Domain Adaptation for Statistical Machine Translation of Corporate and User-Generated Content

Pratyush Banerjee

B. Tech, 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, Prof. Josef van Genabith
and Dr. Johann Roturier

January 2013

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, 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

List of Figures

List of Tables

Abstract

The growing popularity of Statistical Machine Translation (SMT) techniques in recent years has led to the development of multiple domain-specific resources and adaptation scenarios. In this thesis we address two important and industrially relevant adaptation scenarios, each suited to different kinds of content.

Initially focussing on professionally edited 'enterprise-quality' corporate content, we address a specific scenario of data translation from a mixture of different domains where, for each of them domain-specific data is available. We utilise an automatic classifier to combine multiple domain-specific models and empirically show that such a configuration results in better translation quality compared to both traditional and state-of-the-art techniques for handling mixed domain translation.

In the second phase of our research we shift our focus to the translation of possibly 'noisy' user-generated content in web-forums created around products and services of a multinational company. Using professionally edited translation memory (TM) data for training, we use different normalisation and data selection techniques to adapt SMT models to noisy forum content. In this scenario, we also study the effect of mixture adaptation using a combination of in-domain and out-of-domain data at different component levels of an SMT system. Finally we focus on the task of optimal supplementary training data selection from out-of-domain corpora using a novel incremental model merging mechanism to adapt TM-based models to improve forum-content translation quality.

Acknowledgments

I would first like to express my profound sense of gratitude to my supervisors Prof. Andy Way, Prof. Josef van Genabith and Dr. Johann Roturier, 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 grateful to Prof. Gareth Jones and Dr. John Tinsley, who provided insightful suggestions on my transfer report, much of which is incorporated into this thesis. I would also specially thank Dr. Fred Hollowood, for allowing me to work with Symantec data and tools, and for welcoming me to his team and workspace in Symantec.

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 Dr. Sudip Naskar, who has always been a keen researcher and a warm-handed friend. I would also like to extend my gratitude towards Dr. Jinhua Du, Dr. Baoli Li and Prof. Carl Vogel for helping me out with some parts of my initial work. I would also like to thank Dr. Sandipan Dandapat who has been a constant source of support, both professionally and personally, since the beginning of my endeavour.

Thanks to Dr. Yanjun Ma, Dr. Yifan He, Dr. Ankit Srivastava, Dr. Tsuyoshi Okita, Dr. Sergio Penkale, Dr. Rejwanul Haque, Dr. Hala AlMaghout, Rasoul Samad Zadeh Kaljahi and Dr. Raphael Rubino and many other members of my research group for their support and interest in my work. Special thanks to Dr. Joachim Wagner for maintaining our computing cluster in excellent shape and consistently providing unix tips. Thanks to Eithne McCann, Ríona Finn, and Fiona Mcguire for their kind help since the first day I arrived in Ireland. Finally thanks to my wider research group, Debasis, Maria, Javed and Yalemisew for providing much needed distraction during the tea breaks.

A special set of thanks to Dr Junhui Li, Dr. Xiaofeng Wu, Dr. Yanli Sun, Dr. Johann Roturier and Linda Mitchell and all the staff of shared engineering services (SES) in Symantec for helping me with the examples, manual evaluations and supervising manual translations process for my datasets.

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. Finally my deepest gratitude to my wife Shreyoshi Banerjee, without whose continuous support and immense patience, it would have been impossible for me to complete this work. I thank all my well-wishers who directly and indirectly contributed to 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) as well as Symantec Limited, Dublin for supporting this research.

Chapter 1

Introduction

Domain adaptation has been a core problem in the field of machine learning (ML) and natural language processing (NLP) for many years. While many early approaches used rule- or constraint-based methods, the recent advent of statistical methods has drawn an increasing number of researchers to study this problem. One of the basic assumptions in statistical learning theory is the uniformity of training and test distributions. In real-life applications, however, this is seldom the case. Accordingly, considerable effort has focussed on tackling the problem of domain adaptation which involves the development of techniques which enable easy porting of models trained in one domain to applications in other domains with minimum error in terms of the relevant evaluation metrics. Statistical machine translation (SMT), in which statistical models are developed to enable translation between two natural languages, assimilates research from both ML and NLP. This makes domain adaptation a pertinent problem in SMT as well. In the context of SMT, the problem of domain adaptation involves designing techniques and algorithms that allow models trained on data from a particular domain to successfully translate sentences from other domains.

State-of-the-art SMT systems are trained on large bilingual parallel corpora. Since the internal models in an SMT system comprise statistics estimated on the training data, the performance of such systems is dependent on the quality and quantity of the available training data (Axelrod et al., 2011). Therefore, larger amounts of training data generally leads to more accurate estimates of the statistics and hence better models. The popularity of SMT techniques have led to the development of large amounts of freely available parallel corpora (e.g. Europarl (Koehn, 2005), OPUS (Tiedemann, 2009), etc.) on the web, which addresses the issue of training data quantity. However, many specific translation tasks, especially those encountered in commercial or industrial applications, have their own linguistic characteristics or vocabulary requirements resulting in the 'target' domain corpus characteristics being substantially different from those of the available training data. For example, the linguistic style required to translate parliamentary proceedings is quite different from that used to translate medical transcriptions or software manuals. This variation in corpus characteristics means that the performance of SMT systems is not only dependent on the quantity but also the quality, or the domain appropriateness of the training data (Ozdowska and Way, 2009). For every translation task, one would ideally use training data having the same (or very similar) corpus characteristics ('in-domain') as that of the targeted application domain. Therefore, the performance of an SMT system is dependent on the amount of available 'in-domain' data. Depending on the translation task under consideration, such data could either only be sparsely available or, in fact, be completely unavailable. In such situations it becomes imperative to find ways to use available training data from other ('out-of-domain') sources unrelated to the task at hand. This necessity of using out-of-domain data to translate in-domain data forms the primary motivation of domain adaptation techniques in SMT.

Domain adaptation applies to a wide range of scenarios in real-life applications of SMT, in particular in the localisation and language services industries. Language Service Providers (LSP) tend to create and customise translation systems for specific clients and requirements (Vashee and Gibbs, 2010). Reusing existing translation systems and tuning them to new requirements or combining multiple domain-specific translation systems to widen system coverage essentially require domain adaptation techniques. Style adaptation in SMT is also synonymous to domain adaptation as the variation of linguistic styles are a major source of divergence between different domains (Pecina et al., 2012).[1] The localisation industry – which often needs to use existing resources to create translation systems for stylistically different data in related domains (Schwarm et al., 2004) (e.g. knowledge-base articles used to train models translating online Help files in the IT industry, or Technical Specification documents used to translate user manuals in the Automotive industry) – also extensively relies on adaptation techniques. However, domain adaptation techniques by nature are very specific and closely related to the particular problems they are designed to solve. A well designed domain adaptation solution for one particular scenario might not be at all well suited for a different purpose.

The research reported in this thesis is carried out as part of a larger research group, Center for Next Generation Localisation (CNGL)[2] – a large academia-industry consortia project focussing on localisation challenges. Considering the impact of domain adaptation research on real-life applications of MT, and having access to a number of leading localisation industry partners, we wanted to align our research objectives to industrially relevant scenarios. The localisation industry, which extensively uses machine translation for increased productivity, qualifies as the obvious source and test bed for such domain adaptation scenarios. Symantec[3] – a global leader in security, storage and systems management solutions is well known (in the localisation circles) for developing and maintaining a highly efficient and flexible workflow for multi-stream localisation. Symantec's centralised localisation group drives product localisation for over forty different languages. In order to effectively support its high scale translation requirements, Symantec employs an internal R&D group to drive innovation in automating localisation tasks, testing tools and technologies, and measuring each aspect of the localisation process for efficiency.

[1] The terms 'style-adaptation' and 'domain-adaptation' has been used interchangeably in this thesis.

[2] http://www.cngl.ie/index.html

[3] http://www.symantec.com/index.jsp

Symantec's distinction as a world leader in localisation management combined with their position as one of the important industry partners in CNGL enabled us to use certain scenarios from their localisation workflows as research objectives for the work reported in this thesis.

In this thesis, we present our approach to domain adaptation in SMT using two specific scenarios. In the first part of the thesis we investigate the issue of mixed-domain data translation using an industry-driven scenario based on professionally edited Symantec corporate data. Due to the presence of different domains pertaining to individual product lines or services, Symantec's localisation workflow often needs to handle mixed-domain data. Moreover, new product or service acquisitions often add data from new domains to the existing mix. Therefore, handling the translation of such a dynamic mixture of data from different domains is an essential scenario within Symantec's localisation workflow. In the first part of the thesis we concentrate on professionally edited content.

In the later stages of the thesis, we shift our focus to another relevant scenario of translating possibly 'noisy' user-generated content from Symantec's user forums, where in-domain parallel training data is unavailable. Having a multilingual customer base, Symantec runs and supports online web-forums discussing their different products and services, in multiple languages. These forums are not only platforms for easy and direct interaction between Symantec and its customers, but also act as an effective alternative to more traditional forms of customer service. Hence, the availability of the information present in these forums across different languages is advantageous to both Symantec and its customers. This translation requirement drives the other set of research objectives we address in this thesis. By using a number of existing techniques and developing novel methods to address these issues, we further aim to plug some gaps in the current state-of-the-art in domain adaptation research for SMT.

1.1 The Notion of "Domain"

Domain Adaptation being the central theme of this dissertation, we start by introducing the notion of "domain" as discussed in the computational linguistics, machine learning or natural language processing literature. In most of the previous work on domain adaptation, the term domain has been used somewhat broadly– sometimes to refer to topics, or to capture distinction in terms of mode (spoken text versus written text) or even to refer to the variation in registers (formal written prose versus SMS communications) (Finkel and Manning, 2009). The categories specified in the Brown corpus, like 'general fiction', 'romance and love story', 'press: reportage' etc. are considered domains in Sekine (1997) and Ratnaparkhi (1999). By comparison, Gildea (2001) does not explicitly mention the term 'domain' but uses 'text types' and 'genres of text' instead. Genre in general is defined as a category assigned to a body of text on the basis of external criteria such as purpose, intended audience and activity type (Biber, 1988). The task of adaptation in the context of differences across genres have also been discussed in the literature (Lease et al., 2006). While Biber (1988) attributes the differences in domain from a sociolinguistic stand point, Blitzer et al. (2006) attributes domain difference mostly to differences in vocabulary. Overall, the notion of domain in the context of adaptation is not strictly defined and the terms 'domain', 'genre', 'register', 'text type' and 'style' are often used interchangeably in different communities (Lee, 2001).

Intuitively, texts can differ along several dimensions or parameters. These dimensions can range from sentence length (longer sentences in written registers compared to shorter ones in spoken registers), variation in vocabulary (presence of domain-specific terms in technical documentation or medical transcriptions) to characteristics of posts in social media (presence of emoticons, spelling errors, URLs etc). Hence the variation of texts between domains can be attributed to the combination of variations between multiple parameters (Plank, 2011). In the context of the experiments reported in this thesis, we use datasets from Symantec which mostly comprise data from the technical domain. For the first scenario of mixed-domain data translation, we work with two sub-domains which are defined by natural distinctions occurring between documentation associated with two different product lines within Symantec. A detailed description of the sub-domains and associated datasets is presented in Chapter 2 (Section 2.5.2). The major differences in the datasets between the two sub-domains are in terms of vocabulary. In this scenario, the term 'in-domain' refers to any text belonging to the same domain as the training data, while 'out-of-domain' refers to texts belonging to the other domain in the context. 'Mixed-domain' refers to text that contains a mixture of data from both the domains.

Our notion of domain changes slightly when we move into the second scenario of forum data translation. In this scenario, we aim to translate potentially noisy user-generated forum content by utilising available parallel training data present in the form of professionally-edited internal corporate documentation. Here the differences between the training data and forum data are mostly in terms of vocabulary, spelling, punctuation and style, although both datasets are from the same technical domain. Hence, in this context the term 'in-domain' refers to any data directly related to or originating from Symantec (this includes both forum content and internal documentations). 'Out-of-domain' refers to any data that is freely available on the web and is not directly related to Symantec. The training data and the forum content being from the same domain in this scenario, we differentiate between them by using the terms 'source-domain' and 'target-domain'. 'Source-domain' refers to the professionally-edited corporate content while the user-generated forum content is labelled as the 'target-domain'. The use of these terms will henceforth refer to their specific meanings in the context of this thesis.

1.2 Research Questions

The primary setting in domain adaptation of SMT systems is one where out-of-domain data is needed to boost the translation performance of an in-domain system with sparse in-domain data. In a related scenario, however, the performance of SMT systems for translating groups of sentences which come from a mixture of different domains also falls into the scope of domain adaptation. Usually, SMT systems trained on in-domain data perform best when translating sentences from the same domain (Axelrod et al., 2011) while the quality drops when translating out-of-domain sentences (Haque et al., 2009). When translating a mixed-domain dataset, an individual sentence from a particular domain might be better translated by a corresponding domain-specific model rather than a generic model trained on the entire heterogeneous training data. At the same time, the overall system should be able to handle translations of sentences from every domain present in the mixed-domain dataset. This situation raises the first research question of this thesis:

> **(RQ1)** *Given a mixed domain and a set of mixed-domain training data, does a combination of translations from different domain-specific models, each trained on a subset of the data, provide better translation quality when compared to those from generic models, trained on the full dataset?*

In **RQ1**, we approach the problem of mixed domain data translation using a combination of different domain-specific systems. However, this approach is only feasible when appropriate amounts of training data for the in-domain models are available.

In the second part of the thesis, we target a real-life scenario where the absence of in-domain data presents a new set of challenges. In order to translate user-generated content from Symantec web-forums[4] in the IT domain, we utilise available professionally-edited parallel corporate content (translations of Symantec internal documentations) in related domains to train the models. Being a part of corporate documentation, the training data is clean, quality controlled and by-and-

[4] http://community.norton.com

large conforming to controlled language guidelines (Doherty, 2012). In contrast the user-generated target data is potentially noisy, loosely moderated and takes liberties with commonly established grammar, punctuation and spelling norms. This difference between the target domain and the training data leads to our second research question:

> **(RQ2)**_In a scenario, such as the translation of user-generated content, where the target domain is different from the training domain, how effective are normalisation and data selection methods in improving translation quality?_

In order to address **RQ2**, we use out-of-vocabulary (OOV) rates as a measure of difference between training and target domains and use normalisation and data selection methods to systematically reduce different categories of OOVs. The data selection methods used to this effect are particularly useful and involve querying out-of-domain parallel corpora with a set of OOVs to select relevant sentence pairs which are combined with the existing training data to improve coverage. While in our experiments we perform the combination at the corpus level, it can also be done at the model level. Investigating the effect of such model-level combination techniques raises the third research question:

> **(RQ3)**_How can multiple models be adapted at different component levels of an SMT system, and what is the effect of component-level adaptation on translation quality?_

Considering the success of data selection methods as a domain adaptation method in the context of our task of forum data translation, we are motivated to investigate this approach further. Observing the trend of data selection methods in the literature (Zhao et al., 2004; Hildebrand et al., 2005; Moore and Lewis, 2010; Axelrod et al., 2011), we find different measures of similarity are used to select supplementary data for domain adaptation of SMT systems. Unfortunately, the most widely used measures such as perplexity or cross-entropy are often found not to correlate with improvements in translation quality of SMT systems (Axelrod, 2006). In order to address this gap in state-of-the-art approaches to data selection methods, we propose to use translation quality directly to select relevant data from out-of-domain datasets. This leads to our fourth and final research question:

> **(RQ4)** *How can translation quality be directly used to select relevant data from an out-of-domain corpus and effectively combine it with in-domain data to drive domain adaptation?*

In order to address **RQ4**, we develop a novel technique of translation-quality-based data selection from supplementary datasets and empirically show how the method outperforms existing techniques in the field in the context of our domain adaptation task.

1.3 Roadmap

In the remaining chapters of this thesis, we seek to address the research questions proposed in Section 1.2. In addition to presenting our experiments and contributions, we also provide the necessary background information on generic techniques and review relevant previous research to make this thesis self-contained. The following paragraphs describe how the remainder of this thesis is organised.

Chapter 2 provides the background to the different techniques and tools used throughout this thesis. Starting with a brief summary of the major paradigms in MT, we briefly sketch the evolution of SMT as the most dominant paradigm today. We provide a brief account of the different flavours of SMT systems before introducing state-of-the-art phrase-based SMT (PBSMT) (Och and Ney, 2003). Since PBSMT is used as the basis of all experiments in this thesis,[5] we describe the core components of a standard PBSMT setup, followed by a discussion on the specific tools and software used. As the main focus of this thesis is domain adaptation in SMT, we present a review of the techniques and methods reported in domain adap-

[5] Henceforth in this thesis, any reference to an SMT system, unless explicitly stated, would mean a PBSMT system

tation research, and align our research questions to the current state-of-the-art. Finally we present a brief description of the datasets, tools and evaluation metrics used in our experiments throughout the thesis.

Chapter 3 presents our experiments aimed at addressing research question RQ1. Using professionally edited corporate documentation-based content from Symantec in two different domains, we simulate the scenario of mixed-domain data translation setting. We empirically show how in-domain models better translate in-domain data and how quality suffers for out-of-domain data. We present a method of combining two domain-specific SMT systems using an automatic classifier to identify the domain of each input sentence, routing each to the most appropriate domain-specific SMT system. We compare the effect of our approach with existing methods in the literature (Koehn and Schroeder, 2007; Du et al., 2010; Foster et al., 2010) and show how our technique provides better translations for a mixed-domain setting, a finding further supported by a manual evaluation of translation quality.

Chapter 4 details our experiments to address RQ2 in the industrially relevant setting of possibly 'noisy' web-forum data translation, given professionally edited 'clean' corporate training data. Quantifying the differences between the training data and the target domain using OOV rates, we categorise the OOVs into specific categories. We devise individual normalisation and data selection techniques to systematically reduce each OOV category and report their effect on translation quality in our experiments. Using two different testsets, we present a comparative study of the effect of normalisation and data selection and provide results from both automatic and manual evaluation to support our claims.

In **Chapter 5**, we study data selection, focussing on the aspect of how out-of-domain data can be combined with in-domain data to achieve the best translation quality. In contrast to traditional corpus-based combination, we present model-based combination experiments using a mixture modelling framework (Hastie et al., 2001). Using different supplementary datasets, we show how model-based combination outperforms corpus-based combination for our task. Furthermore, we also present a comparative study of the effect of component-level adaptation on translation quality in this chapter.

Chapter 6 introduces a novel translation-quality-based supplementary data selection method directly using translation quality as a selection criterion to perform domain adaptation in SMT. Comparing our data selection methods to existing techniques demonstrates how our method outperforms other approaches for our adaptation setting. In addition we present a phrase-table merging method as an alternative means of data combination. A comparison with state-of-the-art approaches of data combination shows that our phrase-table merging approach performs nearly as well as the best method.

Chapter 7 concludes the thesis with overall conclusions from our experiments and some future directions of research.

1.4 Publications

Core parts of the research presented in this dissertation were published in a number of peer-reviewed conference proceedings. Our classifier-based approach to the combination of domain-specific systems is presented in Banerjee et al. (2010). Banerjee et al. (2011b) presents our experiments on data combination approaches and the effect of component-level adaptation on translation quality. Our experiments involving normalisation and data selection methods to translate web-forum content are reported in Banerjee et al. (2012a). Finally Banerjee et al. (2012b) presents our novel approach to translation-quality-based supplementary training data selection.

There are also a few additional publications which are partly related to this thesis and our research on domain adaptation. Our initial experiments in supplementary data selection are reported in Penkale et al. (2010), while domain adaptation experiments in language modelling are reported in Banerjee et al. (2011a) as a part of our participation in the evaluation campaigns in WMT-2010,[6] and IWSLT-2011,[7]

[6] http://www.statmt.org/wmt10/translation-task.html

[7] http://iwslt2011.org/doku.php?id=06_evaluation

respectively.

Publications from the thesis

- Banerjee, P., Naskar, S., Roturier, J., Way, A. and Genabith, J. (2012). Translation Quality-Based Supplementary Data Selection by Incremental Update of Translation Models. In Proceedings of 24th International Conference on Computational Linguistics (COLING-2012), Mumbai, India (To Appear)

- Banerjee, P., Naskar, S., Roturier, J., Way, A. and Genabith, J. (2012). Domain Adaptation in SMT of User-Generated Forum Content Guided by OOV Word Reduction: Normalisation and/or Supplementary Data? In *Proceedings of the 16th Annual Conference of the European Association for Machine Translation (EAMT-2012)*, pages 169–176, Trento, Italy.

- Banerjee, P., Naskar, S. K., Roturier, J., Way, A., and van Genabith, J. (2011). Domain Adaptation in Statistical Machine Translation of User-Forum Data using Component Level Mixture Modelling. In *Proceedings of the Thirteenth Machine Translation Summit (MT Summit XIII)*, pages 285–292, Xiamen, China.

