_{TM} with an SMT system based on certain features to make the best use of the two individual systems. This integration has proven to be successful in our experiments, when exploiting both small and medium-sized data.

Finally, we tackled **RQ4** in Chapter 6 by applying an IR-based technique to the matching stage of the EBMT_{TM} system. Experimental results showed that the integration of IR-based matching improves the scalability of the EBMT_{TM} system without hampering the translation quality, thus providing a positive answer to RQ4.

7.1 Contribution

In sum, we have explored different EBMT techniques and have proposed a new EBMT approach using TM to mitigate some of the problems of SMT. In this research we have made the following contributions:

- We have explored a runtime EBMT approach (using proportional analogy) and have drawn some conclusions on the best scenario to use this approach. We have proposed a new heuristic and have compared this with other heuristics from the literature. Finally, we have shown how these approaches can be effectively incorporated into a state-of-the-art phrase-based SMT system to produce better quality translations.

- We have also explored a compiled approach to EBMT and have shown the effect of combining it with an SMT-based approach for translating homogeneous data in a resource-poor setting.

- We have presented a novel runtime EBMT system using TM that performs well with limited amounts of homogeneous data. We have also presented the use of different features to improve the output quality when combining our EBMT_{TM} system with an SMT system.

- We have shown the effective integration of IR technique (indexing and retrieval) within the workflow of a runtime EBMT system. Incorporating IR technology provides us with a much more scalable solution when using a large example-base.

7.2 Future Work

While this thesis has described a number of data-driven approaches to MT for translating homogeneous domain-specific data in a resource-poor setting, there remain a number of avenues for future work which we believe warrant further exploration.

As noted in Chapter 5, our EBMT_{TM} system uses a subsentential TM for both alignment and recombination. The entries in the subsentential TM may contain incorrect source–target translation equivalences as it is automatically built using Moses word/phrase alignments. Due to incorrect TU equivalents in the TM, the EBMT_{TM} system sometimes produces inappropriate alignment between the closest matched sentence pair $\langle s_c, t_c \rangle$. Finding the alignments between source and target sentences to identify possible edits (to assist CAT users) is still an area of active research (Esplà et al., 2011b). Instead of fully relying on the TM, we can use *alignment strength* to identify the target correspondence in t_c for each of the unmatched segments in s_c, using a geometrical alignment strategy (Esplà et al., 2011a).

In Section 5.3.3, in the recombination stage of the EBMT_{TM} system, we obtain the translation of the unmatched source segments (segments that need to be added or substituted in the skeleton translation t_c) using subsentential TM. We choose the most probable target equivalent for the unmatched source segment solely based on the phrase translation probability and lexical weighting. This is of course a risky strategy as it will select the same target equivalent for all instances of a given source segment. This method can be further improved by incorporating an n-gram language model. The use of a language model will enable the selection of context-informed target equivalents from the TM.

Though the fuzzy match score (FMS) has shown to be a good estimator for triggering the use of EBMT systems, the use of more sophisticated features may produce better quality translations. Following this direction, we have found that equal number of unmatched segments (EqUS) (in Section 5.5) used in conjunction with FMS is a good estimator for this purpose. Additionally, more features (e.g. the maximum length of mismatches, average length of the mismatch) can be explored to find a better triggering environment for an EBMT system. In our current experimental setup, we empirically decide the threshold of a feature which would necesitate the use of an EBMT system. This can potentially be extended by using a machine learning strategy to set the threshold for the features.

Finally, our present matching algorithm relies solely on the surface form of the words to find the closest matching sentence (in Section 5.3.1). This hypothesis may have some drawbacks for morphologically-rich languages (e.g. Bangla, Turkish) as they take on different inflected forms based on agreement with other words. Therefore, further linguistic investigation might help to achieve better accuracy for the EBMT_{TM} approach. Instead of using the surface form of the word, the EBMT system's processes can be applied at the *morpheme* level using a source-side morphological analyzer (to split words into morphemes). Furthermore, a target-language morphological generator could be used to produce the target-language surface forms.