- Banerjee, P., Li, B., Naskar, S. K., Way, A., and Van Genabith, J. (2010). Combining Multi-domain Statistical Machine Translation Models using Automatic Classifiers. In *Proceedings of the Ninth Conference of the Association for Machine Translation in the Americas (AMTA-10)*, pages 141–150, Denver, CO.

- Banerjee, P., AlMaghout, H., Naskar, S. K., Roturier, J., Jiang, J., Way, A., and van Genabith, J. (2011). The DCU Machine Translation Systems for IWSLT, 2011. In *Proceedings of International Workshop on Spoken Language Translation (IWSLT-2011)*, pages 254–260, San Francisco, CA.

- Penkale, S., Haque, R., Dandapat, S., Banerjee, P., Srivastava, A. K., Du, J., Pecina, P., Naskar, S. K., Forcada, M. L., and Way, A. (2010). Matrex: the DCU MT system for WMT 2010. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR, (WMT -10)*, pages 143–148, Uppsala, Sweden.

Chapter 2

Background and Related Work

Machine translation (MT) comprises a family of techniques and algorithms aimed at automatically translating text from one natural language to another. Since Warren Weaver's (1949) first attempt at the problem, a number of techniques and paradigms have emerged in the field of MT over the past six decades. The initial approaches to MT ranged from simple direct translation approaches using rules to directly map input to the output, to more sophisticated transfer methods which used morphological and syntactic analysis. The initial success of these early techniques in MT lead to a considerable research focus on and funding for MT until the publication of the ALPAC report in 1966, whose negative assessment of the field lead to a cut in funding. Continuing MT research focussed on using linguistic rules to perform translation, making rule-based MT (RBMT) the most dominant paradigm until the end of the 1980s. The popularity and success of this paradigm further lead to the development of the first set of commercial RBMT systems like Systran and Météo. MT research continued to focus on using sophisticated linguistic rules to generate advanced transfer-based (Vauquois and Christian, 1985) and interlingua-based (Muraki, 1987) MT systems. The high cost of manually developing rules and the lack of portability of rules across different language pairs motivated subsequent MT research based on alternative empirical approaches. By the end of the 1980's the dominance of RBMT was challenged by the emergence of a new paradigm in MT research– data driven or corpus-based MT.

Data-driven approaches to MT comprise two major strands of research. Based on ideas borrowed from the field of speech processing, Brown et al. (1990) introduced the concept of statistical MT (SMT). At the same time Nagao (1984) used translation examples for MT leading to the emergence of example-based MT (EBMT) approach. Over the past few decades, MT research has been dominated by these two data-driven approaches with SMT being, by far, the most dominant of the two. In order to translate new data, data-driven approaches rely on specific information extracted from pre-existing parallel corpora. Parallel corpora with aligned source and target sentences are required to train a new SMT or EBMT system. Given such training data, an MT system can be automatically, quickly and inexpensively created in contrast to the much more time-consuming and laborious way of handcrafting an RBMT system. SMT being currently the most dominant paradigm in MT research, all our translation experiments presented in this thesis are based on SMT.

Despite the advantages of SMT over other paradigms, SMT models extensively rely on the availability of parallel sentence aligned datasets for training its models. Usually, these datasets are obtained by aligning source texts with translations produced by human translators. SMT models and features are sensitive to the size and quality of the training data and also in many cases to the specific 'domain' of the data (Axelrod et al., 2011). The concept of 'domain' in the context of SMT usually refers to the specific linguistic style, nature and/or vocabulary characteristic of a particular dataset which sets it apart from a generic all-encompassing model of the language. This dependence on the domain of training data leads to the problem of domain adaptation, where an SMT system trained on data from one domain is used to translate data from another domain. The actual motivation of domain adaptation can range from insufficient availability of in-domain data (insufficient for creating a reliable statistical model with reasonable generalisations) or the complete lack of such. The relevance of this problem in a large variety of real-life situations makes domain adaptation an important sub-field in SMT research and the primary focus of our research presented in this thesis.

The rest of the Chapter is organised as follows: Section 2.1 formally introduces the concept of SMT along with techniques and algorithms used to train model components in a standard SMT setup. Sections 2.2 and 2.3 describe the language modelling and decoding aspects in SMT. In Section 2.4 we introduce the problem of domain dependence and adaptation in the context of machine learning and natural language processing followed by a brief introduction of the previous research in domain adaptation in NLP. Eventually we present relevant related research in the field of domain adaptation in SMT followed by a description of the specific scenarios we address in this thesis. Finally Section 2.5 presents the tools, data and automatic evaluation metrics used in our experiments.

2.1 Statistical Machine Translation

The first statistical approach to MT was proposed by Weaver (1949) who was motivated by the success and popularity of statistical techniques in the fields of cryptography and information theory. However limitation of the computing power of machines along with the lack of machine-readable text, during that time, eventually resulted in shifting the focus of MT research to more linguistically motivated approaches. After being in hibernation for nearly four decades, statistical approaches re-emerged in MT, this time motivated by statistical techniques used in speech recognition (Brown et al., 1988, 1990). These statistical methods developed by the speech community were so successful that they presented themselves as a viable alternative methodology to address the MT problem. Moving away from traditional linguistic- and rule-based approaches to MT not only allowed easier and faster development of MT systems, but also opened up the possibility of addressing inherent limitations and problems of RBMT systems at the time. In addition the increased availability of machine readable text and computational resources further helped the development of SMT. In this section we present a brief history of the SMT technology outlining significant developments leading up to the current state-of-the-art. Finally we present the details of an open-source SMT toolkit which has been used in our experiments throughout this thesis.

2.1.1 The Noisy Channel SMT Model

The original approach to SMT was first proposed by Brown et al. (1990, 1993) and was motivated by the noisy channel approach used in speech recognition for representing probabilistic models of pronunciation. In this model, the source sentence S is assumed to be a modified version of the the equivalent target sentence T, where the modification is induced by passing T through the noisy channel. Figure 2.1 depicts a diagrammatic representation of the model. In this formulation, the decoder's job is to retrieve the original target language sentence from which the given source language sentence is generated. Using this model, the translation problem is formulated using Bayes Theorem as presented in Equation (2.1):

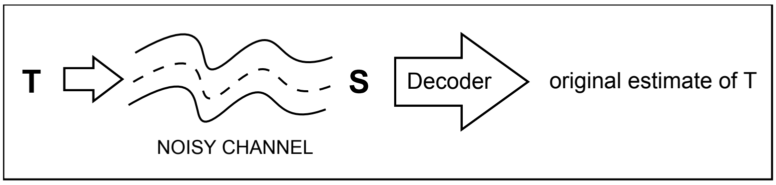


Figure 2.1: Diagrammatic representation of the Noisy Channel model

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

where $p(t|s)$ represents the probability of producing the target language sentence t when translating a source language sentence s. Ignoring the denominator in the equation (since it is independent of t), the most probable translation t can be obtained by maximizing its probability in $p(t|s)$ as is depicted in (2.2):

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

In Equation (2.2), $\hat{t}$ denotes the translation with the maximum probability that is computed from the products of two component probabilities: (i) $p(s|t)$ which represents the conditional probability of the source sentence given its corresponding translation t and is known as the *translation model* probability and (ii) $p(t)$ which denotes the probability of the sentence t in the target language and is otherwise known as the language model probability. The translation model is responsible for assigning the probabilities of the words in the target language sentence t which can be generated by translating words from the source language sentence, thus ensuring the coverage and adequacy of the translation. On the other hand, the language model aims to organise the order of the target language words such that the fluency of the target sentence is maximised. The process of finding the particular translation $\hat{t}$ that maximises the product in Equation 2.9 is known as *decoding*. Hence this formulation presents the problem of machine translation as a search problem where the decoder searches for the most likely translation $\hat{t}$ by maximising the product of translation and language model probabilities. However, the number of possible translations being potentially exponential, a beam-search or pruned Viterbi algorithm is usually used to restrict the search space.

2.1.2 Log-linear Model in SMT

The state-of-the-art models in SMT have gradually shifted from the classical noisy channel based approach (Equation 2.1) to use a log-linear model to compute the translation model directly (Och and Ney, 2003; Zens and Ney, 2004; Zens et al., 2005). Using a log-linear combination allows a weighted combination of M feature functions where each function is derived from a particular component (including translation model or language model) of the SMT system. The log-linear SMT model is presented in Equation (2.3)

$$\hat{t} = \arg\max_{t}\{exp \sum_{m=1}^{M} \lambda_m h_m(t, s)\} \tag{2.3}$$

where $h_m(t,s)$ represents a feature function and λ_m indicates its corresponding weight. This formulation enables the combination of several component models as feature functions, in addition to the already existing translation and language models, within the SMT framework. The ease of integrating additional models into the system thus forms the primary advantage of log-linear models. Additionally each feature function is multiplied by a scaling factor which controls the relative importance of the feature function in the final product. This provides additional flexibility to the existing SMT framework. Note that the noisy channel approach presented in Equation (2.2) can easily be expressed as a special case of the log-linear model with two feature functions ($M = 2$) and each scaling weight set to 1 ($\lambda_1 = \lambda_2 = 1$).

The process of estimating the feature function weights in the log-linear model is known as *tuning* or *parameter tuning.* In this phase, the feature weights are often estimated using a discriminative training method aimed at optimising the translation quality of the system in terms of an automatic evaluation metric on a held-out dataset. This dataset is known as the development set and is expected to be representative of the true (unseen) test data. This tuning mechanism is also known as the *minimum error rate training (MERT)* (Och, 2003) as it tries to minimise the error rate in development set translation setting optimal feature weights in the process.

2.1.3 Word-based SMT

In the initial approach to SMT (Brown et al., 1990, 1993), the basic units of translation were words. With this setting, the translation probability of the target sentence is composed of the product of individual word translation probabilities for each of the constituent words according to (2.4):

$$p(t|s) = C \times \prod_{j=1}^{l_t} p(t_j|s_{a\{j\}}) \tag{2.4}$$

where the source sentence $s = (s_1, ..., s_{l_s})$ having l_s words is translated into the target sentence $t = (t_1, ..., t_{l_t})$ having l_t words and the mapping between the source and target words is denoted by an alignment function a. C is a normalisation constant based on the number of words in the source and target sentences. The most important aspect of training a word-based translation model therefore lies in estimating the alignment function a.

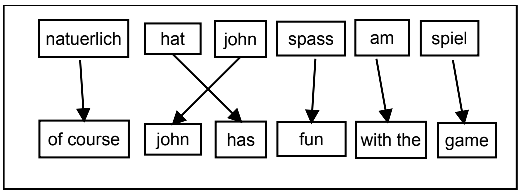


Figure 2.2: An example of word alignment adapted from Koehn (2010)

A translation model is usually trained on a large bilingual sentence aligned corpus. The process of estimating the alignment function from the training data is known as *word alignment* and is one of the central problems in SMT. In order to estimate the alignment, first the correspondences between words in the source and target language sentences need to be defined. Figure 2.2 shows an example alignment for a German–English sentence pair with the lines indicating alignment between the German and English words in the sentence. As is evident from the example, a single word in a language might be aligned to multiple words in the other language (e.g. the German word *natuerlich* aligns to two words *of course* in English). There might also be a case where a word in the source language sentence does not align to any word in the target language sentence. In such a case the source word is said to align to a special 'NULL' token. Different aspects of word-alignment are often modelled using a family of statistical word-alignment models collectively known as the IBM models (Brown et al., 1990, 1993).

Using alignment models automatically introduces a concept of re-ordering of the words in the translated sentence. However, IBM model 1– the most basic word alignment model only relies on lexical translation probabilities (independent word to word translation probabilities) to generate translations of a source sentence. Under this model, all possible re-orderings of the target words are considered equally likely. This aspect of every reordering being equally likely is handled in IBM model 2 whereby an explicit re-ordering model based on relative positions of aligned words between the source and the target sentence is introduced. IBM model 3 addresses the aspect of *fertility* in translation. *Fertility* is the notion of a single word in the source language getting translated into multiple words in the target language. IBM model 4 introduces the concept of relative distortion, whereby the placement of the translation of a source word is guided by the placement of translation of the preceding source word. Finally the last model in the family, IBM model 5 addresses the issue of *deficiency* by keeping track of the vacant positions in the target language sentence and using them to restrict positioning of new target words.

A number of different techniques exist to estimate the word-to-word correspondence between source and target starting from a sentence aligned parallel training corpus. The most commonly used toolkit for extracting the word alignments from bilingual training data is GIZA++ [1] (Och and Ney, 2003) which utilises the expectation-maximisation (EM) algorithm (Dempster et al., 1977) to statistically estimate the different word alignment models.

Despite the IBM models capturing different aspects of translation, the word-based models had serious drawbacks in terms of the quality of translations. Devoid of any syntactic information, these models had limitations in capturing linguistic aspects of multi-word units in translation and non-local dependencies. The ordering of the words in the target language is only influenced by the relatively weak distortion probabilities and most work of ensuring the grammaticality of the target sentence is left to the language model. All these limitations motivated researchers to move into using phrases instead of words as the basic units of translation leading to a new paradigm– phrase-based SMT.

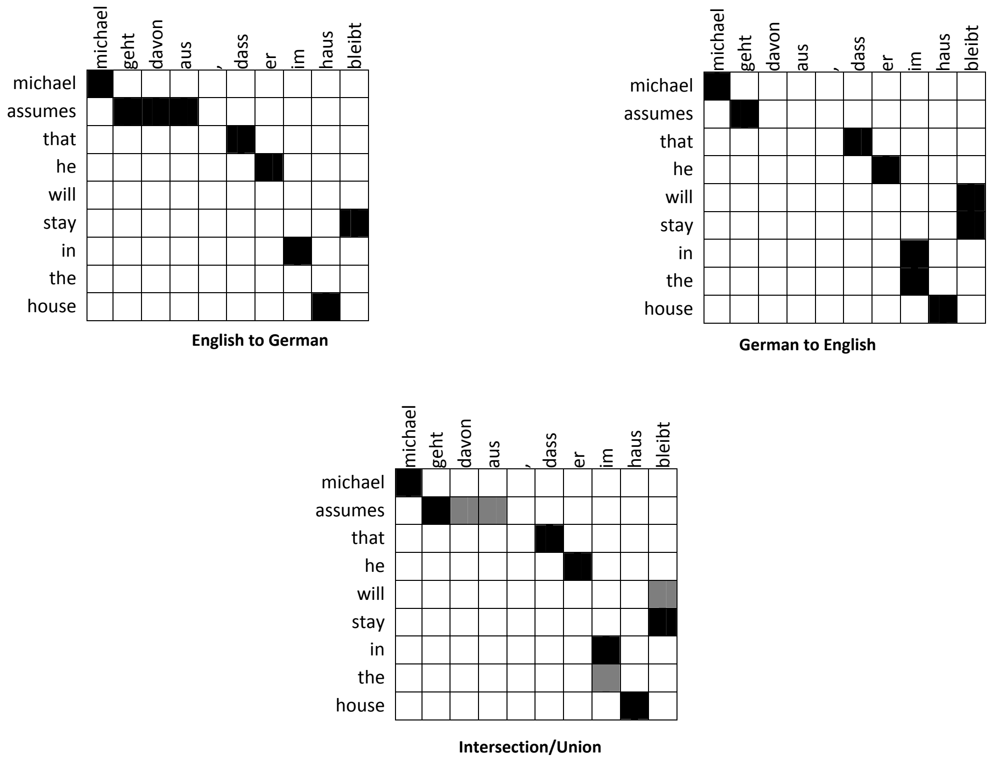


Figure 2.3: Extracting phrase alignments from word alignments for a German–English sentence pair.

2.1.4 Phrase-based SMT

The drawbacks of word-based SMT models lead to the development of phrase-based SMT (PBSMT) models (Och and Ney, 2003) which use phrases as the basic translation units. In PBSMT, phrases are just contiguous chunks of text, and are not linguistically motivated. The first step of training a PBSMT model involves identifying source and target phrases from the parallel training data and identifying the alignments between them. A method outlined by Och and Ney (2003) uses phrasal extraction heuristics based on initial word alignments to achieve a mapping between sequences of n words in the source language (source phrases) to m words in the target language (target phrases). In the first step, word alignments are extracted for both directions, source–target and target–source. Since IBM model based word alignments only allow one-to-one or one-to-many mappings, performing this in both directions provides a set of alignments containing one-to-one, one-to-many

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

and many-to-many alignments. Figure 2.3 represents an example of such alignments between an English–German sentence pair. The two matrices in the top show English–German and German–English alignments while the matrix below show the intersection (black) and the union (grey) of these alignment sets. This figure has been adapted from Koehn (2010). Initially only the intersection of alignments from both sets are considered to maximise precision. Starting from the highly precise alignments (since these alignments are common to both directions), the intersection set is extended to the union of two sets by iteratively adding adjacent alignments[2] present in the alignment matrix (Koehn et al., 2003). This technique allows the gradual build-up of contiguous phrase alignments. In the final stage, all the remaining alignments in the alignment union set are added to the final alignment to improve recall.

Using this final alignment set, all possible phrase-pairs are extracted providing the intended set of phrase alignments. The process is repeated for every sentence pair in the parallel training data and all possible phrase-pairs along with their associated probabilities are accumulated in the translation model. The conditional phrase translation probabilities between a source–target phrase pair p_s, p_t are estimated from relative frequencies of the phrase pairs according to Equation (2.5):

$$p(p_s|p_t) = \frac{C(p_s, p_t)}{C(p_t)} \tag{2.5}$$

In principle phrasal extraction can result in phrases of arbitrary length, even incorporating entire sentences. To manage the scalability of the translation model and to ensure efficient usage of computing resources, the phrase length is often restricted to a specific number of words.

[2] Adjacent alignments in the current context refers to the horizontal and vertically adjacent alignments in the two-dimensional bitext grid space (Figure 2.3).

2.2 Language Models

So far we have presented different techniques aimed at estimating the translation model component in an SMT system. The other important component within the noisy channel framework (Equation 2.1) or the log-linear framework (Equation 2.3) of SMT models is the language model. A language model aims to compute the probability of a sequence of words in a particular language trying to assign higher probabilities to syntactically correct word arrangements (according to the language) and lower probabilities to ill-formed word sequences. Using generative modelling, a language model tries to estimate the probability of a string $t = t_1, t_2, t_3, ...t_n$ as the product of individual probabilities of each of the constituent words ($t_1, t_2...t_n$ etc.) given their corresponding history (i.e. all the words preceding the current word in the string) as in (2.6).

$$p(t) = p(t_1, t_2, t_3, ...t_n) = p(t_n|t_1t_2t_3...t_{n-1}) \times p(t_{n-1}|t_1t_2t_3...t_{n-2})...... \times p(t_2|t_1) \times p(t_1) \tag{2.6}$$

However, estimating the probabilities of every word given their complete history is clearly non-trivial for any string of reasonable length as eventually there would be too many histories to consider. Hence, this probability is approximated by estimating the probability of the current word conditioned only on the preceding $n-1$ words. A language model built on this approximation is known as the n-gram language model. For most real-life applications the value of n ranges from 2 to 5. The actual probabilities are estimated from the relative counts of specific n-grams as described in equation (2.7):

$$p(t_n|t_1t_2...t_{n-1}) = \frac{C(t_1t_2....t_{n-1}t_n)}{\sum_t C(t_1t_2....t_{n-1})} \tag{2.7}$$

No matter how large a corpus is used to train an n-gram language model, we would always encounter data sparseness issues especially for higher order n-gram estimation. Since the training corpus is always finite while the entire set of possible n-grams in a language is infinite, some of the n-grams will have zero probabilities assigned to them (since they are unseen in the training data hence the count is 0) which would eventually lead to a zero probability for the entire string. A specific set of techniques employed to avoid the phenomenon of assigning zero probabilities to unseen n-grams are collectively called *smoothing* methods. Different smoothing methods exist in the literature ranging from the simple add-one smoothing (Lidstone, 1920) to more sophisticated methods of weighted linear interpolation backoff (Jelinek and Mercer, 1980). The Modified Kneser-Ney smoothing (Kneser and Ney, 1995) which is based on the weighted interpolation approach but employs absolute discounting on the higher-order n-gram probabilities, is generally found to outperform other smoothing methods (Chen and Goodman, 1996) and hence is the chosen method for all our experiments.

2.3 Decoding

As already stated in Section 2.1.1, given the language model and translation model probabilities, the decoding process searches for the best translation which maximises the product of both these probabilities. In other words the job of the decoder is to search through all possible translations which are likely to have produced the source sentence, and select the particular target translation that is the most likely. However, searching through all possible target translations is not feasible in practice due to the large number of possible translations for an input sentence generated by the SMT model. Therefore, in order to make the process efficient and implementable a beam-search strategy is usually employed for decoding in SMT systems (Germann et al., 2001).