Bibliography

Armstrong, S., Caffrey, C., Flanagan, M., Kenny, D., O'Hagan, M., and Way, A. (2006). Improving the Quality of Automated DVD Subtitles via Example-Based Machine Translation. In *Translating and the Computer **28***, [no page number], London: Aslib, UK.

Banerjee, P., Dandapat, S., Forcada, M. L., Groves, D., Penkale, S., Tinsley, J., and Way, A. (2011). OpenMaTrEx, a free, open-source hybrid data-driven machine translation system. Technical report, Centre for Next Generation Localization, School of Computing, Dublin City University, Ireland.

Banerjee, S. and Lavie, A. (2005). METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgements. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, page 65–72, Ann Arbor, MI.

Barreiro, A., Scott, B., Kasper, W., and Kiefer, B. (2011). OpenLogos machine translation: philosophy, model, resources and customization. *Machine Translation*, **25**(2):107–126.

Berger, A. L., Pietra, S. A. D., and Pietra, V. J. D. (1996). A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, **22**(1):39–71.

Biçici, E. and Dymetman, M. (2008). Dynamic Translation Memory: Using Statisti-

cal Machine Translation to Improve Translation memory Fuzzy Matches. *Lecture Notes in Computer Science*, **4919**:454–465.

Bourdaillet, J., Huet, S., Gotti, F., Lapalme, G., and Langlais, P. (2009). Enhancing the Bilingual Concordancer TransSearch with Word-level Alignment. In *Proceedings, volume 5549 of Lecture Notes in Artificial Intelligence: 22nd Canadian Conference on Artificial Intelligence (Canadian AI 2009), Springer-Verlag*, page 23–38.

Bowker, L. (2002). Computer-aided Translation Technology: a Practical Introduction. In *Translation Memory Systems*, page 92–127, University of Ottawa Press, Ottawa.

Brown, P., Cocke, J., Della Pietra, S., Della Pietra, V., Jelinek, F., Mercer, R., and Roossin, P. (1988). A Statistical Approach to Language Translation. In *Proceedings of the 12th International Conference on Computational Linguistics, (COLING 1988)*, page 71–76, Budapest, Hungary.

Brown, P., Pietra, J., Pietra, S. D., Jelinek, F., Mercer, R., and Roossin, P. (1990). A Statistical Approach to Machine Translation. *Computational Linguistics*, **16**:79–85.

Brown, P. F., Pietra, V. J. D., Pietra, S. A. D., and Mercer, R. L. (1993). The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, **19**(2):263–311.

Brown, R. (1996). Example-Based Machine Translation in the Pangloss System. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING 1996)*, page 169–173, Copenhagen, Denmark.

Brown, R. (1999). Adding Linguistic Knowledge to a Lexical Example-based Translation System. In *Proceedings of the 18th Conference on Theoretical and Methodological Issues in Machine Translation (TMI 1999)*, page 22–32, Chester, England.

Brown, R. (2000). Automated Generalization of Translation Examples. In *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*, page 125–131, Saarbrücken, Germany.

Brown, R. (2001). Transfer-Rule Induction for Example-Based Translation. In *Proceedings of the MT Summit VIII Workshop on Example-Based Machine Translation*, page 1–11, Santiago de Compostela, Spain.

Brown, R. (2004). A Modified Barrows-Wheeler Transform for Highly Scalable Example-Based Translation. In *Proceedings of the 6th International Conference of the Association for Machine Translation in the Americas (AMTA 2004)*, page 27–36, Washington, DC.

Brown, R. and Frederking, R. (1995). Applying Statistical English Language Modeling to Symbolic Machine Translation. In *Proceedings of the 6th International Conference on Theoretical and Methodological Issues in Machine (TMI 1995)*, page 221–239, Leuven, Belgium.

Brown, R. D. (2011). The CMU-EBMT machine translation system. *Machine Translation*, **25**(2):179–195.