The Moses decoder (Koehn et al., 2007) which is a part of the open-source state-of-the-art Moses SMT toolkit[3] implements the beam-search strategy in its decoding phase. In addition to the previously discussed phrase-based translation and language models, Moses uses an additional *reordering model* to capture the reordering of

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

the target phrases in the final translation. The reordering of the target phrases is modelled by a relative distortion probability distribution based on the start position of the source phrase that is translated into the current target phrase and the end position of the source phrase corresponding to the preceding target phrase. This reordering model probability in combination with the phrase translation probability and language model probability is used to estimate the quality of a translated phrase during decoding.

For translation, the input sentence is segmented into phrases. All these segmentations are considered equally probable. Based on the phase alignments, for each such source phrase multiple translation phases can be applied. Each such translation phrase is known as a *translation option*. Before the actual decoding takes place, all translation options are collected for the input phrases.

The target language sentence is generated from left to right in the form of hypotheses. Each such hypothesis is associated with the concept of a cost which is computed from the product of the phrase translation model, language model and distortion model probabilities. The concept of cost is analogous to probability with the two being inversely related (i.e. a higher cost indicates lower probability). The search process starts with an initial empty hypothesis (with cost $= 1$) where no source input words are translated and no target words have been generated. At every step new hypotheses are generated by attaching the phrasal translation of yet untranslated source phrases to the current hypothesis. The cost of the new hypothesis is computed by multiplying the cost of the current hypothesis by the translation, distortion and language model costs of the added phrase translation. Finally once all the input source words are covered the hypothesis with the minimum cost (and hence maximum probability) is chosen as the best translation for the input.

During the search process, the Moses decoder employs a priority queue in order to maintain the partial hypotheses in stacks based on the number of source words covered by the hypotheses. Going through each hypothesis in a stack the search process extends them into new hypotheses and place them on the appropriate stack.

If the size of a stack grows beyond a limit, stacks are pruned using histogram pruning where only the top n scoring partial hypotheses are retained. Furthermore, to restrict the size of the search space, weaker hypotheses are pruned based on their current cost multiplied by an estimated future cost. The future cost is estimated on the basis of the translation model and language model probabilities of the remaining sequence of untranslated words. Since multiple overlapping translation options might exist for the remaining source words, the future cost estimator selects the option with the least cost i.e. the translation option with the highest probability.

2.4 Problem of Domain Dependence

The primary objective of NLP is to create systems that can perform the task of understanding or producing natural language the way humans do. In order to achieve this broad goal, NLP systems often focus on specific language oriented tasks such as part-of-speech tagging, named-entity recognition, sentiment analysis, natural language parsing or natural language translation, to name a few. To create such systems, NLP often relies on *supervised machine learning* (ML) algorithms to learn or train models which perform the specific task under consideration. Supervised ML algorithms require *annotated training data* to learn or infer these models. For example, in part-of-speech tagging, the training data usually comprises of sentences (training instances) where every word is annotated with the corresponding part-of-speech tag (class labels). Eventually the trained model is evaluated on held-out *test data* to measure its performance or generalizability on unseen data.

Depending upon the availability of annotated training data, ML algorithms can broadly be divided into three categories– (i) *Supervised ML* wherein the training instances are annotated with the actual class labels (by human annotators). (ii) *Unsupervised ML* where the training instances are not annotated and hence are not associated with the actual class labels and finally (iii) *Semi-supervised ML* where the training data comprises both labelled and unlabelled instances with the unlabelled instances far outnumbering the labelled instances (since labelling by human annotators is costly). Irrespective of the approach, the basic assumption in ML approaches is that the *training* data comprises instances that are independently sampled from an underlying distribution and that the *test* data follows the same underlying distribution. This assumption obviously does not hold true whenever the underlying distribution of the test data differ from that of the training data. This violation of the basic ML assumption causes the problem of domain dependence in ML (and NLP) systems in general.

A model trained using any supervised ML approach is heavily dependent on the training data. The estimated model parameters best reflect the characteristics of the training data. Therefore, if the characteristics of the test data are substantially different from that of the model parameters, the performance of the systems drops. For example McClosky et al. (2010) shows how the performance of a statistical parser trained on Penn Treebank Wall Street Journal (newspaper text) drops drastically when tested on datasets from fictional/non-fictional literature or biomedical texts. As previously mentioned in Chapter 1 (Section 1.1), texts may differ along many dimensions related to domain, topic, style, genre, register etc. This loss of performance in ML systems due to domain variation between training and target domains is referred to as the problem of domain dependence in ML. This problem being inherent to the basic assumptions of ML, appears in nearly all ML-based NLP tasks including SMT.

There are two generic approaches towards solving the problem of domain dependence:

1. Manually annotate data for the new domain.

2. Domain Adaptation: Adapt the model trained on a specific source domain to a new target domain.

Manually annotating data for every new domain is clearly expensive and a non-elegant solution to the problem of domain-dependence. In comparison, domain adaptation aims to adapt the already trained model (on some source domain) by exploiting either limited amounts of labelled data or large amounts of unlabelled data from the target domain. In this section we will introduce the basic approaches to domain adaptation in NLP, focussing extensively on the adaptation approaches used in the context of SMT.

2.4.1 Approaches to Domain Adaptation

Based on the three general ML approaches, domain adaptation techniques discussed in the literature can also broadly be divided into three major categories:

1. Supervised Domain Adaptation (Hara et al., 2005; Daume III, 2007)

2. Unsupervised Domain Adaptation (Blitzer et al., 2006; McClosky et al., 2006)

3. Semi-supervised Domain Adaptation (Daumé et al., 2010; Chang et al., 2010)

These three approaches differ primarily in terms of the type of data available for the new target domain. In *supervised domain adaptation*, large amounts of labelled source data and a limited quantity of labelled target domain data are available. The objective is to leverage the limited amount of labelled target domain data along with the abundant source domain data to create a model which performs well in the target domain. In contrast, in the *unsupervised adaptation* scenario, only large quantities of unlabelled data are available in the target domain with the objective of exploiting this data to adapt the source domain model to the target domain. The *semi-supervised adaptation* approach utilises both labelled and unlabelled data (the amount of unlabelled data being much greater than that of labelled data) in the target domain to adapt models.

Domain adaptation is a relevant problem in multiple research areas and hence related work can be found in different research fields (Plank, 2011). A large part of previous work in domain adaptation stems from the fields of core machine learning and natural language processing. The problem of domain adaptation has been studied under different names, particularly in the field of text classification including *class imbalance* (Japkowicz and Stephen, 2002), *covariate shift* (Shimodaira, 2000) and *sample selection bias* (Heckman, 1979). Additionally a few ML problems closely related to that of domain adaptation have also been investigated including *multi-task learning* (Caruana, 1997) as well as *semi-supervised learning* (Zhu, 2005; Chapelle et al., 2006).

Previous work on supervised domain adaptation involves the use of the abundant source-domain data as a prior when estimating a model on limited target domain data. This idea was examined by Roark and Bacchiani (2003) in order to adapt a PCFG parser, while Chelba and Acero (2006) successfully applied the same approach to the task of automatic capitalisation. Daume III (2007) introduced an alternative approach to supervised domain adaptation by altering the feature space using the proposed *easy adapt* algorithm. Using a simple pre-processing step every feature in the general feature space is replicated to a produce a source and a target-specific version. This transformation of the feature space allows any generic learning algorithm to identify which features are best suited to be transferred between the source and target domains. Daume III (2007) used this approach to achieve decreased error rates for different data sets and tasks (e.g. named entity classification, PoS Tagging). In contrast to changing the feature space, an alternative approach is based on modifying the instance distribution using the technique of *instance weighting* (Jiang and Zhai, 2007) for the tasks of PoS tagging, named entity classification and spam filterring. By weighing the source training instances based on their similarity to target domain distribution, this approach was shown to be effective in domain adaptation of multiple NLP tasks.

In contrast to supervised methods, unsupervised methods in domain adaptation utilise unlabelled training data in the target domain in addition to limited amounts of labelled data in the source domain to perform domain adaptation. The most commonly used unsupervised domain adaptation method is called *bootstrapping* where a baseline model trained on the labelled source domain data is used to label the target domain data. This newly labelled target domain data is then combined with the originally labelled source domain data to train a new model, and this process might be iterated until the necessary accuracy is reached.[4] *Self-training* is a popular bootstrapping method in domain adaptation where a model is used to label the data for itself. This approach was used by McClosky et al. (2006) to improve the parsing accuracy of a PCFG parser when used in combination with a discriminative parse reranker. Reichart and Rappoport (2007) further used the same technique without the reranking to improve parsing performance. More recently, Sagae (2010) investigated the effect of self-training (with or without reranking) on the task of semantic role labelling. Another flavour of the bootstrapping method is known as *co-training* where instead of using the same model to label new data, two or more models are involved with one model labelling the data for another model. Steedman et al. (2003) used co-training for bootstrapping statistical parsers using a PCFG parser and a lexicalised tree adjoing grammar parser as the two core systems. Blitzer et al. (2006) introduced the concept of *structural correspondence learning* (SCL) to exploit unlabelled data across both source and target domains to learn a common feature representation that are valid for both domains. They used the SCL technique to achieve domain adaptation in two different tasks of PoS tagging and sentiment classification.

While most of the previous work on domain adaptation has focussed on learning algorithms that return single class classification, some work has been done in the area of ensemble learning to combine multiple models to construct a complex classifier for the classification problem. One of the most common techniques used in this area is mixture modelling (Hastie et al., 2001). Daumé and Marcu (2006) used a mixture modelling approach for domain adaptation using three mixture components– one shared by both the source and target domains with two others which were specific to

[4] Note that this particular approach is referred to as semi-supervised learning in the general ML literature. However in domain adaptation this specific setting is known as unsupervised domain adaptation to differentiate it from the semi-supervised adaptation where both labelled and unlabelled data are available on the target domain.

the source and target domains, respectively. While their three component approach required labelled data for both source and target domains, Storkey and Sugiyama (2007) used a more general mixture modelling approach where the target domain specific component was not used, thus allowing them to work without labelled data in the target domain. The component weights for the model were trained using the expectation maximisation algorithm (Dempster et al., 1977).

2.4.2 Domain Adaptation in SMT

So far we have briefly reviewed previous research in domain adaptation for general NLP tasks. In the context of SMT, domain adaptation involves designing techniques and algorithms that allow models trained on data from a particular domain to successfully translate sentences from other domains. In general the domain adaptation approaches in SMT are training-centric, where the training process is altered or tweaked to perform the adaptation. An alternative stream of approaches is to perform the domain adaptation in the tuning process (MERT) by tuning an out-of-domain system with domain-specific development sets. Domain adaptation in SMT being the primary focus of our research, we provide a discussion on the related work in this area presented in the literature to-date. In the course of describing the current state-of-the-art in domain adaptation research, we also point out how our research questions are aligned to the state-of-the-art.

The initial technique of domain adaptation was imported to SMT from the field of speech recognition. Topic dependent modelling (Carter, 1994) and domain adaptation were extensively applied on statistical models of speech recognition especially for language model adaptation (Iyer et al., 1997). The first application of domain adaptation in SMT was reported by Langlais (2002) who integrated domain-specific lexicons into the translation model resulting in Word Error Rate (WER) reduction on the testset. This was also the first work to empirically prove that general purpose SMT engines (with models trained on general purpose training data) perform poorly when translating domain-specific texts, due to poor translation of domain-specific terms and presence of out-of-vocabulary units. Wu and Wang (2004) and Wu et al. (2005) incorporated domain adaptation for the word alignment model aiming to improve domain-specific word alignments in a situation where limited domain-specific data was available. Combining word alignments from large out-of-domain and smaller in-domain datasets, they achieved considerable improvements in both alignment precision and recall.

Eck et al. (2004) introduced the use of information retrieval theories to propose a language model (LM) adaptation technique for SMT following the approach of Mahajan et al. (1999) in speech recognition. Term-frequency/Inverse document frequency (tf/idf) similarity was used to retrieve documents and sentences from large out-of-domain data collections. Using both story-level and sentence-level retrieval, they observed improvements in language model perplexity as well as translation quality as measured by the NIST (Doddington, 2002) automatic evaluation metric. Their approach was further refined by Zhao et al. (2004), who created a set of structured queries based on the SMT output hypotheses to extract related data from monolingual out-of-domain corpora. Individual language models created on the selected datasets were interpolated with a generic language model to effect the adaptation, leading to significant improvements in translation quality. Hildebrand et al. (2005) utilised Eck's approach to select those sentences from the training data which are similar to the testset sentences and create a translation model on them. They further used language model perplexity to re-rank the retrieved sentences and determine the optimal number of sentences to be retrieved. Combining translation and language model adaptation they reported significant improvements in translation performances with respect to the baseline systems.

Hasan and Ney (2005) proposed a method for building class-based language models by clustering sentences using regular expressions and interpolating them with global language models using mixture models for model combination (Iyer and Ostendorf, 1999). This approach lead to improvements in terms of perplexity reduction and error rates in MT. This work was further extended by Yamamoto and

Sumita (2008) as well as Foster and Kuhn (2007) to include translation models. Using entropy-reduction based clustering techniques (Carter, 1994) on the bilingual training data, automatic clusters were created and each cluster was treated as individual domains (Yamamoto and Sumita, 2008). Using these domain-specific models to translate sentences resulted in improvements in translation quality. Foster and Kuhn (2007) investigated a broad set of adaptation techniques involving discriminative combination of domain-specific models for translation as well as language models. Representing mixture weights as a function of distance between the training and test data allowed dynamic adaptation of existing models to unseen test data. Finch and Sumita (2008) also used probabilistic mixture weights to combine multiple models for interrogative and declarative sentence translation. However, instead of using the distance based measure of Foster and Kuhn (2007), they used a probabilistic classifier to identify class-membership and used the same to determine mixture weights. Civera and Juan (2007) further suggested a mixture adaptation approach to word alignment, generating domain-specific Viterbi alignments to feed a state-of-the-art phrase-based SMT system. Bulyko et al. (2007) explored the use of unsupervised adaptation and discriminative estimation of language model weights, optimised with respect to translation scores to achieve language model adaptation. Instead of using perplexity minimisation on the held out development set, they used Powell's hill climbing algorithm on the n-best lists to minimise translation edit rate (TER) (Snover et al., 2006) directly with respect to a development set resulting in small improvements in translation quality. Using linear interpolation of translation models between a small in-domain model and a out-of-domain model on selected data was carried out by Yasuda et al. (2008). Using an average perplexity score on monolingual language models, relevant data was selected from out-of-domain datasets. Individual models trained on these datasets were combined with in-domain models using linear interpolation to achieve significant improvement in translation quality while considerably reducing the translation model size.

Integrating an in-domain language model with an out-of-domain one using the log-linear features of an SMT model was carried out by Koehn and Schroeder (2007). This work also saw the first use of multiple decoding paths for combining multiple domain-specific translation tables within the framework of the Moses decoder (Koehn et al., 2007). The same idea was explored using a different approach by Nakov (2008) using data-source indicator features to distinguish between phrases from different domains within the phrase tables. Later, Lim and Kirchhoff (2008) proposed a method of out-of-domain data incorporation through phrase generalisation to improve Italian–English translation quality, reporting a noticeable improvement in the process.

Automatic selection of in-domain bilingual data from comparable corpora to enhance existing in-domain bitext was explored by Munteanu and Marcu (2005). Using a maximum entropy classifier, they extract reasonable translations from a related domain comparable (non-parallel) corpora and add them to the small in-domain parallel data to improve translation quality. The use of semi-supervised transductive learning as means of domain adaptation in SMT was proposed by Ueffing et al. (2007a). Testset sentences were repeatedly translated by a baseline system and some of the translations (selected based on a quality threshold) were added back to the training data to eventually improve translation quality significantly. This approach also used confidence estimation and importance sampling techniques for selecting the appropriate translations. Furthermore, the selected parallel data was added to the existing data using both concatenation and mixture modelling. Wu et al. (2008) later used this technique to perform domain adaptation for SMT in a setting where in-domain bilingual data was absent. The in-domain dictionaries were adapted based on different probability distributions (uniform, constant and corpus-probability). Both language models and translation models were combined using both linear and log-linear interpolations with the log-linear approach slightly outperforming the linear one. Gahbiche-Braham et al. (2011) further reported a bootstrapping strategy using an existing SMT engine to detect parallel sentences in comparable data and provide an adaptation corpus for translation model adaptation.

Xu et al. (2007) used language models and information retrieval approaches to classify the input test sentences based on the domains, and translated the same using a generic translation model along with a domain dependent language model. This effort resulted in a significant improvement in domain-dependent translation when compared to domain-independent translation. More recently, Bertoldi and Federico (2009) experimented with using in-domain monolingual resources to improve translation quality achieving considerable improvements. They used domain-specific baseline systems to translate in-domain monolingual data, creating a synthetic bilingual corpus and using the same for adapting the SMT system, which resulted in better performance.

Axelrod et al. (2011) report a new technique of supplementary data selection based on difference of cross-entropy of sentences on in-domain and out-of-domain data. Compared with the common approach of perplexity-based data selection, their approach performs better consistently across different datasets and combination techniques. Daume III and Jagarlamudi (2011) investigate the aspect of OOV reduction in a domain adaptation setting. Mining supplementary datasets for translations of unseen words and incorporating the same directly to existing phrase tables provides them with a statistically significant improvement. Lavergne et al. (2011) reported a mixture model based adaptation strategy using n-code (M. Crego et al., 2011) – their in-house implementation of the bilingual n-gram approach to SMT. Using linear interpolation of static n-gram language models and log-linear interpolation for adapting the bilingual language models along the lines of Foster and Kuhn (2007), they reported moderate improvements in translation quality, with linear interpolation outperforming log-linear mixture adaptation. More recently, Sennrich (2012b) presented a translation model adaptation technique using perplexity minimisation to effectively set the interpolation weights in a linear interpolation setting. In addition this paper also presents experiments optimising perplexity independently for each of the features of a Moses-based translation model. Although interpolating translation models using weights set by perplexity minimisation allows modest improvements over the baseline scores, it is shown to be particularly successful when scaling the number of models to combine from 2 to 10. In Sennrich (2012a), the perplexity minimisation method was further applied to combine translation models created by unsupervised clustering of the training data. Combining individual models trained on unsupervised clusters was found to provide minor performance boost in comparison to training a single model on the combined data.

2.4.3 Alignment of Research Questions to Domain Adaptation Research

Considering the overall trend of domain adaptation research in SMT, we observe that 'relevant' data selection from out-of-domain corpora and model combination form the two major research interests in the field. Varied techniques in data selection ranging from information retrieval methods to perplexity-based ranking have been reported in the literature both for translation model as well as language model adaptation. Mixture models have been particularly popular as a model combination technique in domain adaptation research, in addition to the standard method of data concatenation. Data selection and model combination methods have traditionally been used to boost sparse in-domain models with out-of-domain or related-domain data. While this is the primary objective of domain adaptation, there are related scenarios where adaptation is required to handle translations of mixed-domain data. Some research (Xu et al., 2007; Yamamoto and Sumita, 2008; Sennrich, 2012a) reported in the literature, has investigated this particular aspect of domain adaptation, although it has not been as widely studied as the data selection methods. Our first research question RQ1 (cf Chapter 1) thus aims at addressing this particular adaptation scenario using a combination[5] of independent SMT systems to translate mixed-domain data.

In the second phase of our work, we have focused on the scenario of user-

[5] Note that 'combination' in this context does NOT refer to traditional methods of model combination with mixture models (Foster and Kuhn, 2007) or multiple translation models (Koehn and Schroeder, 2007).

generated web-forum translation for the Symantec web-forums.[6] Web forums are rich sources of information for the tools and products offered by Symantec and offers the company a platform to directly interact with its customers on a daily basis. Furthermore, these forums have become an effective alternative to traditional approaches of customer service (Mitchell and Roturier, 2012), making them an invaluable resource to the company. Being a multinational company with customers from all over the world, Symantec hosts its web forums in multiple languages but most of the content is siloed in individual language-specific forums. This information imbalance across language-specific forums combined with their importance in the business forms the primary motivation behind considering this scenario for investigation. Moreover, the problem of forum-content translation, despite being a relevant one, has not received much attention in the SMT literature (Flournoy and Rueppel, 2010). The major challenge in translating forum content is in the lack of parallel 'forum-style' training data which could be used for training the SMT models. In order to work around this, we use parallel corporate content (internal documentation from Symantec) as the training data for our SMT models. Although the training and the target domains are broadly the same, there is considerable differences in the style, vocabulary and nature of the two datasets. The corporate content which forms the training data is guided by strong controlled language guidelines (Doherty, 2012) and is professionally edited to ensure clean and noise-free text. Compared to professionally edited text, user generated forum data is often more noisy, taking some liberty with commonly established grammar, punctuation and spelling norms (Mosquera and Moreda, 2011). We quantify this difference in terms of the number of out-of-vocabulary words (OOV) (words present in forums but not in training data) and use different normalisation techniques to systematically reduce their number in the test data. Normalisation is an effective technique to reduce OOV rates (Yvon, 2010). Our second research question, RQ2 is aimed at investigating the effect of such normalisation and data selection techniques in the current scenario.