Callison-Burch, C., Bannard, C., and Schroeder, J. (2005). Scaling Phrase-Based Statistical Machine Translation to Larger Corpora and Longer Phrases. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, page 255–262, University of Michigan, Ann Arbor.

Carl, M. and Way, A., editors (2003). *Recent Advances in Example-Based Machine Translation*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Carroll, J. J. (1999). Repetitions Processing Using a Metric Space and the Angle of Similarity. Report No. 90/3. Centre for Computational Linguistics, UMIST, Manchester, England.

Chiang, D., Lopez, A., Madnani, N., Monz, C., Resnik, P., and Subotin, M. (2005). The Hiero Machine Translation System: Extensions, Evaluation and Analysis. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP 2005)*, page 779–786, Vancouver, Canada.

Cicekli, I. (2005). Inducing Learning Translation Templates with Type Constraints. *Machine Translation*, **19**(3–4):283–299.

Cicekli, I. and Güvenir, H. A. (2001). Learning Translation Templates from Bilingual Translation Examples. *Applied Intelligence*, **15**(1):57–76.

Cranias, L., Papageorgiou, H., and Piperidis, S. (1994). A Matching Technique in Example-Based Machine Translation. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING 2000)*, page 100–104, Kyoto, Japan.

Dandapat, S., Biswas, P., Choudhury, M., and Bali, K. (2009). Complex Linguistic Annotation – No Easy Way Out! In *Proceedings of the 3rd Linguistic Annotation Workshop (LAW-III), a Workshop at the Joint Conference of 47th ACL and 4th IJCNLP)*, page 10–18, Singapore.

Dandapat, S., Forcada, M. L., Groves, D., Penkale, S., Tinsley, J., and Way, A. (2010a). OpenMaTrEx: A Free/Open-Source Marker-Driven Example-Based Machine Translation System. In *Proceedings of the 7th International Conference on Natural Language Processing (IceTAL 2010)*, page 121–126, Reykjavík, Iceland.

Dandapat, S., Morrissey, S., and Somers, H. (2010b). Mitigating Problems in Analogy-Based EBMT with SMT and Vice Versa: a Case Study with Named Entity Transliteration. In *Proceedings of the 24th Pacific Asia Conference on Language Information and Computing (PACLIC 2010)*, pages 146--153, Sendai, Japan.

Dandapat, S., Morrissey, S., and Somers, H. (2010c). Statistically Motivated Example-Based Machine Translation using Translation Memory. In *Proceedings of the 8th International Conference on Natural Language Processing (ICON 2010)*, pages 168--177, Kharagpur, India.

Dandapat, S., Morrissey, S., Way, A., and Forcada, M. L. (2011). Using Example-Based MT to Support Statistical MT when Translating Homogeneous Data in Resource-Poor Settings. In *Proceedings of the 15th Annual Meeting of the European Association for Machine Translation (EAMT 2011)*, page 201–208, Leuven, Belgium.

Dandapat, S., Morrissey, S., Way, A., and van Genabith, J. (2012). Combining EBMT, SMT, TM and IR Technologies for Quality and Scale. In *Proceedings of the EACL 2012 Joint Workshop on Exploiting Synergies between Information Retrieval and Machine Translation (ESIRMT) and Hybrid Approaches to Machine Translation (HyTra)*, pages 48--58, Avignon, France.

Denoual, E. (2005). The Influence of Example-data Homogeneity on EBMT Quality. In *Proceedings of the 2nd Workshop on Example-based Machine Translation, a Workshop at the MT Summit X*, page 35–42, Phuket, Thailand.

Denoual, E. (2007). Analogical Translation of Unknown Words in a Statistical Translation Framework. In *Proceedings of the 11th Machine Translation Summit, (MT SUMMIT XI)*, page 135–141, Copenhagen, Denmark.

Doddington, G. (2002). Automatic Evaluation of Machine Translation Quality using N-gram Co-occurrence Statistics. In *Proceedings of the 2nd International Conference on Human Language Technology Research, (HLT 2002)*, page 128–132, San Diego, California.