[6] http://community.norton.com/

Although OOV reduction as a domain adaptation approach has been reported by Daume III and Jagarlamudi (2011), data normalisation has seldom been successfully used in domain adaptation. RQ2, is thus aimed at addressing this particular gap in domain adaptation research for noisy data translation.

Our third research question focuses on the effect of data or model combination on translation quality of noisy forum content. We compare the effect of model combination using mixture modelling to that of standard data concatenation on translation quality of forum content. While mixture modelling has been used in combining models from in-domain and out-of-domain data (Foster and Kuhn, 2007; Civera and Juan, 2007; Sennrich, 2012b) none of them considers translating text (user-generated forum data) different in nature from the training data. The experiments aimed at answering RQ3 thus aims at addressing the effect of mixture adaptation on translation quality of noisy forum data.

Our final research question aims at finding a new technique of data selection for our specific scenario of forum data translation. Selecting 'relevant' data from supplementary data sources has been strongly motivated and widely practised in domain adaptation research. However, the criteria for data selection are mostly based on monolingual metrics such as tf/idf (Eck et al., 2004; Zhao et al., 2004) or perplexity (Hildebrand et al., 2005; Yasuda et al., 2008), or in some cases bilingual metrics such as difference in cross entropy on in-domain and out-of-domain data (Axelrod et al., 2011). Despite the use of perplexity as a measure of relevance of out-of-domain training data, perplexity reduction has been shown not to correlate with translation quality improvement (Axelrod, 2006). In answering RQ4, we therefore implement a data selection method based on actual translation quality as evaluated by automatic evaluation metrics on the development set.

2.4.4 Related Work on Associated Research Areas

In the course of our experiments in domain adaptation, we have used a number of associated technologies to assist our core objective. In the first phase of our work (cf. Chapter 3), we used a Support Vector Machine (Joachims, 1999) based sentence classifier to identify the domain of the input sentence to be translated. Subsequently in the third phase (cf. Chapter 5), we used mixture modelling (Hastie et al., 2001) to combine multiple components within the SMT framework. In this section, we provide a brief background of the relevant research for these two technologies.

The problem of text classification involves assigning labels to unlabelled text based on models learnt from labelled text. This is one of the classical applications of supervised learning. Support Vector Machine (SVM) classification algorithms have been shown to outperform other well-established classification methods like Naive Bayes (Good, 1965) and Decision Trees (Chickering et al., 1997) for the task of binary text classification (Dumais et al., 1998). Research has also revealed that SVM scales well with decent performance on large datasets (Kwok, 1998). There are non-linear extensions to the SVM classifier but Yang and Liu (1999) show the linear kernel to outperform non-linear kernels for the task of text classification. This motivates the use of an SVM classifier with linear kernel for our task of text classification as reported in Chapter 3.

Mixture Modelling (Hastie et al., 2001), a well-established technique for combining multiple models, has been extensively used for language model adaptation, especially in speech recognition. Iyer and Ostendorf (1999) used this technique to capture topic dependencies of words across sentences within language models. Cache-based language models (Kuhn and De Mori, 1990) and dynamic adaptation of language models (Kneser and Steinbiss, 1993) for speech recognition successfully used this technique for sub-model combinations. Mixture models have also been used in SMT to combine multiple domain-specific models as reported in Foster and Kuhn (2007), Sennrich (2012b) and Civera and Juan (2007) which motivates the usage of the technique to combine component-level models in our case.

2.5 Tools, Data and Evaluation Metrics

The previous sections introduced the basic concepts in SMT as well as relevant related work in the field of domain adaptation. In this section we introduce the tools and techniques that have been used to train, tune and test the SMT models in our experiments throughout this thesis. Furthermore, we also provide a brief introduction of the datasets used in our experiments and the automatic evaluation metrics used to measure the quality of translations.

2.5.1 SMT Toolkits and Techniques

All the SMT models used in our experiments reported in thesis are based on the Moses Toolkit (Koehn et al., 2007)– a widely used open-source implementation of phrase-based SMT. Moses utilises a log-linear implementation of the SMT as is explained in Section 2.1.2, using eight different features in its standard configuration. The translation model in Moses is implemented using a data structure known as the *phrase-table*, which contains the source and the target phrase pairs along with the following five features for each phrase pair:

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

2. The inverse lexical weights $\phi(s|t)$ and the direct lexical weights $\phi(t|s)$ computed by taking an average of the word-level translation probabilities (lexical translations) over the best alignment for each phrase pair

3. A phrase penalty having a constant value of 2.718 ($exp(1)$) such that longer phrases are favoured during the decoding phase

In addition to this, the Moses decoder uses three more features during the decoding phase for estimating hypothesis costs (cf. Section 2.3):

1. A language model score based on the n-gram language model used in the model (cf. Section 2.2)

2. A distortion penalty to limit reordering of the target phrase-pairs

3. A word penalty $w(t) = exp(length(t))$ to balance the language model's bias towards short sentences.

A detailed description of the Moses features can be found in Koehn (2010). While additional features have often been added to the Moses framework as part of different research outcomes, these eight features have been found to perform consistently well within the log-linear framework.

We use GIZA++ [7] (Och and Ney, 2003) for computing the word-alignments from the bilingual sentence aligned training data (cf. Section 2.1.3). The phrasal alignments are computed from the word alignments using the training scripts associated with Moses[8] using the 'grow-diag-final' heuristic. The maximum phrase length is set to 7 for all our experiments. The weight of the log-linear features are estimated using the MERT algorithm (Och, 2003) on a held out development set maximising BLEU. The Moses SMT toolkit along with these specific configuration settings have consistently been used for all our experiments in the thesis. Since the same configuration is used in all of our experiments, we refrain from repeating this information in every chapter, and only focus on variations of the configuration specific to the experiments in respective chapters. For all our language modelling requirements we use the open-source IRSTLM language modelling toolkit (Federico et al., 2008). The language models used in our experiments comprise 5-gram models with Modified Kneser-Ney smoothing and interpolated backoff (Kneser and Ney, 1995). The computation of perplexity values for full datasets or individual sentences are also carried out using the tools and scripts associated with IRSTLM. The estimation of mixture weights used in Chapter 5 for language model and translation

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

[8] http://www.statmt.org/moses/?n=FactoredTraining.TrainingParameters

model interpolation is achieved using additional scripts in the toolkit. Finally merging linear interpolation of language models into a single model is carried out using the mechanism provided in the SRILM toolkit (Stolcke, 2002). In the last phase of our experiments, we used Ken-LM (Heafield, 2011) for binarizing our language models for fast multi-threaded access during decoding. Similar to the description of the Moses toolkit in the previous section, these particular tools and settings are used for all our language modelling experiments throughout the thesis. Hence in every chapter of this thesis, we only highlight the specific modifications (if any) in language modelling settings.

2.5.2 Datasets

Since the domain adaptation experiments reported in this thesis are driven by real-life translation adaptation scenarios in Symantec, the datasets used in these experiments are also derived from Symantec documentation. In this section we briefly introduce the different datasets we use for our experiments throughout the thesis.

Symantec Datasets

The first adaptation scenario handled in Chapter 3, is one involving the translation of mixed domain datasets in the presence of domain-specific datasets. The datasets used for training the models in this set of experiments comprise Simplified Chinese–English parallel training data from two specific product domains in Symantec: 'Availability' and 'Security'. The domain 'Availability' covers data backup, recovery and the associated Symantec products (e.g. Backup Exec[9]) in the area. The data in this domain mostly comprise instructions for usage and tuning of Symantec's Backup Exec tool along with error handling and user manual instructions. The 'Security' domain data on the other hand covers system and data security as well as protection from malware vulnerability and attacks (e.g. Endpoint Protec-

[9] http://www.symantec.com/products-solutions/families/?fid=backup-exec

tion Family[10]). Here sentences are mostly obtained from the product manuals and user manuals instructing the users about the usage of data or system security products. Table 2.1 presents some examples of English sentences from each of the two domain-specific corpora.

Availability Sentences	**Security Sentences**
configuring backup exec settings and options .	*send the message to the end-user quarantine .*
backup exec provides details on each device connected to a media server and the first robotic library drive .	*to delete reporting servers from the symantec system center .*
please remove the media from the portal .	*you can use this tab to set miscellaneous antivirus and antispyware policy options .*

Table 2.1: Example sentences from Availability and Security corpora.

While the parallel data is used for translation model training, the target side of the data is used for language modelling in our experiments. The combined domain testset comprises of randomly selected sentences from each of the two domains. The specific details of the datasets in terms of number of sentences and average sentence length is described in Chapter 3 (Section 3.3.1).

The remaining Chapters in this thesis (Chapter 4, 5 and 6) are directed towards addressing the scenario of user-generated web-forum content translations for the Symantec online forums. In our experiments we focus on translating the content from the Norton Community forums[11]. These forums are intended to be a meeting place for Symantec customers, employees and enthusiasts to discuss the different products or features supported by the company. Additionally, discussions in the forum often provide a viable alternative to traditional customer service options by allowing tech savy users to self service existing or known issues. Since forums are monolingual by nature, we work around the lack of 'forum-style' parallel training data by using professionally edited documentation across different Symantec tools and product lines organised in the form of translation memories (TM). The actual content of these datasets range from user manuals, customer portal communications, internal technical documentation to software strings and marketing content. This

[10] http://www.symantec.com/products-solutions/families/?fid=endpoint-protection

[11] http://community.norton.com/

data is in-domain but ‘out-of-style’, as user-generated content prevalent in the forum data is substantially more informal and noisy compared to the professionally edited TM data. We translate the English forums to French and German, and hence use the TM datasets for both these language pairs. In addition to the parallel training data, we also have a considerable amount of monolingual English forum data (about 1.1 million sentences of actual forum content collected over a period of two years (2008-2010) from the Norton web-forums) as reference for the target domain in our experiments. We also used small amounts (about 40K sentences) of actual forum content in the target languages (German and French) for language modelling purposes. Table 2.2 shows the example of a few English sentences from the Symantec TM and Forum datasets to highlight their respective differences. The example sentences from the forum data clearly shows the noisy and informal nature of the data with spelling errors (e.g. *notron* or *sloved*), informal sentence structure (e.g. *the lights simply flicker once then nothing.*), and words fused using punctuation symbols (e.g. *guys.No* or *everything.Approaching*).

Symantec TM Sentences	**Symantec Forum Sentences**
check i have read and agree to the symantec terms of service and privacy notice .	*the laptop notron antivirus is installed on now no longer boots, the lights simply flicker once then nothing.*
helps block known phishing websites and warns against suspicious ones .	*antiphishing program has sloved the problem.*
if liveupdate informs you that no updates are available , then you have the latest update .	*Thanks guys.No luck.Tried everything.Approaching thirty hours to fix a ten minute undo.*

Table 2.2: Example sentences from Symantec TMs and Forum Data.

Selecting the development and testsets for the task of forum content translation involved randomly selecting sentences from the English forum data and manually translating them into German and French using professional translators and specific guidelines to maintain the characteristics of forums. In order to ensure that the development or testsets selected are reflective of the true characteristics of the forum content, we used the following set of features to compare the selections with the remaining forum data:

1. Average sentence count (ASC) per post
2. Standard deviation of ASC
3. Average word count (AWC) per post
4. Standard deviation of AWC
5. Average sentence length (ASL)
6. Standard Deviation of ASL
7. Type/Token Ratio
8. Stop word/Function word ratio
9. Average punctuation characters per post
10. Perplexity of selected data on a forum data language model

The specific details of these datasets in terms of sentence count and average sentence length are provided in the 'Dataset' section in each of the subsequent Chapters 4, 5 and 6.

Supplementary Datasets

In addition to the 'in-domain' (but 'out-of-style') Symantec content, we also used bilingual MT training data freely available over the web as sources of supplementary data to boost the TM-based in-domain models. In our experiments in Chapter 4, 5 and 6 we have used the following datasets to supplement the Symantec content:

1. Europarl (Koehn, 2005):[12] Parallel corpus comprising the proceedings of the European Parliament.

2. News Commentary Corpus:[13] Released as a part of the WMT 2011 Translation Task.

3. OpenOffice Corpus:[14] Parallel documentation of the Office package from OpenOffice.org, released as part of the OPUS corpus (Tiedemann, 2009).

4. KDE4 Corpus:[15] A parallel corpus of the KDE4 localisation files released as part of OPUS.

5. PHP Corpus:[16] Parallel corpus generated from multilingual PHP manuals also released as part of OPUS.

[12] http://www.statmt.org/europarl/

[13] http://www.statmt.org/wmt11/translation-task.html

[14] http://opus.lingfil.uu.se/OpenOffice.php

[15] http://opus.lingfil.uu.se/KDE4.php

[16] http://opus.lingfil.uu.se/PHP.php

6. OpenSubtitles V2 (2011) Corpus:[17] A collection of parallel movie subtitles collected from OpenSubtitles.org[18] released as part of OPUS.

7. EMEA Corpus:[19] A parallel corpus from the European Medical Agency also released as part of OPUS corpus.

All these datasets have been used to select supplementary datasets guided by the out-of-vocabulary words in the experiments reported in Chapter 4. In the following chapters 5 and 6, we have only used Europarl, Open-Subtitles and News Commentary as sources of supplementary data for data selection and model combination experiments. Like the Symantec datasets, the specific details of each of the datasets used in our experiments are presented in the 'Dataset' sections of the respective chapters.

2.5.3 Evaluation Metrics

In order to evaluate the performance of the empirical models produced by our experiments and to compare their translations to those of the baselines, the quality of translation has to be evaluated. Conducting human evaluations to judge the quality of translations is a difficult and costly task. Hence we rely mostly on automatic evaluation metrics to measure and compare the quality of translations produced by our experimental models. These metrics compare the hypothesis translations[20] to one or more sets of manually generated reference translations and provide a relative score for hypothesis translations. The basic principle guiding the process of automatic evaluation is, the closer a hypothesis sentence is to the reference translation(s), the better is its quality. Using automatic evaluation metrics not only provides a cheaper alternative to manual evaluations, but also supports fast and large scale evaluations of MT systems. Some of the most commonly used automatic evaluation metrics

[17] http://opus.lingfil.uu.se/OpenSubtitles_v2.php

[18] http://www.opensubtitles.org/

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

[20] translations produced by the MT system.

for MT 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 (Snover et al., 2006) etc. In the course of our experiments we use BLEU and METEOR to evaluate the translation quality of our experimental models since these two metrics capture two different aspects of translation quality. While BLEU uses n-gram precision to measure translation fluency and fidelity, METEOR relies on unigram precision and recall with higher emphasis on recall using linguistic resources to capture near matches[21] between hypothesis and reference translations.

BLEU

The BLEU (Bilingual Evaluation Understudy) metric estimates translation quality in terms of n-gram co-occurrence statistics i.e. the number of n-grams that occur in both hypothesis and reference translations. The actual score computed is based on a modified n-gram precision for each hypothesis translation and its corresponding references according to Equation (2.8):

$$p_n = \frac{|c_n \cap c_r|}{|c_n|} \tag{2.8}$$

where c_n and c_r are the multiset of n-grams occurring in the hypothesis and the reference translations respectively. $|c_n|$ denotes the number of n-grams in c_n and $|c_n \cap c_r|$ denotes the number of n-grams occurring in both c_n and c_r.

While the modified n-gram precision value p_n can be computed for any value of n, Papineni et al. (2002) computed a weighted average for a range of values of n[22]. However, since p_n implicitly penalises hypothesis translations that are longer (in number of words) than their corresponding reference translations, BLEU uses a brevity penalty (BP) to counter the effect by penalising hypothesis translations

[21] 'near matches' refer to synonyms or semantically close words or different surface forms of the same word.

[22] Papineni et al. (2002) report that $n = 4$ is sufficient for adequate correlation to human judgements.

shorter than their reference counterparts. BP is computed as in Equation (2.9)

$$BP = exp^{max(1-\frac{length(r)}{length(c)},0)} \tag{2.9}$$

where r and c denote the reference translation and translation option, respectively. If the hypothesis and reference translations have the same length, then BP has the value of 1.0, but the value increases for shorter hypothesis translations. Finally the BP is multiplied with the average modified n-gram precision score to compute the actual BLEU score for a testset translation as in (2.10):

$$BLEU = BP \times exp(\sum_{n=1}^{N} \frac{1}{N} logp_n) \tag{2.10}$$

Note that the value of BLEU ranges between 0 and 1 and is usually reported as a percentage value between 0% and 100% with higher BLEU score indicating better translation quality.

METEOR

Although BLEU is the most popular evaluation metric used in MT research (Koehn, 2010), there are certain limitations in its formulation (Lavie and Agarwal, 2007). BLEU is primarily based on n-gram precision but does not take into account the recall i.e. the proportion of matched n-grams out of total n-grams in the reference translation. It also does not consider near matches between the hypothesis and the reference translation and the use of geometric mean often leads to unreliable sentence-level scores (Banerjee and Lavie, 2005). METEOR (Metric for Evaluation of Translation with Explicit ORdering) is an evaluation metric that aims to address these weaknesses in BLEU. It is based on the harmonic mean of unigram precision and recall between the hypothesis and the corresponding reference translations, with the recall being given a higher weight than precision in the final computation. The algorithm starts by creating an alignment between the unigrams in the hypothesis and reference translations based on matching words at their surface level. METEOR employs stemming, synonyms and semantic closeness of words to assist the matching process. Once the alignment is computed, the precision and recall values are computed as in equation (2.11):

$$P = \tfrac{m}{w_t} \qquad R = \tfrac{m}{w_r} \tag{2.11}$$

where m refers to the number of unigrams in hypothesis translation that are also in the reference and w_t and w_r are the number of unigrams in the hypothesis and reference translations, respectively. Finally the precision P and recall R are combined using a harmonic mean with the recall being weighted 9 times higher than the precision as in (2.12):

$$F_{mean} = \frac{10PR}{R + 9P} \tag{2.12}$$

The F_{mean} measure is only based on unigram precision and recall. In order to handle longer n-gram matches a penalty p is computed such that the more non-adjacent mappings between hypothesis and reference exist, the higher is the value of the penalty. This penalty is computed by grouping the unigrams into the least possible number of chunks where a chunk is defined as the set of adjacent unigrams in both translations. Hence a longer size of individual chunks automatically means a smaller number of chunks. The penalty is computed according to the formula in (2.13):

$$p = 0.5\left(\frac{c}{u_m}\right)^3 \tag{2.13}$$

where c is the number of chunks and u_m is the number of unigrams that have been mapped in the alignment. Finally the METEOR score M is computed by combining the penalty with the F_{mean} as in (2.14):

$$M = F_{mean}(1 - p) \tag{2.14}$$

When multiple references are used, the algorithm computes the scores for the hypothesis against each of the references separately and selects the highest scores. Note that METEOR scores, like BLEU, are also between 0 and 1 and are reported in this thesis as a percentage score between 0% and 100%.

2.6 Summary

In this chapter we have briefly presented the evolution of different MT techniques culminating in the advent of SMT as the dominating paradigm in MT research today. We have presented a brief description of the PBSMT models and its components and discuss their training process. Establishing the theoretical and empirical foundations of the PBSMT paradigm, we focussed on the specific problem of domain dependence and adaptation in ML and NLP which forms the central theme of the research presented in this thesis. We have presented the problem of domain dependence and briefly introduced the standard domain adaptation approaches presented in the literature for general NLP tasks. Eventually we have presented a detailed account of relevant related work on domain adaptation focussed in the area of SMT. In the context of this previous research on domain adaptation of SMT models, we have discussed the relevance of the research questions we address in this thesis. Presenting the scenarios which motivate our research questions, we have shown how our research addresses gaps in the existing domain adaptation research literature.

Finally we have presented an overview of the tools and techniques, including the Moses SMT training toolkit used for our experimentation throughout the thesis. Furthermore a brief introduction of the datasets used followed by a discussion on the automatic evaluation metrics used for translation quality evaluation concludes the chapter. Motivated by the relevance of our research questions in domain adaptation research, using the resources and techniques outlined in this chapter, we aim to provide conclusive answers to all four of the research questions through the research and experiments presented in Chapters 3 to 6.

Chapter 3

Combining Domain-specific SMT Systems using Automatic Classifiers

Statistical Machine Translation (SMT) systems tend to improve translation quality with increasing amounts of training data. The underlying models in an SMT system rely heavily on statistics computed from sentence-aligned parallel training data. Greater amounts of training data lead, therefore, to richer statistics and better models. However, the conventional wisdom of more data being better does not always hold true for domain-specific models and translations (Ozdowska and Way, 2009), particularly in a scenario where the data to be translated come from a mixture of different domains. In such a scenario, creating a generic translation system from the combined data of all available domains might not always provide the best results (Banerjee et al., 2010). Domain-specific translation often performs best with systems trained only on in-domain data (Haque et al., 2009). Large training corpora, which are traditionally used to train SMT systems, usually comprise a heterogeneous collection of homogeneous sub-parts. Considering the high degree of homogeneity within such sub-parts, each of them could be treated as a domain and be capable of translating domain-specific data from the same domain. Therefore, in contrast to building generic systems trained on large quantities of heterogeneous data, the embedded homogeneity within sub-parts of training data could be leveraged to create a combination of domain-specific systems.