Elliston, J. S. G. (1979). Computer Aided Translation: A Business Viewpoint, in Barbara M. Snell (ed.), *Translating and the Computer: Proceeding of a Seminar, London, 14th November, 1978.* page 149–158. North-Holland, Amsterdam.

Esplà, M., Sánchez-Martínez, F., and Forcada, M. L. (2011a). Target-Language Edit Hints: a Basic Description of the Method. Technical Report. Dep. de Llenguatges i Sistemes Informàtics, Universitat d'Alacant, Spain.

Esplà, M., Sánchez-Martínez, F., and Forcada, M. L. (2011b). Using Word Alignments to Assist Computer-aided Translation Users By Marking Which Target-side Words to Change or Keep Unedited. In *Proceedings of the 15th Annual Meeting of the European Association for Machine Translation (EAMT 2011)*, pages 81--88, Leuven, Belgium.

Fleiss, J. L. (1971). Measuring Nominal Scale Agreement among Many Raters. *Psychological Bulletin*, **76**(5):378–382.

Forcada, M. L., Ginestí-Rosell, M., Nordfalk, J., O'Regan, J., Ortiz-Rojas, S., Pérez-Ortiz, J. A., Sánchez-Martínez, F., Ramírez-Sánchez, G., and Tyers, F. M. (2011). Apertium: a free/open-source platform for rule-based machine translation. *Machine Translation*, **25**(2):127–144.

Gentner, D. (1983). Structural Mapping: A Theoretical Model for Analogy. *Cognitive Science*, **7**(2):155–170.

Gough, N. and Way, A. (2004). Robust Large-Scale EBMT with Marker-Based Segmentation. In *Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation, (TMI 2004)*, page 95–104, Baltimore, MD.

Green, T. (1979). The Necessity of Syntax Markers: Two Experiments with Artificial Languages. *Journal of Verbal Learning and Behavior*, **18**:481–496.

Groves, D. and Way, A. (2006). Hybridity in MT: Experiments on the Europarl Corpus. In *Proceedings of the 11th Annual Conference of the European Association for Machine Translation, (EAMT 2006)*, page 115–124, Oslo, Norway.

Güvenir, H. A. and Cicekli, I. (1998). Learning Translation Templates from Examples. *Information Systems*, **23**(6):353–363.

Haizhou, L., Kumaran, A., and Valdimir, P. (2009). Report of NEWS 2009 Machine Translation Shared Task. In *Proceedings of Named Entities Workshop (NEWS) 2009, a Workshop at the Joint conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, page not found, Suntec, Singapore.

He, Y. (2011). *The Integration of Machine Translation and Translation Memory*. PhD thesis, School of Computing, Dublin City University.

He, Y., Ma, Y., van Genabith, J., and Way, A. (2010). Bridging SMT and TM with Translation Recommendation. In *Proceedings of the 48th Annual Meeting of the Association of Computational Linguistics (ACL 2010)*, page 622–630, Uppsala, Sweden.

Hearne, M. (2005). *Data-Oriented Models of Parsing and Translation*. PhD thesis, Dublin City University, Ireland.

Hermjakob, U., Knight, K., and Daume III, H. (2008). Name Translation in Statistical Machine Translation Learning When to Transliterate. In *Proceedings of the 46th Annual Meeting of the Association of Computational Linguistics (ACL 2008)*, page 389–397, Columbus, Ohio.

Hiemstra, D. (2001). *Using Language Models for information Retrieval*. PhD thesis, University of Twente, The Netherlands.

Hildebrand, A. S., Eck, M., Vogel, S., and Waibel, A. (2005). Adaptation of the Translation Model for Statistical Machine Translation Based on Information Retrieval. In *Proceedings of the 10th Conference of the European Association for Machine Translation (EAMT 2005)*, page 133–142, Budapest, Hungary.