This chapter describes our adaptation approach aimed at addressing such a scenario based on multi-domain corporate content from Symantec. An SMT system trained on domain-specific data is best suited to translate text from the same domain, but the translation quality suffers for out-of-domain texts (Haque et al., 2009). Hence, in the presence of domain-specific data or pre-trained SMT systems for different domains, we aim to figure out an effective way of combining such systems to improve the translation quality for mixed domain texts. Such a technique is anticipated to address our research question RQ1 which focuses on the effectiveness of combining translations from domain-specific SMT systems to achieve better translation quality for combined-domain texts. To accomplish this, we propose a method of combining translations from multiple domain-specific SMT systems using automatic classifiers. Our experiments reveal that this approach not only outperforms the more traditional approach of training a single SMT system on concatenated data (from multiple domains), but also performs better than the state-of-the-art method of combining multiple models using the multiple decoding path functionality of the Moses decoder (Koehn and Schroeder, 2007).

Since our approach is based on combining translations from multiple domain-specific systems, we further compare our approach to a state-of-the-art system combination technique (Du et al., 2009, 2010). For the system-combination experiment, we use a minimum Bayes risk (MBR) (Kumar and Byrne, 2004) decoder to select the best hypothesis as an alignment reference for a confusion network (CN) (Mangu et al., 2000). The final translation is generated by searching for the best translation on the CN built using the TER metric (Snover et al., 2006). For fair comparison we use the same systems used in our classifier-based approach in the system combination experiments. Experimental results show our classifier-based combination technique to be more effective than the system-combination approach for the current task.

The rest of the chapter is organised as follows. In Section 3.1, we present the motivation for this work followed by a detailed description of the approach in Section 3.2. In Section 3.3 we present an account of the datasets used for experimentation followed by details of the classifier and the different experimental setups. Section 3.4 comprises experimental results in terms of automatic evaluation metrics followed by manual evaluation experiments in Section 3.5. Finally we conclude the chapter with our observations in Section 3.6 followed by contributions and summary.

3.1 Motivation

The domain adaptation techniques presented in this chapter are aimed at achieving high-quality translations for mixed-domain texts in a scenario where sufficient training data is available for each of the constituent domains. Let us consider a case where a user wants to automatically translate the Microsoft Office Help files, presented when (s)he hits the F1 button while using any of the individual products in MS Office (Word, Excel, Powerpoint, Outlook etc.). In such a case, the user might at one instance want to translate Help files from Microsoft Excel, while in the next moment he might want to translate data from Microsoft Outlook. In this example, the data to be translated come from a mixture of different domains, if we consider each product line to be a domain. In order to translate such a dataset, existing domain-specific datasets on each of the product lines could easily be concatenated to produce one generic training dataset which is used to train a single translation model aimed at the mixed-domain data. However, SMT systems are known to perform best when the data to be translated come from the same domain as the training data. Therefore, in contrast to a generic model, utilising the best translations from individual domain-specific SMT models should ideally provide a better translation in the given scenario. Investigating the feasibility of this hypothesis forms the primary motivation of the experiments reported in this chapter. In addition to improving translation quality, such a combined approach provides a flexible and scalable framework for addition or removal of new domains. Furthermore, as domain-specific models are much smaller than their generic counterparts, this also provides improved ease of maintenance.

Such a scenario is particularly relevant to the localisation or translation service industry. Language service providers (LSPs) need to reuse existing MT systems and tune them to new requirements, or combine multiple domain-specific translation systems to widen system coverage to suit customised translation requests (Vashee and Gibbs, 2010). The industrial relevance of the problem scenario, combined with the necessity of better translation quality and more flexibility forms the primary motivation for this domain adaptation technique.

3.2 Approach

Since domain-specific sentences are best translated by in-domain SMT systems, our approach uses an automatic classifier to identify the domain of an input sentence prior to translation. The primary objective of the approach is to allow the sentences from a mixed-domain dataset to be translated by the appropriate in-domain SMT system. Depending upon the label assigned by a classifier, a sentence is routed to the appropriate domain-specific system for translation. Hence for a mixed-domain dataset, the correctly classified sentences (whose actual domain and the one predicted by the classifier are the same) are translated by an 'in-domain' system. The wrongly classified (actual domain do not match predicted domain) ones are effectively routed to an 'out-of-domain' system for translation.

Obviously, the accuracy of a classifier plays an important role in the success of our approach. Depending upon the implementation technique of the classifier, it might be unable to assign a proper label to some of the sentences in the dataset. Usually shorter sentences, or sentences without any discriminative features (with respect to the features computed on the training data) fall into this category (Sriram, 2010).

In our approach these sentences are routed to a system that is trained on a combination of different domain-specific systems. This technique ensures that the translation quality is at least as good as that provided by the combined domain-specific system. Generally however, only a few sentences are routed into the combined-domain SMT system and majority of the sentences are routed to the appropriate in-domain systems. The block diagram in Figure 3.1 explains the approach of combining multiple domain-specific and a combined-domain system using an automatic sentence classifier.

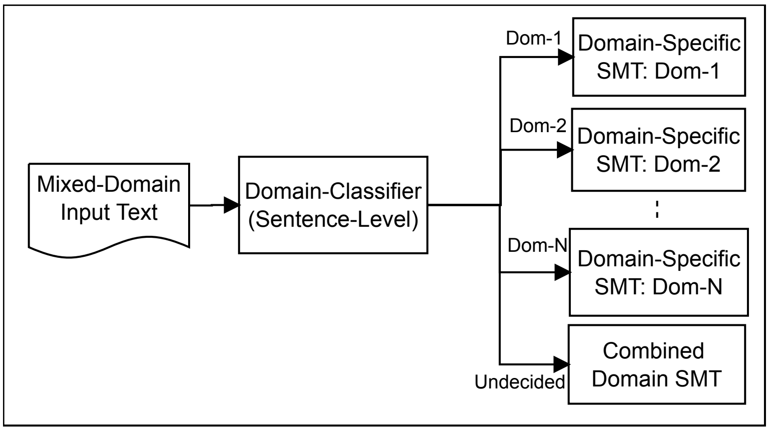

Mixed-Domain Input Text
Domain-Classifier (Sentence-Level)
Dom-1
Domain-Specific SMT: Dom-1
Dom-2
Domain-Specific SMT: Dom-2
Dom-N
Domain-Specific SMT: Dom-N
Undecided
Combined Domain SMT

Figure 3.1: Domain-specific SMT systems combined using a classifier

Although the concept of classifying input sentences to identify domains has been tried by Xu et al. (2007) as well as Yamamoto and Sumita (2008), our work differs significantly, both in terms of the scenario and approach. While Xu et al. (2007) use domain-specific language models and a generic translation model to translate domain-specific sentences, we combine the translations from two completely independent systems comprising domain-specific language and translation models. Yamamoto and Sumita (2008) create synthetic domains by clustering the training data and treat each such cluster as separate domains. Our work differs from theirs in that we use naturally defined domains (defined by Symantec product lines) in the data. Furthermore, our approach to domain-classification of input data and techniques to handle failures in domain classification is quite different from what is reported in Xu et al. (2007) as well as Yamamoto and Sumita (2008). While Xu et al. (2007) use a language model-based and an information retrieval-based classification approach, Yamamoto and Sumita (2008) select the cluster which maximises the likelihood of a source sentence as its domain. In contrast, we use a support-vector machine-based classifier which is known to be more robust for the task of text classification (Dumais et al., 1998). Using a combined-domain model to handle the failures in text classification further discriminates our approach from the ones present in the literature.

3.3 Experimental Setup

This section presents the data resources used for experimentation followed by the implementation details of the classifier. The section concludes with a detailed description of the different experimental setups we use to compare our approach to more traditional as well as state-of-the-art approaches.

3.3.1 Datasets

To effectively emulate the scenario for our experimental setup, we use real-world corporate content available in the form of translation memories (TMs) from Symantec in two distinct domains: 'Availability' and 'Security'. The sentences in both domains are obtained from the documentation of the two product lines. The domain 'Availability' (Ava) covers data protection, reliability and recovery and associated Symantec products in the area. The sentences in this domain mostly comprise instructions for usage and tuning of Symantec's data availability tools. The 'Security' (Sec) domain data on the other hand covers system and data security as well as protection from vulnerability and attacks. Here sentences are primarily obtained from the product manuals instructing users about the usage of data or system security products. Our experiments are conducted for the –English(En)–Simplified Chinese(Zh) language pair in both directions.[1]

Table 3.1 reports the number of sentences along with the average sentence length

[1] The Chinese translations were produced by human translators from the English source.

Dataset	Availability			Security			Combined		
	Sent. Count	En ASL	Zh ASL	Sent. Count	En ASL	Zh ASL	Sent. Count	En ASL	Zh ASL
Train	130,162	13.59	13.74	95,067	13.29	13.59	225,229	13.46	13.68
Devset-1	500	28.95	28.97	500	27.07	27.71	1,000	28.01	28.34
Devset-2	500	13.55	13.52	500	11.88	11.87	1,000	12.72	12.69
Testset-1	1,000	28.74	28.62	1,000	27.47	28.29	2,000	28.10	28.45
Testset-2	1,000	14.47	14.62	1,000	14.01	14.27	2,000	14.24	14.45

Table 3.1: Size and Average Sentence Length (ASL) of Training, Development and Test Corpus for 'Availability', 'Security' and 'Combined' datasets

(ASL) for all the datasets used in our experiments. Since the Zh datasets originally have no word-boundaries in them, we segment the Zh sentences into words using the Stanford Chinese Word Segmenter (Tseng et al., 2005) in the pre-processing steps in order to compute the ASL. Each domain-specific dataset is split into training, development sets (devsets) (500 sentences) and testsets (1000 sentences) for training, tuning and testing, respectively. The Combined train, testsets and devsets are created by combining the sentences from the individual domain-specific testsets in a random order. All the training, dev and testsets are subjected to tokenisation and lowercasing using the scripts associated with the Moses SMT system.

Since our approach comprises the combination of SMT systems and a classifier, we use two different dev and testsets (Set-1 and Set-2) to tune and evaluate the systems. Notably, the ASLs for the test and devsets in Set-1 are nearly double that of the training sets. We deliberately chose longer sentences (with an ASL of 27–30 words) for devsets and testsets, since longer sentences are supposed to be harder to translate for the standard phrase-based SMT systems (Shimohata et al., 2004). Longer sentences are harder to translate when compared to shorter sentences due to multiple reasons. Firstly, longer sentences tend to have larger hypothesis search spaces which makes it difficult for the decoder to reach good translations (cf. Chapter 2, Section 2.3). Secondly, translating longer sentences is often computationally more expensive than translating shorter sentences. Finally longer sentences often tend to contain complex syntax and long-range dependency structures, which are difficult for phrase-based SMT models to capture (Bach, 2012). Hence this set is particularly targeted towards testing the performance of the SMT systems involved in our experiments.

While longer sentences are harder to translate, it is easier for the classifiers to handle them, as longer sentences ensure richer features (Sriram, 2010). Accordingly, we designed the second set of dev and test data (Set-2) to have an A.S.L closer to that of the training data in order to estimate the performance of the classifier and emulate a scenario where the classifier works at a lower accuracy. The test and devsets for Set-2 were randomly chosen from the available data for both domains. All the experimental configurations reported in Section 3.3.3 are subsequently evaluated with both testsets in our experiments.

3.3.2 Domain Classification

As previously stated, the automatic sentence classifier used to identify the domain of a test sentence is one of the most vital components in our approach. Since the premise of our experiments is to utilise domain-specific translation models to translate in-domain sentences, the domain assigned to a testset sentence by the classifier helps us route the sentence in question to the appropriate domain-specific SMT model. In the case of misclassification by the classifier (a sentence originally from the 'Ava' domain is classified as 'Sec' domain), the sentence would be routed to the incorrect domain-specific model, e.g. a 'Ava' sentence is routed to and hence translated by the 'Sec'-based translation model. Therefore, the accuracy of the classifier deeply affects the final translation score of the sentences from the combined testset. As already pointed out in Chapter 2.4.4, Support Vector Machines (SVM) (Joachims, 1999) with linear kernels report the best performance for text categorisation tasks,[2] thereby prompting us to choose this technology for classification. In this section we give a brief account of the design of the sentence classifier, the features and the values used and the subsequent classification results with respect

[2] We experimented with the polynomial and radial-basis function kernels for the task at hand, but the best results were obtained for linear kernels.

to our experiments.

SVM Classifier

We use an SVM-based classifier with a linear kernel for domain-wise classification of the testset sentences. The classifier was initially trained on the source side of the training data for each domain. The features used to train the classifier included the word bigrams occurring in the training datasets. This choice of the features are guided by the nature of the differences between the two domains under consideration. Both datasets (Ava and Sec) selected from Symantec corporate documentation are different primarily in terms of content words. Hence, word-based features were considered for the task.[3] For the feature values, we utilised the widely used 'Term Frequency-Inverse Document Frequency' (tf-idf) values of the word bigrams appearing in the training corpus (Salton and Buckley, 1997). However, in our case the classification is required at the sentence level instead of the document level. Hence we slightly modified the tf-idf to generate 'Term Frequency Inverse *Sentence* Frequency' (tf-isf) as the values of the features. For implementation we used the SVM Light[4] application which is an open source implementation of the SVM.

Feature Extraction

The very first step in document categorisation is the transformation of documents into a representation suitable for the SVM learning algorithm and classification task. As the requirement is domain-based categorisation of sentences, the content words in the sentence constitute the most appropriate features. In order to capture limited context in the features, we use word bigrams as the feature sets for our classification task. The feature values are the frequencies of the word bigrams in the corpus scaled by their inverse sentence frequency. Tf-isf is known to work well in the scenario (Banerjee et al., 2010) owing to its closeness to the tf-idf (Salton and

[3] We also experimented with unigram word features, but the results failed to outperform the bigram features.

[4] http://svmlight.joachims.org/

Buckley, 1997) features.

The training set sentences are pre-processed to extract the feature vectors. Pre-processing includes lowercasing and stop-word removal. However, since we use bigrams as features, a bigram is considered to be a stop word only when both its constituents belong to the stop word list. In order to compute the tf-isf feature vectors for every distinct word bigram in the corpus, formula 3.1 is used:

$$\begin{aligned} \text{feature-value}(b_i) &= \text{tf}(b_i) \times \text{isf}(b_i) \\ \text{tf}(b_i) &= \tfrac{n_i}{\sum_k n_k} \\ \text{isf}(b_i) &= log \tfrac{N}{|\{s : b_i \epsilon s\}|} \end{aligned} \tag{3.1}$$

Here, tf(b_i) denotes the term-frequency of the bigram b_i. n_i is the count of the bigram in the sentence, k being all such bigrams in the same sentence, while N denotes the total number of sentences. isf(b_i) is the Inverse Sentence Frequency for the bigram, and s denotes the set of all sentences containing the bigram. The English side of the training data (as reported in Table 3.1) was used to train the classifier. The feature set comprising unique word bigrams from the training data contains 141,884 entries.

Classification Results

Once the classification model is trained, it is used to classify sentences from both domain-specific as well as combined domain testsets. The test data also undergoes the same pre-processing as the training data and is eventually converted into a vector representation in terms of the training model feature vectors. However, some test sentences might not have any features in common with the training feature set and so would be converted to a null feature vector. Such sentences cannot be handled by the SVM classifier and need to be dealt with separately. In order to evaluate the performance of the classifier (taking into account the phenomenon of the failed classification due to the null feature) we use the widely used metrics of *precision* and *recall* defined in equation (3.2):

$$\begin{aligned} Precision &= \tfrac{t_p}{t_p+f_p} \\ Recall &= \tfrac{t_p}{N} \end{aligned} \tag{3.2}$$

where t_p is the number of sentences which are correctly classified, f_p is the number of sentences which are wrongly classified and N is the total number of sentences used in the classification task. Thus *precision* defines the percentage of correct classifications amongst those sentences which could be handled by the classifier, while *recall* takes into account the sentences which could not be handled by the classifier due to null features. The results of the classification tasks for the three different domains (Ava, Sec and combined) for both testsets (Testset-1 and Testset-2) along with the number of failed sentences are reported in Table 3.2.

Testset	**Domain**	**Precision**	**Recall**	**Undecided**
Testset-1	Availability	95.67	94.90	8
	Security	96.33	94.50	19
	Combined	95.99	94.70	27
Testset-2	Availability	91.78	80.40	124
	Security	95.76	76.70	199
	Combined	93.68	78.50	323

Table 3.2: SVM classifier accuracy for domain-specific and combined testset data

The 'Precision' and 'Recall' columns denote the respective metrics while the 'Undecided' column indicates the number of sentences which have null feature vectors and hence could not be handled by the classifier. Evidently, the number of 'Undecided' sentences is much greater for Testset-2 compared to Testset-1 as the latter has much longer sentences leading to lower chances of generating null feature vectors. For this reason, while the precision values for both testsets are roughly the same, the recall values drop drastically for Testset-2.

3.3.3 Experiments

In order to effectively compare our approach to the traditional approach of concatenating multi-domain data to train a single generic model as well as to more state-of-the-art approaches of using multiple domain-specific models within the Moses decoder and pure system combination methods, we use the following five experimental setups:

1. SDS: Simple domain-specific models.
2. CD: Concatenated data models.
3. Multiple Decoding Path-based Combination (MDPC): Domain-specific models combined using multiple decoding paths of Moses decoder.
4. Classifier-based Domain-specific SMT combination (CDS): Our approach of combining domain-specific SMT systems using an automatic classifier.
5. SYSCMB: A CN-based system combination technique to combine translations from domain-specific SMT systems.

SDS: Simple Domain-specific Models

In the first stage of our experiments we start by training simple domain-specific SMT models on both domains. As shown in Figure 3.2, we train two independent SMT models on the 'Ava' and 'Sec' training data. After training, these models are MERT-tuned using the domain-specific development sets. We evaluate these models using both domain-specific as well as combined domain testsets. Cross-domain evaluation is also done by exposing the models trained in one domain to the testsets of the other domain which allows us to observe how well the out-of-domain testsets are translated. These models are henceforth referred to as the 'Simple Domain Specific' (SDS) models.

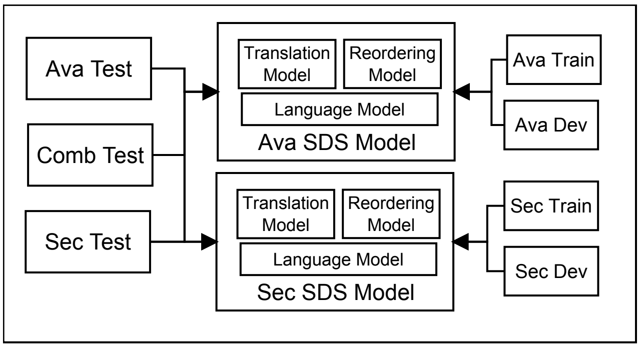


Figure 3.2: Simple Domain-Specific Model

Concatenated Data Models

In the second phase, training data from both domains are concatenated and used to train a single generic model, as shown in Figure 3.3. This model is then subjected to domain-specific as well as combined-domain data tuning. As in the previous phase, testing is carried out using both domain-specific as well as combined domain training data. The results for this phase not only provide insight into the system performance for the generic model trained on the concatenated data, but also give an idea about how increasing the training set with data from the other domain might affect the translation scores. Since this model is based on the traditional approach of handling mixed-domain datasets, the translation results at this stage were considered to be the baseline results. We call this model the 'Concatenated Data' (CD) model in the rest of the chapter.

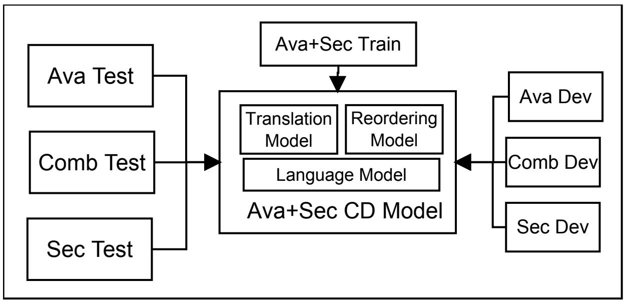


Figure 3.3: Concatenated Data Model

Domain-specific Models Combined Using Multiple Decoding Paths

In contrast to the more traditional approach described in the previous section, the Moses decoder provides a more state-of-the-art technique of combining multiple translation and reordering models in a single SMT system. This particular feature utilises the decoder's ability to use multiple decoding paths (Koehn and Schroeder, 2007). Moses supports two different configurations for this feature:

- **Both**: In this configuration, all constituent models are used to score a particular translation option. Since all of the models are used to score a single phrase, this configuration requires all the constituent phrase-tables to have common phrase pairs. If a phrase-pair occurs only in a single phrase table and not in others, it is ignored.