Hutchins, J. (2005). Example-Based Machine Translation: a Review and Commentary. *Machine Translation*, **19**(3–4):197–211.

Islam, M., Tiedemann, J., and Eisele, A. (2010). English–Bangla Phrase-based Machine Translation. In *Proceedings of the 14th Annual Conference of the European Association of Machine Translation (EAMT 2010)*, [no page number], Saint-Raphaël, France.

Jelinek, F. and Mercer, R. L. (1980). Interpolated Estimation of Markov Source Parameters from Sparse Data. In *Proceedings of the Workshop of Pattern Recognition in Practice*, page 381–397, Amsterdam, The Netherlands.

Kaji, H., Kida, Y., and Morimoto, Y. (1992). Learning Translation Templates from Bilingual Text. In *Proceedings of the 15th [sic] International Conference on Computational Linguistics, (COLING 92)*, page 672–678, Nantes, France.

Kay, M. (1980). The Proper Place of Men and Machines in Language Translation. Technical report, CSL-80-11, Xerox Palo Alto Research Center, Palo Alto, Calif. Reprinted in *Machine Translation* (1997) **12:**3–23.

Khalilov, M., Fonollosa, J., Skadina, I., Bralitis, E., and Pretkalnina, L. (2010). English–Latvian SMT: the Challenge of Translating into a Free Word Order Language. In *Proceedings of the 2nd International Workshop on Spoken Language Technologies for Under-resourced Languages (SLTU 2010)*, page 87–94, Penang, Malaysia.

Knight, K. (1999). Decoding Complexity in Word-Replacement Translation Model. *Computational Linguistics*, **25**(4):607–615.

Knight, K. and Graehl, J. (1998). Machine Transliteration. *Computational Linguistics*, **24**(4):559–612.

Koehn, P. (2004). Statistical Significance Tests for Machine Translation Evalua-

tion. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, (EMNLP 2004)*, page 388–395, Barcelona, Spain.

Koehn, P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. In *MT Summit X: The 10th Machine Translation Summit*, page 79–86, Phuket, Thailand.

Koehn, P. (2010). *Statistical Machine Translation*. Cambridge University Press, Cambridge, UK.

Koehn, P., Axelord, A., Mayne, R., Callison-Burch, C., Osborne, M., and Talbot, D. (2005). Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT 2005)*, page no page number, Pittsburgh, PA.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting for the Association for Computational Linguistics (ACL 2007)*, page 177–180, Prague, Czech Republic.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical Phrase-Based Translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (NAACL 2003)*, page 48–54, Edmonton, Canada.

Koehn, P. and Senellart, J. (2010a). Convergence of Translation Memory and Statistical Machine Translation. In *Proceedings of the 2nd Joint EM+/CNGL, Workshop "Bringing MT to the User: Research on Integrating MT in the Translation Industry"*, page 21–31, Denver, CO.

Koehn, P. and Senellart, J. (2010b). Fast Approximate String Matching with Suffix Arrays and A* Parsing. In *Proceedings of the 9th Annual Conference of the*

Association for Machine Translation in Americas (AMTA 2010), page 45–57, Denver, CO.

Kumaran, A. and Kellner, T. (2007). A Generic Framework for Machine Transliteration. In *Proceedings of the 30th Annual International Conference on Research and Development in Information Retrieval (SIGIR 2007)*, page 721–722, Amsterdam, The Netherlands.

Landis, J. R. and Koch, G. C. (1977). The Measurement of Observer Agreement for Categorical Data. *Biometrics*, **33**:159–174.

Langlais, P., Foster, G., and Lapalme, G. (2000). TransType: A Computer-Aided translation Typing System. In *Proceedings of the Embedded Machine Translation Systems, a Workshop at the Joint Conference of the 6th Applied Natural Language Processing and 1st Meeting of the North American Chapter of the Association for Computational Linguistics (ANLP/NAACL 2000)*, page 46–51, Seattle, WA.

Langlais, P. and Patry, A. (2007). Translating Unknown Words using Analogical Learning. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, (EMNLP-CoNLL 2007)*, page 877–886, Prague, Czech Republic.