- **Either**: This configuration allows a translation option to be scored by any one of the constituent models. For a particular source phrase, translation options are collected from one table, and additional options are collected from the other tables. If the same translation option (in terms of identical input phrase and output phrase) is found in multiple tables, separate translation options are created for each occurrence with different scores.

In our case, since we want to combine two different domain-specific models, we use the 'either' configuration to ensure model combination using this technique. The individual domain-specific SDS models trained in the first phase of experiments are used as the constituent models. The combined models are tuned using both domain-specific and combined-domain development sets. Evaluation of these models against domain-specific as well as combined testsets reveals the effect of model combination using Multiple Decoding Paths. Comparing the results of this phase with those of the CD model gives us an idea of the effectiveness of combining pre-trained models in the context of the current domain adaptation task. This model, as depicted in Fig 3.4 is referred to as 'Multiple Decoding Path Combined' (MDPC) models in the rest of the chapter.

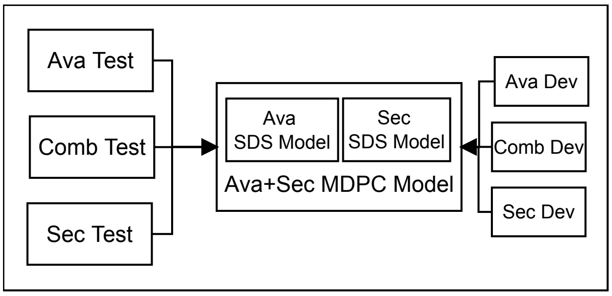


Figure 3.4: Domain-specific Models combined using Multiple Decoding Paths of Moses

Domain-specific Systems Combined Using an Automatic Classifier

The third phase of our experiments utilises our approach, where we combine the two SDS models (one for each domain) and a combined MDPC model using the automatic sentence classifier (Section 3.3.2). The classifier labels the input sentences with either 'Ava', 'Sec' or 'undecided' labels. Depending upon the label, the sentence is routed to the appropriate SDS model (for 'Ava' or 'Sec') or to the MDPC model (if'undecided') in order to utilise the best of the domain-specific models as shown in Figure 3.5. We prefer the MDPC model over the CD model for translating the undecided sentences since it allows us to maintain separate domain-specific models. This is driven by the objective of developing a flexible architecture for easy incorporation of new domain-specific models. We use this model to test both domain-specific as well as combined-domain testset data to observe its effect on translation quality.

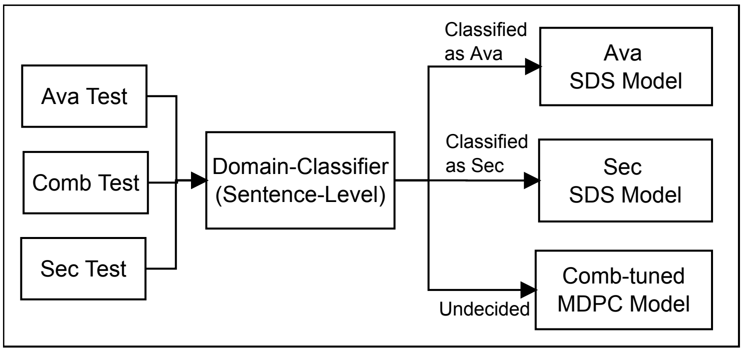


Figure 3.5: Domain-Specific Models combined using an SVM-Based sentence classifier

Automatically classifying a domain-specific testset results in labelling a part of the testset as out-of-domain. This basically means that the sentences which originally belonged to one domain are assumed by the classifier to be statistically closer to the training data of the other domain (according to the classification features). When repeating the experiments with combined domain test data, since the classifier routes the majority of the in-domain sentences to the appropriate domain-specific SDS models, the quality of translation is assumed to improve.

System Combination of Domain-specific SMT systems

The CDS approach described in the previous section aims to capitalise on combining 'good' translations from in-domain SDS systems for each input sentence in the mixed-domain data. Hence the translations of a mixed-domain input comprise combination of translations provided by individual SDS systems. While the CDS approach combines translations from multiple systems at the sentence level, the possibility of using combinations at a more granular level (i.e. phrase level or word level) are covered by system combination techniques (Rosti et al., 2007). System combination (SYSCMB) methods are widely used to combine translations from individual MT systems (usually from different paradigms) in order to improve the translation quality over those provided by each of the constituent systems (Callison-Burch et al., 2010). The objective of improving translation quality by combining translations from multiple systems is thus shared by both SYSCMB and CDS methods. This motivates us to use a SYSCMB technique and compare its effect on translation quality with that of the CDS approach in the current setting of mixed-domain data translation.

For the system combination experiments we use an in-house implementation of the word-level combination scheme (Rosti et al., 2007) to combine multiple translation hypotheses. This implementation is based on the MBR-CN (Du et al., 2009, 2010) framework as depicted in Figure 3.6. The first step in the process involves combining 1-best hypothesis from each of the individual systems to form a new N-best list. Since each entry in this N-best list comes from a different MT system, they can have varying word orders. Hence, it is essential to define a *backbone* which

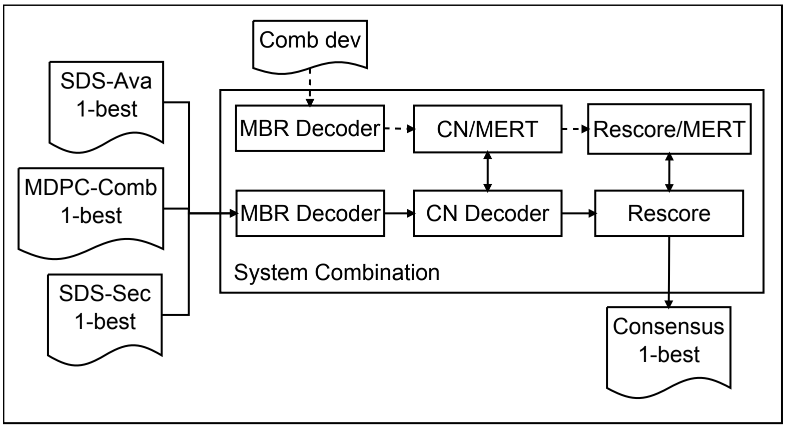


Figure 3.6: System Combination Method

determines the general word order of the CN. This is achieved by using an MBR decoder to select the best single system output E_r from the N-best list by minimizing BLEU (Papineni et al., 2002) loss as per Equation (3.3).

$$r = arg\, min_i \sum_{j=1}^{N_s}(1 - BLEU(E_j, E_i)) \tag{3.3}$$

where N_s indicates the number of translations in the merged N-best list and E_i are the translations. To fairly compare the effect of SYSCMB method to that of the CDS approach, we use the output of 3 systems (SDS-Ava, SDS-Sec and MDPC) within the system combination framework.

Once the MBR identifies the *backbone*, all other hypotheses are aligned to it using the TER metric (Snover et al., 2006) to construct the CN. The alignment process allows NULL words. Each arc in the CN represents an alternative word at that position in the sentence. During the process of constructing the CN, the number of votes for each word is also computed. The following features are used in the process:

- word posterior probability
- 4, 5-gram target language model
- word length penalty

- Null word length penalty

The weights of the CN are tuned using MERT (Och, 2003) on an held out devset.

Finally a rescoring module is used to process the N-best list generated by the combination process. The rescoring module uses a set of global features[5] to select the best hypothesis from the N-best list. The feature weights are again optimised using the MERT algorithm on the devset. Once the features weights are set by MERT, the testsets are decoded using a CN decoder to generate new N-best lists and the rescorer is used to find the 1-best consensus translations. As in the case of the other models, we use the SYSCMB method to translate both domain-specific as well as combined-domain testsets in our experiments.

3.4 Results and Analysis

In this section we present the evaluation results in terms of the automatic evaluation metrics for the different phases of experiments conducted. Separate results are reported for both the sets of test/devsets (Set-1 and 2) used in our experiments along with a subsequent analysis of the results.

3.4.1 SDS Model Results

Table 3.3 reports the results for our first phase of the experiments involving simple domain-specific models evaluated on in-domain, out-of-domain and combined-domain testsets. The 'Trn' column indicates the training data on which the models are trained, while the 'Test' column indicates the nature of the test data. All the models in this phase of experiments have been MERT-tuned using only in-domain devsets. The scores for both Testset-1 and Testset-2 in Table 3.3 show that a domain-specific model performs much better in translating in-domain sentences compared to out-of-domain ones, thereby confirming our initial assumption as well as the basic premise on which our approach is based.

[5] For the full set of features refer to Du et al. (2010)

	Test	**Train**	**Testset-1**		**Testset-2**	
			BLEU	METEOR	BLEU	METEOR
Zh–En	Ava	Ava	**57.42**	**45.48**	**68.65**	**50.59**
		Sec	25.96	33.34	25.18	33.24
	Sec	Ava	25.76	33.48	29.93	34.09
		Sec	**57.03**	**45.44**	**65.98**	**49.00**
	Comb	Ava	41.86	39.00	**50.90**	**41.88**
		Sec	**41.88**	**39.02**	46.59	40.26
En–Zh	Ava	Ava	**52.58**	**69.10**	**66.83**	**78.43**
		Sec	24.16	44.94	23.21	42.81
	Sec	Ava	21.95	43.99	28.85	47.88
		Sec	**53.60**	**69.44**	**64.32**	**77.12**
	Comb	Ava	**39.05**	**57.39**	**49.00**	**63.99**
		Sec	38.31	57.00	44.74	60.42

Table 3.3: Automatic Evaluation Scores for SDS models. Best scores for each testset are in bold

The best translation scores are obtained for the models where the training data and test data are from the same domain. For Zh–En translations, the in-domain translation scores are on an average 31.37 and 39.99 absolute BLEU points better than the corresponding out-of-domain testset scores for Testset-1 and Testset-2, respectively.[6] We observe similar trends for the En–Zh language direction. All these improvements are statistically significant at the p=0.05 level using bootstrap resampling (Koehn, 2004). The METEOR scores show a similar trend of improvements with in-domain scores being on average 12.05 and 16.13 absolute points better than the out-of-domain translation scores for Zh–En translations for Testset-1 and Testset2, respectively. While the trend of improvements in METEOR are similar to that of BLEU, the range of improvements in METEOR are much smaller, due to METEOR's ability to take near matches in translation into consideration.

Table 3.3 also reports the results of translating combined-domain testsets using the SDS models. Comparing the Zh–En results of the combined testset with those of the domain-specific ones it can be observed that the scores are on average 15.36 and 18.57 absolute BLEU points lower than in-domain test scores for Testset-1 and -2, respectively. Again the same scores are on average 16.01 and 21.19 absolute

[6] The average improvement is computed by taking an average of individual improvements over Ava and Sec testsets.

BLEU points better than the corresponding out-of-domain testsets for Testset-1 and -2, respectively. Similar effects are observed for the combined-domain translation scores considering En–Zh translations on both Testset-1 and Testset-2. All these improvements and deteriorations of the combined-domain testsets over out-of-domain and in-domain translation scores, respectively, are statistically significant at the p=0.05 level. The METEOR scores despite having the same trend, the range of improvements or degradation is much lower compared to BLEU, which can again be attributed to the near match factor and also a higher relative weighting on recall.

Since the combined testset contains an equal number of in-domain and out-of-domain sentences, the high quality of translation of the in-domain sentences is offset by the poor-quality translation of the out-of-domain sentences, thereby providing a translation score which is nearly the average of the in-domain and out-of-domain translation scores. Therefore, overall, these results strongly suggest the need for domain adaptation to obtain better quality translation of combined-domain sentences.

3.4.2 CD Model Results

Table 3.4 reports the results in the second phase of our experiments using CD models (Section 3.3.3). The 'dev' column in the table reports the domain of the devset used to tune the model, indicating that we used domain-specific MERT tuning on the models to observe its effect. As in the case of SDS models, we test the CD models using domain-specific as well as combined-domain testsets. The table reports results for both testsets used in our experiments. Best scores for each testset are in bold. $*$ denotes statistically significant improvement of BLEU scores for Comb testset over Comb scores in Table 3.3. † indicates statistical significant drop in BLEU scores on in-domain testsets w.r.t. scores in Table 3.3.

The results in Table 3.4 show an average improvement of 13.73 and 16.52 absolute BLEU points (5.85 and 7.75 absolute METEOR points) for the combined testset translations on Testset-1 and -2, respectively, over the scores reported with SDS models in Table 3.3 (rows 5, 6), for Zh–En translations. For En–Zh trans-

	Test	Dev	Testset-1		Testset-2	
			BLEU	METEOR	BLEU	METEOR
Zh–En	Ava	Ava	†**55.57**	44.85	†**66.53**	49.23
		Sec	†55.41	**44.93**	†65.63	**49.34**
		Comb	†55.46	44.83	†66.29	49.29
	Sec	Ava	†55.67	44.84	†64.05	**48.41**
		Sec	†**55.67**	**44.93**	†**64.55**	48.31
		Comb	†55.59	44.76	†64.47	48.31
	Comb	Ava	∗**55.62**	44.85	∗64.85	**48.88**
		Sec	∗55.57	44.93	∗65.39	48.81
		Comb	∗55.61	44.79	∗**65.56**	48.77
En–Zh	Ava	Ava	†**51.55**	68.44	†**64.72**	**77.18**
		Sec	†50.97	**68.51**	†64.27	76.71
		Comb	†51.37	68.36	†64.52	77.10
	Sec	Ava	†51.90	68.60	†63.52	76.64
		Sec	†52.29	68.79	†**63.94**	**77.13**
		Comb	†**52.37**	**68.91**	†63.83	77.00
	Comb	Ava	∗51.72	68.52	∗64.24	**77.12**
		Sec	∗**52.67**	**69.24**	∗63.9	76.68
		Comb	∗51.98	68.63	∗**64.28**	77.09

Table 3.4: Automatic Evaluation Scores for CD models.

lations, similar improvements are observed on both testsets over the SDS model scores (Table 3.3, rows 11 and 12). The METEOR scores show a similar improvement trend for the combined testsets in both language directions and testsets with a different range of improvements. All these improvements (cells marked by ∗ in the table) are statistically significant at the p=0.05 level. Interestingly, combined-domain or domain-specific tuning does not significantly affect the scores for the model. Compared to the results in Table 3.3, the improvements are simply due to the model being trained on data from both domains. Since data from both domains have been combined to train a single model and tuning only sets the global weights for the model components, domain-specific tuning is not able to bias the model towards one particular domain. This lack of biasing results in more-or-less uniform translation scores irrespective of the domain of tuning.

However, comparing in-domain translation scores (where training and test data are from the same domain) in Table 3.3 to domain-tuned scores (where test and tuning data are from same domain) in Table 3.4, we see an average drop of 1.61 and 1.78 absolute BLEU points in Zh–En translations for Testset-1 and -2, respectively.

For En–Zh translations, similar drops are observed. Again, the METEOR scores follow a similar trend with in-domain translation scores being lower than those reported in Table 3.3. These statistically significant drops (cells annotated by † in the table) in translation scores can be attributed to a higher degree of generalisation present in the CD model, which is introduced due to training on combined data from both domains. Furthermore, the results clearly indicate that while increasing training data by adding out-of-domain corpora is useful for combined-domain translations, domain-specific translations suffer slightly due to the increased generality of the system.

In spite of the significant improvement in translation scores for the combined domain data, one major disadvantage of this approach is its lack of maintainability. Every time data for a new domain becomes available, we need to retrain the entire system. This also complicates the issue of adding or removing existing domain-specific data from the mix. Therefore from the point of flexibility and maintenance the CD configuration is definitely not preferable. Moreover, domain-specific testset translation scores suffered due to increased generality of the models. As data from more and more domains are added to the system, it further increases the generality in the model leading to poorer scores for domain-specific test data, irrespective of domain-specific tuning, as is evident from the results. Hence the quality of translation might suffer when scaling this approach to more than two domains. These major disadvantages of the CD approach thus motivate us to search for a more flexible and scalable solution to mixed-domain data translation in the current setting.

3.4.3 MDPC Model Results

The third phase of our experiments involves the usage of multiple decoding path feature of the Moses decoder to combine multiple domain-specific models into a single SMT system.[7] Similar to the CD models (Section 3.3) the MDPC models

[7] We also ran a few experiments with the recently developed 'Back-off' models (http://www.statmt.org/moses/?n=Moses.AdvancedFeatures#ntoc18), where a second phrase table is used to handle any phrase pair which is not found in the first table. However, the overall

are also subjected to domain-specific MERT tuning and evaluation in terms of both domain-specific as well as combined-domain testsets. Table 3.5 presents the translation results for both our testsets in this phase of experiments. The best scores for each testset are in bold. § indicate statistically significant drop in BLEU scores for Comb testsets compared to corresponding scores for the CD model in Table 3.4

	Test	**Dev**	**Testset-1**		**Testset-2**	
			BLEU	METEOR	BLEU	METEOR
Zh–En	Ava	Ava	**55.34**	**44.90**	**67.56**	**49.57**
		Sec	32.74	36.36	23.25	30.21
		Comb	51.37	42.60	62.68	47.14
	Sec	Ava	35.27	37.31	29.66	34.56
		Sec	**55.71**	**44.94**	**65.00**	**48.40**
		Comb	52.98	43.41	62.73	47.61
	Comb	Ava	§46.81	41.43	§52.12	42.58
		Sec	§45.17	40.66	§47.48	39.32
		Comb	§**51.37**	**42.60**	§**62.72**	**47.37**
En–Zh	Ava	Ava	**51.82**	**68.65**	**64.94**	**77.73**
		Sec	20.60	43.95	32.45	51.72
		Comb	47.19	65.08	58.65	71.58
	Sec	Ava	32.81	54.20	27.38	48.15
		Sec	**52.34**	**68.93**	**64.16**	**77.43**
		Comb	50.04	66.98	61.69	74.75
	Comb	Ava	§43.15	61.94	§49.27	64.67
		Sec	§37.64	57.06	§50.44	65.94
		Comb	§**48.89**	**66.03**	§**60.15**	**73.15**

Table 3.5: Evaluation Scores for MDPC models.

The results in Table 3.5 indicate that the translation scores on the combined-domain testset using MDPC models with combined-domain tuning are on an average 4.24 and 2.84 absolute BLEU points (2.19 and 1.4 METEOR points) lower than the corresponding scores obtained using the CD model for Zh–En translations on Testset-1 and 2, respectively. En–Zh translations demonstrate a similar behaviour of lower scores compared to the CD model. Using domain-specific tuning to translate combined-domain testsets leads to worse results when compared to subsequent CD model scores. Intuitively, the MDPC models tuned on domain-specific devsets bias the component weights towards the specific domain on which the tuning was performed. In contrast combined-domain tuning distributes the component weights

results of the table are quite comparable to the MDPC models evenly leading to better scores than domain-specific-tuned models for combined-domain test data. However, these scores still fail to match the respective scores in Table 3.4. The primary reason for lower performance of the MDPC models can be attributed to the nature of translation option combination in such models. In MDPC models, every input phrase pair is scored separately by both constituent phrase tables (the Ava model phrase table and the Sec model phrase table in our case). This leads to a larger number of translation options for an input phrase pair when compared to the CD model containing just a single phrase table. The larger amount of competing translation options might negatively interfere with the different pruning parameters (such as the maximum number of translation options or beam size in the decoder configuration), thereby leading to early pruning of some of the translation options (Bisazza et al., 2011). However, the results strongly indicate the limitation of multiple decoding paths in combining pre-trained models from different domains to produce high quality translations for combined-domain testsets, at least for the task at hand.

Comparing the results for domain-specific testsets to those in Table 3.4, the considerable effect of domain-specific tuning becomes apparent. For Testset-1, the in-domain tuning of MDPC systems provides translation scores quite similar to that of CD models for both language directions. However, using out-of-domain tuning has a negative impact on translation scores. This negative impact can easily be explained by the fact that out-of-domain tuning biases the MDPC model component weights towards the out-of-domain model, resulting in translation options being selected from out-of-domain models rather than in-domain ones. For Testset-2 we observe similar behaviour with the domain-tuned MDPC scores being consistently better than the corresponding CD scores (one with in-domain tuning) in Table 3.4. Using combined-domain tuning on the domain-specific testsets is, however, less successful compared to their CD counterparts which may again be attributed to the fact that combined-domain tuning distributes the component weights uniformly between Ava and Sec models, with the result that some translation options from the out-of-domain table are preferred during decoding.

3.4.4 CDS Results

In the fourth phase, we use our approach (CDS) of classifier-based combination of two SDS models (one for each domain) and an MDPC model. The SDS models used are tuned using in-domain devsets while the MDPC model is tuned using the combined devset. Note that the MDPC models with combined domain tuning is not the best-performing system in terms of translation quality scores, the CD model performs much better. However, the need to keep the domain-specific models separate such that new model integration is easier and more flexible directs us to prefer MDPC models over CD models. Since the MDPC models is used to translate only those sentences for which the domain is not known, we cannot use the domain-specific MDPC systems. As the results in Table 3.5 suggest, while these models are really good at translating in-domain testsets, their performance drops considerably for out-of-domain testsets. On the other hand, the models tuned on the combined-domain devsets perform similarly for both in-domain and out-of-domain testsets and hence are preferable in the current situation. Apart from being tested with combined-domain testsets, the CDS model is also tested with the domain-specific testsets to obtain a better understanding of how well this technique performs for data from each domain. Table 3.6 presents the translation results for this phase. $*, \dagger$ indicates statistically significant improvements over the corresponding scores in Table 3.4 and 3.5, respectively.