Langlais, P. and Yvon, F. (2008). Scaling up Analogical Learning. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008)*, page 51–54, Manchester, UK.

Langlais, P., Yvon, F., and Zweigenbaum, P. (2009). Improvements in Analogical Learning: Application to Translating Multi-terms of the Medical Domain. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2009)*, page 487–495, Athens, Greece.

Lardilleux, A. (2011). *The Contribution of Low Frequencies to Multilingual Subsentential Alignment: a Differential Associative Approach*. PhD thesis, LIMSI-CNRS, France.

Lepage, Y. (1998). Solving Analogies on Words: an Algorithm. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, (COLING-ACL 1998)*, page 728–734, Quebec, Canada.

Lepage, Y. (2004). Lower and higher estimates of the number of "true analogies" between sentences contained in a large multilingual corpus. In *Proceedings of the 20th International Conference on Computational Linguistics(COLING 2004)*, page 736–742, Geneva, Switzerland.

Lepage, Y. and Denoual, E. (2005a). ALEPH: an EBMT System Based on the Preservation of Proportional Analogies between Sentences across Languages. In *Proceedings of the International Workshop on Spoken Language Translation: Evaluation Campaign on Spoken Language Translation, (IWSLT 2005)*, page 8, Pittsburgh, Pennsylvania.

Lepage, Y. and Denoual, E. (2005b). Purest Ever Example-based Machine Translation: Detailed Presentation and Assessment. *Machine Translation*, **19**(3–4):251–282.

Lepage, Y. and Denoual, E. (2005c). The 'purest' EBMT System Ever Built: No Variables, No Templates, No Training, Examples, Just examples, Only Examples. In *Proceedings of the 2nd Workshop on Example-based Machine Translation, a Workshop at the MT Summit X*, page 81–90, Phuket, Thailand.

Lepage, Y. and Lardilleux, A. (2007). The GREYC Machine Translation System for the IWSLT 2007 Evaluation Campaign. In *Proceedings of the 4th International Workshop on Spoken Language Translation, (IWSLT 2007)*, page 7, Trento, Italy.

Levenshtein, V. I. (1965). Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Doklady Akademii Nauk SSSR*, **163**(4):845–848.

Lewis, W. (2010). Haitian Creole: How to Build and Ship an MT Engine from Scratch in 4 Days, 17 Hours, and 30 Minutes. In *Proceedings of the 4th Annual*

conference of the European Association for Machine Translation (EAMT 2010), pages 501--511, Saint-Rapha
"el, France.

Li, H., Kumaran, A., Zhang, M., and Pervouchine, V. (2009). Whitepaper of NEWS 2009 Machine Transliteration Shared Task. In *Proceedings of the 2009 Named Entity Workshop: Shared Task on Transliteration, (NEWS 2009)*, Suntec, Singapore. page number not found.

Lopez, A. (2008). *Machine Translation by Pattern Matching*. PhD thesis, University of Meryland.

Ma, X. and Cieri, C. (2006). Corpus Support for Machine Translation at LDC. In *Proceedings of 5th International Conference on Language Resources and Evaluation (LREC 2006)*, page 859–864, Genoa, Italy.

Macklovitch, E. (2006). TransType2: the last word. In *LREC-2006: Fifth International Conference on Language Resources and Evaluation, Proceedings*, page 167–172, Genoa, Italy.

Manning, C. D., Raghavan, P., and Schütze, H. (2008a). *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, U.K.

Manning, C. D., Raghavan, P., and Schütze, H. (2008b). *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK.

Marcu, D. (2001a). Towards a Unified Approach to Memory- and Statistical-Based Machine Translation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics and 10th Conference of the European Chapter*, page 386–393, Toulouse, France.

Marcu, D. (2001b). Towards a Unified Approach to Memory- and Statistical-based Machine Translation. In *Proceedings of the Association for Computational Lin-*

guistics, 39th Annual Meeting and 10th Conference of the European Chapter, page 386–393, Toulouse, France.

Maruyama, H. and Watanabe, H. (1992). Tree Cover Search Algorithm for Example-Based Translation. In *Proceedings of the 4th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI 1992)*, page 173–184, Kyoto, Japan.

Morrissey, S., Somers, H., Smith, R., Gilchrist, S., and Dandapat, S. (2010). Building a Sign Language corpus for use in Machine Translation. In *Proceedings of the 4th Workshop on Representation and Processing of Sign Languages: Corpora for Sign Language Technologies*, pages 172--177, Valetta, Malta.

Muraki, K. (1987). PIVOT: Two-Phase Machine Translation System. In *Proceedings of the 1st Machine Translation Summit (MT Summit I)*, page 81–83, Hakone, Japan.

Nagao, M. (1984). A Framework of a Mechanical Translation between Japanese and English by Analogy Principle. In Elithorn, A. and Banerji, R., editors, *Artificial and Human Intelligence*, page 173–180. North-Holland, Amsterdam.

Nießen, S. and Ney, H. (2004). Statistical Machine Translation with Scarce Resources Using Morpho-syntactic Information. *Computational Linguistics*, **30**(2):181–204.

Nirenburg, S., Domashnev, C., and Grannes, D. J. (1993). Two Approaches to Matching in Example-Based Machine Translation. In *Proceedings of the 5th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI 1993)*, page 45–57, Kyoto, Japan.

Ó'Baoill, D. and Matthews, P. A. (2000). *The Irish Deaf Community (Volume 2): The Structure of Irish Sign Language*. The Linguistics Institute of Ireland, Dublin, Ireland.

Och, F. and Ney, H. (2002). Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, page 295–302, Philadelphia, PA.

Och, F. and Ney, H. (2003). A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, **29**(1):19–51.

Öz, Z. and Cicekli, I. (1998). Ordering Translation Templates by Assigning Confidence Factors. In *Proceedings of the 3rd Conference of the Association for Machine Translation in the Americas (AMTA 1998)*, page 51–61, Langhorne, PA.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, (ACL 2002)*, page 311–318, Philadelphia, PA.

Phillips, A. B. (2011). Cunei: open-source machine translation with relevance-based models of each translation instance. *Machine Translation*, **25**(2):161–177.

Phillips, A. B. (2012). *Modeling relevance in Statistical Machine Translation*. PhD thesis, Language Technologies Institute, Carnegie Mellon University.

Phillips, A. B., Cavalli-Sforza, V., and Brown, R. D. (2007). Improving Example Based Machine Translation through Morphological Generalization and Adaptation. In *Proceedings of the 9th Machine Translation Summit, (MT SUMMIT IX)*, page 369–375, Copenhagen, Denmark.

Popović, M. and Ney, H. (2006). Statistical Machine Translation with a Small Amount of Bilingual Training Data. In *Proceedings of LREC 20065th SALTMIL Workshop on Minority Languages*, page 25–29, Genoa, Italy.

Sikes, R. (2007). Fuzzy Matching in Theory and Practice. *Multilingual*, **18**(6):39–43.

Simard, M. (2003). Translation Spotting for Translation Memories. In *Proceedings of the conference combining Human Language Technology conference series and the North American Chapter of the Association for Computational Linguistics conference series, (HLT-NAACL 2003)*, page 65–72, Edmonton, Canada.

Simard, M. and Isabelle, P. (2009). Phrase-Based Machine Translation in a Computer-Assisted Translation Environment. In *Proceedings of the 12th Machine Translation Summit (MT Summit XII)*, page 120–127, Ottawa, Canada.

Snover, M., Dorr, B., Schwartz, R., and anmd J. Makhoul, L. M. (2006). A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of the 7th Annual Conference of the Association for Machine Translation in Americas (AMTA 2006)*, page 223–231, Cambridge, MA.