	Test	**Testset-1**		**Testset-2**	
		BLEU	METEOR	BLEU	METEOR
Zh–En	Ava	∗†57.19	45.35	∗†68.01	50.29
	Sec	∗†56.91	45.36	∗†65.30	48.66
	Comb	∗†57.12	45.36	∗†67.10	49.47
En–Zh	Ava	∗†52.20	68.69	∗†65.90	77.45
	Sec	∗†53.43	69.27	∗†64.36	77.18
	Comb	∗†52.83	68.98	∗†65.15	77.32

Table 3.6: Evaluation Scores for CDS Models.

For the combined domain testsets the CDS model (as per results in Table 3.6) provides improvements of 1.51 and 1.54 absolute BLEU points for Testset-1 and -2, respectively, on Zh–En translations over the CD model scores. The improvement figures for En–Zh translations on the combined-domain testsets are 0.85 and 0.87 absolute BLEU points for Testset-1 and -2, respectively. Looking at the domain-specific testsets (both for Ava and Sec testsets) we observe that the CDS model improves over the CD model scores by averages of 1.33 and 1.12 absolute BLEU points for Testset-1 and -2, respectively, for Zh–En translations, while the improvement for En–Zh translations are on average 0.9 and 0.8 absolute BLEU points for Testset-1 and -2, respectively. The METEOR scores show similar trends of improvements both for combined-domain as well as domain-specific testsets in both language directions. All these improvements are statistically significant at the p=0.05 level using bootstrap resampling. Since most of the sentences are eventually handled by the appropriate domain-tuned SDS translation models, the CDS setting performs better than the combined model simply by utilizing the best of both SDS models. Comparing the improvements of CDS models over the CD models between Testset-1 and Testset-2, we observe that the improvements are more significant for Testset-1. The major reason behind this difference is the fact that a much larger percentage of sentences (1.35% on Testset-1 compared to 16.15% on Testset-2) are labelled 'undecided' by the classifier for Testset-2 (Table 3.2), which results in those sentences being translated by the comparatively low-scoring combined-domain-tuned MDPC model.

It is interesting to observe the results of the domain-specific testsets when translated with the CDS model. Since the classifier and the translation models are trained on the same data, it appears that the classifier does a good job at deciding on the appropriate translation model for a particular sentence. However, as the results suggest, the CDS domain-specific testset translation scores (Table 3.6) are not quite as good as the in-domain translations (where training and test data are from the same domain) provided by the SDS models (Table 3.3), although the results are only marginally poorer (the differences are mostly statistically non-significant, with an exception of the Testset-2 Zh–En datasets). This is due to the small percentage of misclassified sentences reported in Table 3.2. In CDS models, misclassification causes a few sentences from one domain to be translated by the SDS translation model trained on the other domain, thereby degrading the quality of translation. However, the results consistently outperform the translations provided by the baseline CD model (Section 3.4), both for the domain-specific and combined-domain datasets. Hence with this approach we not only provide an alternative way of combining multiple domain-specific translation models, but also provide better translations compared to training a single generic model over combined-domain as well as domain-specific testsets.

3.4.5 SYSCMB Results

The system combination tool as described in Section 3.3.3 is used to translate domain-specific as well as combined-domain testsets from both the sets (Set-1 and Set-2) we use in our experiments. Since the SYSCMB system requires a devset for setting the internal weights within the CN and the Rescorer, we use the combined-domain devsets corresponding to each testset being translated. We prefer a combined-domain devset to a domain-specific one (Ava or Sec) so as not to bias the CNs towards one specific domain. Moreover, for fair comparison we use the same three systems (SDS-Ava, SDS-Sec and MDPC) we used in our CDS approach in the system combination setting. Table 3.7 presents the results for system combination.

Comparing the BLEU scores for each testset in Table 3.7 to that of the CDS approach in Table 3.6, we observe a drop of 2.29 and 1.47 absolute BLEU points on the combined-domain Testset-1 and -2, respectively, for Zh–En translation respectively. Similar drops are observed for the En–Zh translations on both testsets. All these drops are statistically significant at the p=0.05 level.

However, when looking at the individual component-level scores for the 3 systems

	Test	**Testset-1**		**Testset-2**	
		BLEU	METEOR	BLEU	METEOR
Zh–En	Ava	55.29	44.89	65.86	49.08
	Sec	54.13	43.98	64.61	48.07
	Comb	54.83	44.42	65.63	48.58
En–Zh	Ava	50.56	67.77	64.20	76.03
	Sec	52.46	68.85	62.49	75.35
	Comb	51.67	68.31	63.41	75.70

Table 3.7: Automatic Evaluation Scores for System Combination of SDS and MDPC models.

involved in system combination, we find that SYSCMB improves on each of the individual scores for both testsets and language directions, for the combined domain data (For Zh–En Testset-1, SYSCMB provides a score of 54.83 BLEU points in comparison to the individual scores of 41.86, 41.88 and 51.37 BLEU points for SDS-Ava, SDS-Sec and MPDC models, respectively). Hence SYSCMB improves the translation scores by 3.46 absolute BLEU points over the best-performing individual system (MDPC with combined devset tuning) in this case. We observe similar improvements for En–Zh translations on Testset-1 and Testset-2. Hence it is evident from the results that system combination is effective in the scenario for translating mixed-domain data, but not as effective as our CDS approach.

The effect of a system combination experiment is known to depend on the relative performance of the systems being combined (Sennrich, 2011). In terms of translation scores, the MDPC model is the best-performing system among the constituents involved in the combination. However, the MDPC system used in our SYSCMB approach, utilises the same phrase tables from the other two constituent systems, SDS-Ava and SDS-Sec. Therefore, the hypotheses generated by the MDPC model is very similar to those generated by either of the SDS models (depending upon the domain of the input sentence, it is similar to either of the SDS hypotheses). With two of the 3 systems generating nearly the same hypotheses, the only variation comes from the other SDS model, which being an out-of-domain model in the context generates poor hypothesis in general. Hence, during the CN-decoding words from the MDPC or the in-domain SDS model are preferred over the out-of-domain SDS models. This allows the the SYSCMB output to improve over the MDPC model scores. In contrast, the CDS model utilises in-domain SDS models to translate a majority of the input sentences, allowing the MDPC model to handle only a few unclassified sentences. Hence, it retains the quality advantage provided by the domain-specific SDS models over the MDPC model, especially for in-domain sentences. Furthermore, performance of a system combination is highly dependent on the number of input hypotheses and the complementarity of the systems providing them (Barrault, 2010). Our use of only three systems (any two of which are nearly similar for all sentences) can be a further reason for the poorer performance of the SYSCMB technique in the current setting.

For the domain-specific Ava and Sec testsets, we find a similar trend of CDS scores outperforming the SYSCMB scores for both testsets and language directions. In comparison to the CDS scores in table 3.6, the SYSCMB scores drop on average by 2.34 and 1.42 BLEU points for the Zh–En translations of Testset-1 and Testset-2, respectively. For En–Zh translations of the domain-specific testsets, SYSCMB scores drop by average of 1.31 and 1.74 absolute BLEU points for Testset-1 and -2, respectively. All these drops are statistically significant at the p=0.05 level. As in the case of combined-domain testsets, METEOR scores follow a similar trend of degradation with respect to corresponding CDS scores. Comparing the SYSCMB scores to those of the constituent systems reveals that for domain-specific testsets, SYSCMB fails to improve over the best-performing constituent systems. Considering the Zh–En translations for the Ava domain Testset-1, we find the SYSCMB score (55.29 BLEU points) to be significantly lower than the best-performing constituent system score (SDS-Ava with score of 57.42 BLEU points). Similar trends are observed for translations across both testsets and language directions and both domains. The primary reason for this degradation in the system combination performance can be attributed to the use of the combined devset to tune the weights of the CN and the rescorer. Note that the best-performing systems within the SYSCMB framework for domain-specific testsets are always the SDS models trained on in-domain devsets. However, the ultimate objective of achieving high quality translations on mixed-domain data justifies the use of combined-domain devset to tune the SYSCMB components.

For easier comparison of the effect of different systems on translation quality of the different testsets and language directions, we present a comparison of the translation scores from all the previous tables in Table 3.8. The best translation scores from each system, testset, domain and language pair are collated in this table. However, the SDS system results are left out of the comparison since they are not specifically designed to translate the mixed-domain data which forms the primary motivation of the experiments in this chapter.

Test	**System**	**Zh–En**				**En–Zh**			
		Testset-1		**Testset-2**		**Testset-1**		**Testset-2**	
		BLEU	METEOR	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR
Ava	CD	55.57	44.85	66.53	49.23	51.55	68.44	64.72	77.18
	MDPC	55.34	44.90	67.56	49.57	51.82	68.65	64.94	77.73
	SYSCMB	55.29	44.89	65.86	49.08	50.56	67.77	64.30	76.03
	CDS	**57.19**	**45.35**	**68.01**	**50.29**	**52.20**	**68.69**	**65.90**	**77.45**
Sec	CD	55.67	44.93	64.55	48.31	52.29	68.79	63.94	77.13
	MDPC	55.71	44.94	65.00	48.40	52.34	68.93	64.16	77.43
	SYSCMB	54.13	43.98	64.61	48.07	52.46	68.85	62.49	75.35
	CDS	**56.91**	**45.36**	**65.30**	**48.66**	**53.43**	**69.27**	**64.36**	**77.18**
Comb	CD	55.67	44.93	64.55	48.31	51.98	68.63	64.28	77.09
	MDPC	51.37	42.60	62.72	47.37	48.89	66.03	60.15	73.15
	SYSCMB	54.83	44.42	65.63	48.58	51.67	68.31	63.41	75.70
	CDS	**57.12**	**45.36**	**67.10**	**49.47**	**52.83**	**68.98**	**65.15**	**77.32**

Table 3.8: Comparison of different systems. The best scores for each testset are in bold.

The comparison of the scores presented in Table 3.8 clearly depicts the superiority of the CDS approach in translating mixed-domain as well as domain-specific datasets. The improvements provided by the CDS system, over all the other systems, for both testsets (Testset-1 and -2) and both language directions (Zh–En and En–Zh) are statistically significant at p=0.05 level. Moreover, comparing the scores provided by SYSCMB to the other systems, we observe that for combined-domain testsets, the SYSCMB approach provides (statistically) significantly better scores than MDPC, but are in general poorer than the CD system scores. This advantage of the SYSCMB approach is, however diminished when considering the performance on the domain-specific testsets. For domain-specific testsets, the MDPC approach provides better scores than both the CD and SYSCMB systems, although the difference is not statistically significant in every cases. Overall, this comparison shows that our CDS approach not only provides better translation scores for both domain-specific as well as combined-domain datasets, but is consistently better than all the other approaches we experimented with.

3.5 Manual Analysis

Section 3.4 compared the relative performance of the different approaches in terms of automatic evaluation metric scores. In order to substantiate the claims made on the basis of the automatic evaluation scores, we performed a subsequent manual evaluation by comparing the translations provided by our approach (CDS models) and the baseline system (CD) model. This exercise provides us with a deeper understanding of the reasons behind the better performance of the CDS systems for the current task under consideration.

3.5.1 Experimental Setup

Manual analysis is performed on a random selection of 100 sentences (50 from each domain) from the combined domain testset for En–Zh translations from Testset-1. We use three independent evaluators (native Chinese speakers) with good English skills to manually evaluate each of these 100 translations from both the systems. The evaluators were provided with the source, reference and hypothesis translations from both the systems, although they were agnostic of source of each hypothesis (i.e. they were unaware of which hypothesis came from which system). The evaluation was performed on the basis of two five-point scales representing *fluency* and *adequacy* (LDC, 2002). The adequacy scale determines the amount of meaning conveyed in the hypothesis translation in comparison to the reference translation. The five-point fluency scale indicates the closeness of the translation to natural text in the target language (Simplified Chinese in our case). The details of the scale are presented in Table 3.9. Once the fluency and adequacy of both system outputs were identified, the evaluators were further requested to mention the reason for better translation quality. While three different reasons were specified in the guidelines, the evaluators were encouraged to specify any other reason beyond the list as the cause of improvements. The last column in Table 3.9 presents the different reasons specified for our experiments. The actual guidelines for manual evaluation are presented in the appendix (cf. Appendix A).

Fluency	**Adequacy**	**Reasons for Improvement**
5=Flawless Chinese	5=All	Better Word Order
4=Good Chinese	4=Most	Better Lexical Selection
3=Non-native Chinese	3=Much	Fewer OOV words
2=Disfluent Chinese	2=Little	Other Reasons
1=Incomprehensible	1=None	

Table 3.9: Adequacy and Fluency scales for Human Evaluation of MT

3.5.2 Manual Evaluation Results

In order to estimate the reliability of the manual evaluation, we measure the Inter-annotator agreement (IA) as as it is known to be a good indicator of the reliability of a manual evaluation by different human evaluators. Fleiss' Kappa measure (Fleiss, 1971) is used to assess the reliability of the the agreement between different evaluators. Values of Kappa can range from -1.0 to 1.0, with -1.0 indicating perfect disagreement, and 1.0 denoting 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). Table 3.10 shows the average fluency and adequacy scores for the CD and CDS systems as computed on the basis of the scores provided by manual evaluators. Additionally it also provides the BLEU scores for the same set of sentences. The IA between the three evaluators for fluency and adequacy of the CD system output are 0.53 and 0.46, respectively. The IA figures for the CDS system outputs are 0.39 and 0.54 for fluency and adequacy respectively. The range of the different IA scores indicates fair to good agreement between the different human evaluators for both systems and metrics (fluency and adequacy).

system	BLEU	Avg. Fluency	Avg. Adequacy
CD	50.91	2.91	3.46
CDS	52.19	3.19	3.74

Table 3.10: BLEU scores and Average Fluency and Adequacy Ratings from Manual Evaluation for CD and CDS model outputs

The adequacy and fluency ratings by the human evaluators (Table 3.10) clearly indicate that the human evaluators found the CDS system translations to be more fluent and adequate in comparison to the CD system translations. While this substantiates the claims based on automatic evaluation (cf. Section 3.4), we also wanted to gain a deeper insight into the actual reasons as to why CDS translations score appear better than those from the CD models. Comparing the translation hypotheses from both systems and collating the reasons for improvements identified by the evaluators we found that out of 100 sentences, 48 had the same translations from both systems. Of the remaining 52, the CDS translation systems were better for 34 translations while for the other 18 translations, the CD system scored better. Since we had three sets of scores from the evaluators, we identify a system better by majority voting of the scores. Hence, a translation is considered better than its counterpart only when two out of three evaluators mark a sentence better (in terms of fluency or adequacy scores). Table 3.11 shows a category-wise breakdown of the manual results, with the first column indicating the overall status of the CDS translation compared to the CD translation. Again, using similar majority voting we found that of these 34 'better' translations, 17 were due to better word ordering, 10 were due to better lexical selection of words or phrases and 7 were simply better due to inclusion of a 'keyword' compared to the CD translation. The 'keywords' considered here could be any word in the translated sentence whose presence or absence might change the meaning of the sentence. Of the 18 translations, where the CDS model actually performed worse than the CD model, 7 were worse in terms of lexical selection, 8 in terms of poor word ordering or syntax, and 3 due to missing one or more important keywords.

CDS Trans	**Category**	**Ava**	**Sec**	**Total**
Better	Better Lexical Selection	6	4	10
	Better Word Order	11	6	17
	Keyword Present	4	3	7
Worse	Worse Lexical Selection	3	4	7
	Worse Word Order	5	3	8
	Keyword Absent	2	1	3
Similar		28	20	48

Table 3.11: Categorical Distribution of Manual Analysis Observations

Observing the breakdown of the three categories, 50% of the better translations (17 out of 34) are due to better word order, which indicates that domain-specific word ordering is important to the translations in the current task. This observation not only supports our assumption that the CDS model produces better translations, but also justifies our approach of combining multiple models instead of training a single model on the entire combined data. Finally, a majority of the sentences translated by the CDS system are found to be better than those produced by the CD system, thus corroborating our claim about the superiority of the CDS system over CD models.

3.6 Observations

The objective of the experiments conducted in this chapter was to compare the effect of translation combination from domain-specific SMT systems on translation quality for a scenario where the data to be translated comes from a mixture of different domains. We conducted our experiments using data from two different domains– Ava and Sec– and used combined-domain testsets for testing. Section 3.3.3 presented the five different experimental setups we used to evaluate and test the different aspects of this scenario. Observing the results presented in Section 3.4, we found that SDS models trained on in-domain data were best at translating in-domain testsets, i.e. provided the best scores in terms of automatic evaluation metrics. However, the same system performed poorly when translating out-of-domain testsets. Considering the combined-domain testsets, the translation quality is found to be somewhat in the middle of the high and low scores produced for in-domain and out-of-domain testsets, respectively. This behaviour is uniformly observed across two language directions and two slightly different testsets. Accordingly, the initial observation from the first phase of experiments with SDS models confirm an existing hypothesis, namely that domain-specific models are best for translating in-domain data and quality suffers for out-of-domain translation.

Using the CD models, the translation quality of the combined-domain testset improved dramatically over the corresponding scores with SDS models. However, for domain-specific testsets CD model translations were slightly worse than the corresponding translations generated by SDS model. While the slight drop in translation quality could be attributed to the increase in generality of the model (data from both domains combine to form a single model), the same reason leads to much better translation for combined-domain testsets. Another important observation in this phase was that domain-specific tuning had a little effect on the actual translation quality for CD models. Since data from both domains are concatenated into one single model, and MERT tuning only provides global weights to the model, domain-tuning fails to bias the model towards one domain. This being the conventional approach of handling multi-domain data, is considered as the baseline model for our experiments. Moreover, although this approach provided good quality translation for combined-domain data, it suffered from the problems of lack of flexibility and ease of maintenance. Considering the industrial relevance of the scenario we aim to address, flexibility and maintenance issues are of considerable importance. These issues motivated us to look for a more effective solution to deal with combined-domain data translation.

In the third phase, we used a more sophisticated technique of combining multiple domain-specific models into one system by utilizing the multiple decoding path feature in Moses. Unlike the CD model, here the system maintains separate translation, reordering and language models for two different domains. This makes it more flexible and easier to maintain in contrast to the CD model. However having multiple model components makes it sensitive to domain-specific MERT tuning. As MERT can set weights separately for individual domain-specific models, domain-specific tuning biased the weights towards one particular domain. Hence these scores were similar to those of the SDS models. The domain-specific testset translations provided by domain-tuned MDPC models were on a par or slightly better (the difference being statistically non-significant) than the domain-specific testset translations provided by the CD models. However, for combined-domain testsets, the results were poorer to those provided by the CD model in the previous phase. This leads to the observation that while domain-tuned MDPC models are good for in-domain testset translation, they are not well-suited to handle combined-domain testsets even when combined-domain tuning is deployed. Combined-domain tuning results in larger numbers of competing translation options which might negatively interfere with the pruning parameters of the decoder configuration, thereby leading to early pruning of some of the translation options.