Somers, H. (2003). An Overview of EBMT. In Carl, M. and Way, A., editors, *Recent Advances in Example-based Machine Translation*, page 3–57. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Somers, H., Dandapat, S., and Naskar, S. K. (2009). A review of EBMT using proportional analogy. In *Proceedings of the 3rd Workshop on Example-Based Machine Translation (EBMT 2009)*, pages 53--60, Dublin, Ireland.

Somers, H., McLean, I., and Jones, D. (1994). Experiment in Multilingual Example-Based Generation. In *Proceedings of the 3rd Conference on the Cognitive Science of Natural Language Processing (CSNLP 1994)*, Dublin, Ireland. pages not numbered.

Soper, H. E., Young, A. W., Cave, B. M., Lee, A., and K., P. (1917). On the distribution of the correlation coefficient in small samples. Appendix II to the papers of "Student" and R. A. Fisher. A co-operative study. *Biometrika*, **11**:328–413.

Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufiș, D., and

Varga, D. (2006). A Multilingual Parallel Corpus with 20+ Languages. In *Proceedings of the 5th International Conference on Language Resource and Evaluation (LREC 2006)*, page 2142–2147, Genoa, Italy.

Stolcke, A. (2002). SRILM — An Extensible Language Modeling Toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, page 901–904, Denver, CO.

Sumita, E., Lida, H., and Kohyama, H. (1990). Translating with Examples: A new Approach to Machine Translation. In *Proceedings of the 3rd International Conference on Theoretical and Methodological Issues in Machine Translation (TMI 1990)*, page 203–212, Austin, TX.

Tiedemann, J. and Nygaard, L. (2004). The OPUS Corpus - Parallel and Free. In *Proceedings of 4th International Conference on Language Resources and Evaluation, (LREC 2004)*, page 1183–1186, Lisbon, Portugal.

Tiedemann, J. and Nygaard, L. (2009). News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces, in N. Nicolov, K. Bontcheva, G. Angelova, R. Mitkov, (eds.). *Recent Advances in Natural Language Processing*, **V**:237–248.

Ukkonen, E. (1983). On Approximate String Matching. In *Proceedings of the International Conference on Foundations of Computing Theory (FCT 1983)*, page 487–496, Borgholm, Sweden.

Varga, D., Németh, L., Halácsy, P., Kornai, A., Trón, V., and Nagy, V. (2005). Parallel Corpora for Medium Density Languages. In *Proceedings of the 5th International Conference on Recent Advances Natural Language Processing (RANLP 2005)*, page 590–596, Borovets, Bulgaria.

Vauquois, B. and Christian, B. (1985). Automated Translation at Grenoble University. *Computational Linguistics*, **11**:28–36.

Wagner, R. A. and Fischer, M. J. (1974). The String-to-String Correction Problem. *Journal of the Association for Computing Machinery*, **21**:168–173.

Way, A. and Hearne, M. (2011). Statistical Machine Translation: A guide for Linguists and Translators. *Language and Linguistic Compass*, **5**:205–226.

Weaver, W. (1949). Recent Contributions to the Mathematical Theory of Communication. In Shannon, C. E. and Weaver, W., editors, *The Mathematical Theory of Communication*, page 94–117. The University of Illinois Press, Urbana, IL.

Yamamoto, M. and Church, K. W. (2001). Using Suffix Array to Compute Term Frequency and Document Frequency for All Substrings in a Corpus. *Computational Linguistics*, **27**(1):1–30.

Zhang, Y. and Vogel, S. (2005). An Efficient Phrase-to-Phrase Alignment Model for Arbitrary Long Phrases and Large Corpora. In *Proceedings of the 10th Conference of the European Association for Machine Translation (EAMT 2005)*, page 294–301, Budapest, Hungary.

Zhechev, V. and van Genabith, J. (2010). Seeding Statistical Machine Translation with Translation Memory Output through Tree-based Structural Alignment. In *Proceedings of the COLING'10, Workshop on Syntax and Structure in Statistical Translation*, page 43–51, Beijing, China.