In the fourth phase of our experiments we test out our approach using an automatic classifier to combine two SDS models (one for each domain) and one combined-domain tuned MDPC model to handle a few unclassified sentences. Figure 3.7 clearly indicates that this approach works better for both mixed-domain as well as domain-specific testsets. Since the classifier identifies the domain of the input sentences prior to translation, the majority of sentences are routed to the appropriate in-domain SDS models for translation. As the different tables in Section 3.4 suggest, such in-domain data is best translated by SDS models. Hence our approach simply tries to use the best of both SDS models. However, the classifier accuracy plays an important part here, and is evident when comparing the results between two of our testsets. The margins of improvement are better for Testset-1, as the classifier does a better job at classifying the sentences, and fails only for a few sentences (27 out of 2000). For Testset-2, where the classifier fails to classify a considerable number

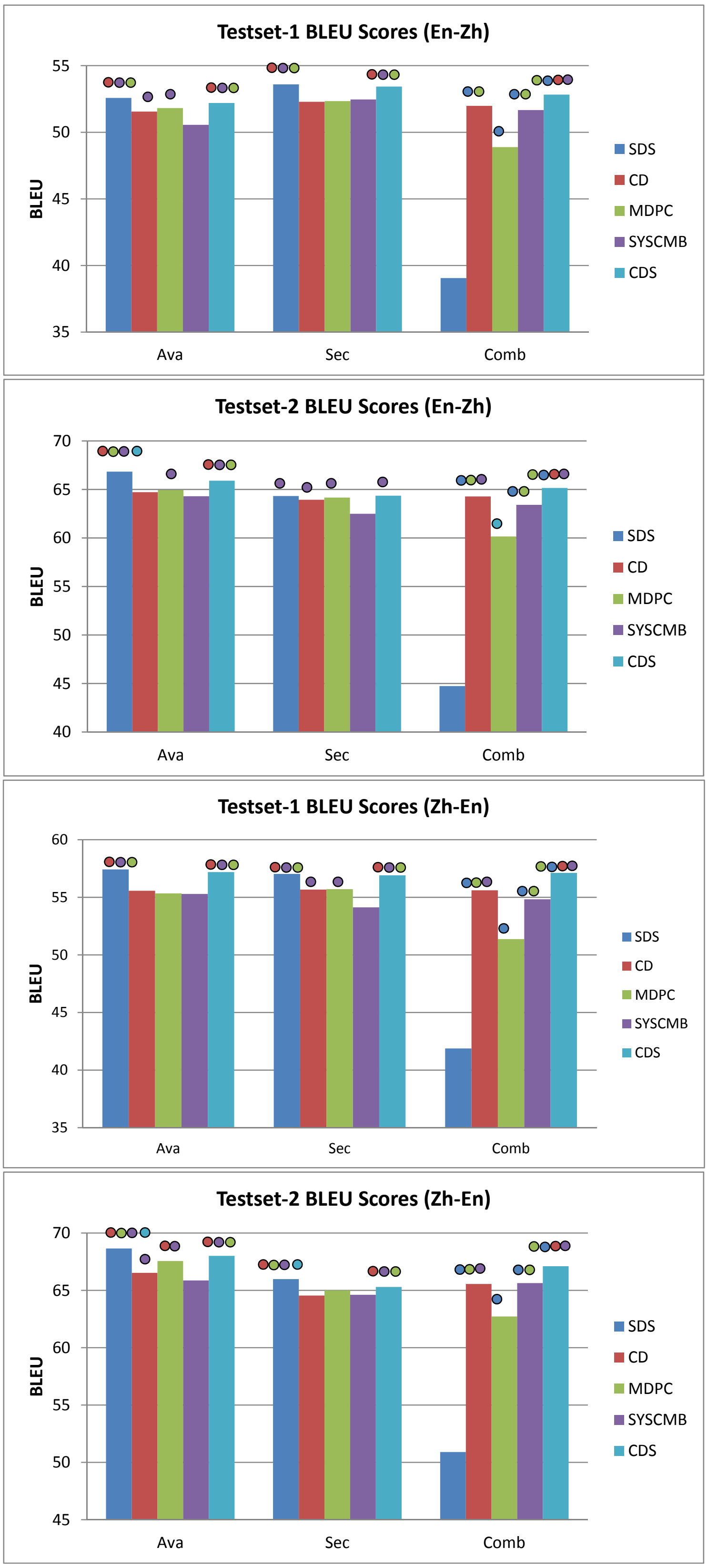


Figure 3.7: Comparison of BLEU scores for testsets generated by SDS, CD, MDPC, SYSCMB and CDS Models

of sentences (323 out of 2000), these sentences are translated by the relatively poor MDPC model leading to lower improvements. However, we observe the improvements to be consistent over language directions and testsets, thus demonstrating the superiority of the approach over CD or MDPC models.

In the final phase, we used a state-of-the-art system combination approach (SYSCMB) to perform a word-level combination of the different system hypotheses. Using the same three systems as is used in the CDS approach, we combined 1-best hypotheses from each of them using a MBR-CN framework (Du et al., 2010). Our experiments in Section 3.4 revealed that SYSCMB is effective in the current setting in improving translation quality of mixed-domain data over each of the constituent models. However, the small number of systems involved in combination and the similar nature of the hypotheses (or the lack of complementarity) provided by each of them made the SYSCMB approach less effective than our CDS technique. For domain-specific testsets our CDS models again outperformed the SYSCMB translation scores for both testsets and language directions. Moreover, for domain-specific testsets, the SYSCMB approach failed to improve the translation quality over the best-performing constituent systems (usually the domain-specific SDS models).

Finally the manual evaluation conducted on a selection of 100 sentences further confirmed our results by showing CDS translations to be better than CD translations. When investigating the reasons for the improvement, the manual evaluation revealed that most of the improvements were due to better word ordering. This observation indicates that translations from the SDS models are better suited than those from the CD models, thereby supporting our decision to combine them using a classifier.

3.7 Summary

Both the automatic and manual evaluation of our experiments reveal the advantage of system-level combination of two independent domain-specific systems in translating data from a mixture of domains. While outperforming generic models trained on a concatenation of multi-domain data, this method also provided better results than sophisticated state-of-the-art combination techniques like multiple decoding paths or MBR-based system combination techniques. The improvements were observed not only for combined-domain testsets, but also for domain-specific testsets, and these were consistent across language directions and different testsets.

Not only does this approach provide better translation results, but it also provides a flexible framework for the addition or removal of domain-specific models in a scenario where mixed-domain data translation is required. Moreover, since domain-specific models are smaller than combined-domain models, this approach further allows easier maintenance of the individual models. Lastly, we present the first research question here, which forms the primary motivation for the experiments presented in this chapter:

> **RQ1:***Given a mixed domain and a set of mixed-domain training data, does a combination of translations from different domain-specific models, each trained on a subset of the data, provide better translation quality when compared to those from generic models, trained on the full dataset?*

Our experimental results and observations help us to answer the first research question (**RQ1**) in the affirmative.

3.7.1 Contributions

The main contributions of this chapter are as follows:

- We have successfully shown that combining domain-specific models provides better translation quality compared to generic models, when translating data from a mixture of domains.

- We have successfully used a sentence-level classifier to identify the domain of input sentences and route them to the appropriate models.

- The results indicate that classifier-based combination works better than combining multiple domain-specific models using the Multiple Decoding Path framework in Moses as well as using a CN-based system combination approach.

- Our method provides a flexible framework to add, remove or re-use existing SMT systems to build customised SMT systems at least for two domains which can be distinguished with a high level of accuracy.

In the next chapter, we focus on another industrially relevant domain adaptation scenario, where we try to translate user-generated forum content from the web. Since forums are by nature monolingual, we use domain-adaptation techniques to utilise models trained on translation memories to translate forum content. Although the content in the forums is broadly in the same domain as the translation memories, there are distinct stylistic differences between the two. Hence we employ normalisation and adaptation techniques to systematically handle these differences and investigate whether translation quality improves.

Chapter 4

Domain Adaptation Guided by Out-of-Vocabulary Word Reduction

In the previous chapter, we described a domain-adaptation approach for handling mixed-domain enterprise content translation by using a combination of domain-specific SMT systems. Our experiments showed that domain-specific data was best translated by SMT systems trained on in-domain data and this particular property of in-domain SMT systems was leveraged in our approach to translate mixed-domain data. However, this approach only works well when sufficient in-domain training data is present to train domain-specific SMT models. Unfortunately, in some real-life scenarios sufficient in-domain training data is not always available. Adapting SMT systems to such a scenario, where no parallel in-domain data is present opens up a different set of challenges compared to what is presented in Chapter 3. This is exactly the scenario in which we focus our domain adaptation techniques discussed in this chapter.

In contrast to the enterprise-quality corporate content used in Chapter 3, we focus on user-generated web-forum content for our experiments in this chapter. In recent years, SMT technology has been widely used to translate large amounts of professionally edited enterprise quality online content (Knowledge base articles, online user help files etc.). However, not much research has gone into adapting SMT technology to the translation of user-generated content on the web. While translation of online chats (Flournoy and Callison-Burch, 2000) have received some attention, there is surprisingly little work on the translation of online user-forum data,[1] despite growing interest in the area (Flournoy and Rueppel, 2010). One of the major challenges in using SMT for forum data translation is the lack of 'forum-style' parallel training data. Forum data by nature is monolingual and hence cannot be used to train the translation models in SMT systems. Therefore, we use available parallel training data in terms of Symantec enterprise translation memories (TM) to train our systems. Symantec TM data, being a part of enterprise documentation, is professionally edited and, by and large, conforms to the Symantec controlled language guidelines (Doherty, 2012),[2] and is significantly different in nature from the user-forum data, which is loosely moderated, does not use controlled language at all and is often 'noisy', taking liberties with spelling, punctuation and the use of acronyms, abbreviations and emoticons. This difference between the training and the test datasets necessitate the use of domain adaptation methods for better translation quality.

To identify the differences between the TM and forum data, we focus on the out-of-vocabulary (OOV) words in the English forum data with respect to the source side (English) of the TM data. OOV words are defined as the words which are present in the test domain (web-forums) but not in the training domain (Symantec TMs). Once the OOVs are identified, we use a semi-manual classification technique to classify them into different categories which require independent attention. In order to optimally handle each individual category, different techniques are developed to make the forum-based testsets better resemble the training data. Broadly

[1] The only commercially known use case is that of TripAdvisor (http://www.tripadvisor.com/) which uses SMT to translate user reviews on their site. Mitchell and Roturier (2012) presents manual evaluation results for Norton forum translation using Bing Translator.

[2] Some parts of the TM data e.g. software strings do not conform to controlled language guidelines

these techniques may be divided into two different classes: *normalisation techniques* which comprise regular-expression based masking and spell-checking to reduce the number of OOVs in the test domain, and *supplementary data selection techniques* that utilise out-of-domain parallel corpora to enhance the existing baseline SMT models. We use two different testsets each resembling different degrees of noise in the forum-data to observe the individual and combined effect of normalisation techniques. Data selection is used both additively as well as in contrast to different normalisation methods to observe its effect on the testsets. Our experiments reveal that both normalisation and supplementary data selection are effective in improving translation quality of forum content when used additively. Furthermore, for moderately noisy data, data selection alone proves to be as effective as the combination of different normalisation techniques, although the two techniques address different classes of OOVs. However, for really noisy data the normalisation effort pays off. Our experiments are conducted for English-French (En–Fr) and English-German (En-De) language pairs considering only translations from English as language direction.

The rest of the chapter is organised as follows: Section 4.1 presents the motivation of our work and a brief background. Section 4.2 details the normalisation and data selection techniques used in the experiments, followed by Section 4.3 describing the datasets used along with an account of the OOV rates in each of them. Section 4.4 presents the results and analysis of the different stages of experiments using automatic evaluation metrics followed by a description of the manual evaluation experiments and corresponding results in Section 4.5. Section 4.6 presents general observations and the chapter concludes with a summary of the main findings in Section 4.7.

4.1 Motivation

Web-forums are online discussion sites where people can hold conversations in the form of posted messages. The advent of web 2.0 communication channels (community forums or social media) has led to users taking a more active part in generating software documentation (Roturier and Bensadoun, 2011). Forums are used as a platform where savvy users communicate with other users regarding specific problems and their solutions pertaining to software products or services. Web-forums are therefore, rich sources of user-generated content on the web. The increasing popularity of technical forums motivated major IT companies like Symantec[3] to create and support forums around their products and services. For individual users or larger customers, such forums provide an easy source of information and a viable alternative to traditional customer service options. Being a multinational company, Symantec hosts its forums in different languages (English, German, French etc), but currently most of the content is siloed in each language. Moreover, there is much more content on the English forums compared to the non-English forums and the number of users is substantially greater in the English forums. Clearly, translating the forums to make information available across languages would be beneficial for Symantec as well as its multilingual customer base. This forms the primary motivation of our efforts in forum data translation.

As already stated in the previous section, forum data by nature is monolingual and hence impossible to train SMT models on. Hence we use Symantec TMs to train the baseline SMT models for our experiments. Broadly speaking, both the TMs and the forums are from the same domain– comprising content on different products and services offered by Symantec. However, in comparison to professionally edited TM content, the Symantec forum data is often more noisy, taking some liberty with commonly established grammar, punctuation and spelling norms. Furthermore, the TM content follows a high level of moderation and quality control and some parts of it must conform to the Symantec controlled language guidelines. On the other hand,

[3] http://www.symantec.com/en/uk

the forum content is only lightly moderated and conforms to very basic publication-quality guidelines.[4] Hence despite being from the same technical domain, there is a significant difference in style and vocabulary between the training and the test data. Normalisation techniques are known to be effective in reducing OOV rates in different NLP applications (Yvon, 2010), while data selection methods are used to increase the coverage of the in-domain models. This motivates our effort to systematically reduce the training and target domain difference through the use of both normalisation and supplementary training material acquisition techniques.

As previously stated in Chapter 2 (cf. Page 32) the technique of using 'out-of-domain' datasets to supplement 'in-domain' training data has been widely used in domain adaptation of SMT. Hildebrand et al. (2005) utilised such an approach to select similar sentences from available bitext to adapt translation models, which improved translation performance. Habash (2008) used spelling expansion, morphological expansion, dictionary term expansion and proper name transliteration to enhance or reuse existing phrase table entries to handle OOVs in Arabic–English MT. More recently an effort to adapt MT by mining bilingual dictionaries from comparable corpora using untranslated OOV words was carried out by Daume III and Jagarlamudi (2011). Our current line of work is related to the work reported in Daume III and Jagarlamudi (2011) and that of Habash (2008). In our case however, the target domain (web-forum) is different from the training data (Symantec TMs) more in terms of style rather than actual domain (Banerjee et al., 2011b). Secondly, in contrast to mining comparable data for bilingual dictionary extraction (Daume III and Jagarlamudi, 2011), we exploit sentence pairs from available parallel training data to handle untranslated OOVs. Moreover, mining supplementary parallel data guided by OOVs is used as a technique complementing the normalisation-based approaches to reduce specific type of OOVs in the target domain. We classify OOVs into different categories and treat each of them separately. In contrast to extending the phrase table entries (Habash, 2008), our normalisation methods mostly com-

[4] http://community.norton.com/t5/user/userregistrationpage

prise pre- and post-processing techniques. Finally we also present a comparison between the normalisation and supplementary training data acquisition techniques for different error-density based scenarios of the target domain. To the best of our knowledge, the use of 'domain-adapted' spell-checkers to reduce OOV rates in the target domain is novel, and is one of the other main contributions of this chapter.

4.2 Approach

Considering the lack of forum-style parallel training data, we use the available Symantec TMs to train our SMT systems in order to translate forum data. As mentioned in Section 4.1, there is significant difference between these two datasets despite being roughly from the same domain. However, to effectively address this, the difference needs to be quantified. In our approach we quantify this difference in terms of the number of out-of-vocabulary (OOV) words i.e. the words which occur in the forums but are absent from the TM-based training sets. Once the OOVs are identified a manual inspection of the different OOV types allowed us to classify most of them into the following four categories:

1. Maskable Tokens (MASK): URLs, paths, registry entries, email addresses, memory locations, date and time tokens and IP addresses or version numbers.

2. Fused Words (FW): Two or more valid tokens concatenated using punctuation marks like '.' or ','. [5]

3. Spelling Errors (SPERR): Spelling errors or typos.

4. Valid Words (VAL): Valid words not occurring in the training data.

5. Non-Translatable (NTR): Tokens comprising of standalone product and service names and numbers (not part of Category-1 tokens) which ideally should not be translated.

[5] Recent investigation reveals that some of the FW tokens are introduced due to the incorrect normalisation and sentence splitting mechanism used to pre-process the forum content.

In order to ensure that the categories are general enough and cover nearly all types of OOVs we use a corpus of forum data collected from the English forums as a representation of the forum-style data. A unigram language model created using the source-side of the TM-based training data is tested against this English forum data to obtain a list of possible OOVs in the test domain. Section 4.3.1 provides further details on each of these datasets.

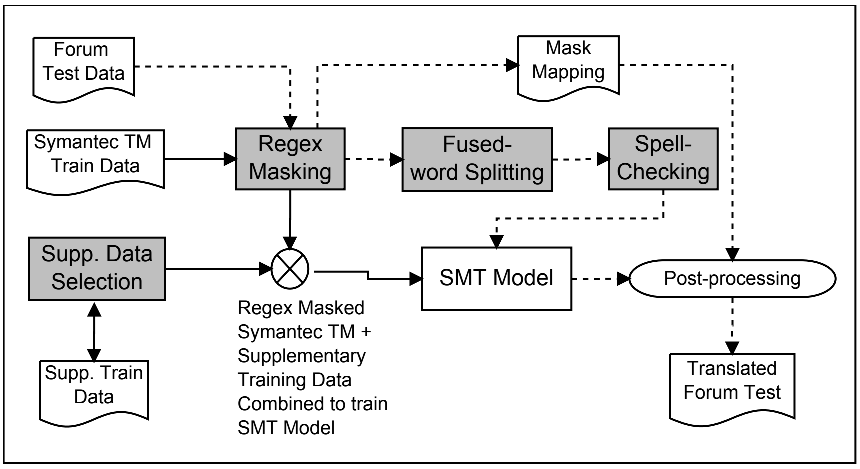


Figure 4.1: Normalisation and Supplementary Data Selection Techniques on Training and Testsets

The primary objective of our approach being to systematically reduce the number of OOVs, we develop individual techniques to handle each categories. Figure 4.1 presents a block diagram of how the different techniques are applied to the training and testset data. Supplementary training data acquired by the technique mentioned in Section 4.2.4 is combined with regex-masked (Section 4.2.1) TM-based training data to train the SMT models. The testset is passed through the regex-masker, fused-word splitter and spell-checker in the pre-processing step before being translated by the SMT model. In the post-processing step, the mapping file generated by the regex-masking technique is utilised to replace the place-holders with the unique tokens.

4.2.1 Regular Expression-based Masking

The MASK category tokens are handled using a set of regular expressions (cf. Appendix B) specifically designed to identify and mask all such tokens in the text with unique place-holders. Table 4.1 presents examples of how the masking affects a set of sentences.

Type	Sentence
URL	1. Go to **http://community.norton.com/norton/board/message?board.id=nis_feedback&message.id=53509#M53509**
	1. Go to ⟨**url_ph**⟩
E-mail id	About 2 weeks ago I emailed **pse_support@symantec.com**
	About 2 weeks ago I emailed ⟨**email-id**⟩
IP-Address/ Version No.	About: says that I am running v**3.5.2.11**.
	About: says that I am running v⟨**ip_ver**⟩.
Dates	Actually I did a system restore on **3/21/09**.
	Actually I did a system restore on ⟨**date**⟩.
Registry Keys	Delete **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Network\Config** and reboot.
	Delete ⟨**winreg**⟩ and reboot.
Path Entry	Do you find a folder Norton.Gadget in the following path **C:/Program files/Windows Sidebar/Gadgets.**
	Do you find a folder Norton.Gadget in the following path ⟨**winpath**⟩.

Table 4.1: Example of Masking Techniques on different types of MASK tokens. Maskable tokens and place-holders in each example are in bold

As observed in the examples in Table 4.1, some tokens in this category are multi-word tokens (e.g. Windows Path or Registry entries). The masking technique converts these multi-word tokens into a single place-holder token which is treated by the SMT system as a single word. This ensures that the token cannot be broken up during the translation phase. Furthermore, this technique also prevents a subpart of a token from inadvertently being translated. The masking is uniformly carried out both on the training data (prior to training) and on the test data (prior to translation) in the pre-processing phase. A one-to-one map of the line number, actual token and corresponding mask is maintained in the pre-processing phase. This mapping is again used in the post-processing phase (after translation) to replace the place-holders in the translated sentences with the actual tokens.

4.2.2 Fused Word Splitting

The FW tokens comprise two or more valid words fused using a punctuation symbol. In the present scenario, we only handle FW tokens which are fused using the *period* (.) and the *comma* (,) symbols only.[6] In order to handle such tokens we start by identifying all tokens which have a period or a comma symbol flanked by alphabetic characters using simple regular expressions. The objective is to split these tokens into its constituent sub-parts such that they could map to words present in the training data thereby reducing OOVs. Splitting the comma-fused tokens are comparatively simpler as we can assume that every instance of comma should have a space after it without any loss of generality. However splitting the period-fused tokens are not that simple since a large number of valid file names, website names or abbreviations (e.g. N.I.S., explorer.exe, shopping.aol.com, etc.) are essentially FW tokens but should not ideally be split at the period symbol. Hence to reduce such instances of 'false positives' we use heuristics based on the training data and a few other resources to identify the valid ones. Lists of known file extensions (e.g. exe, jar, pdf, etc.)[7] and website domain extensions (e.g. com, edu, net, gov, co.uk, etc.)[8] are used to filter out file names and website names. Finally we used a dictionary built on the training data. Every split was validated against the dictionary, with the constraint that all its constituent splits had to occur in this dictionary. This normalisation is only applied on the dev and testsets as the TM training data is found to be clean of such fused-words. However recent investigation has revealed that some of the fused word tokens have occurred due to an erroneous sentence splitting mechanism used in the pre-processing phase (primarily normalising the character '...' to a single '.'). But since FW tokens constitute a small percentage of the OOV tokens identified on the forum data (as per the figures in Table 4.4), this should have a negligible effect on actual translation quality as is revealed in the experimental

[6] Tokens fused by comma and period are the most commonly available FW tokens in the forum data.

[7] http://en.wikipedia.org/wiki/List_of_file_formats

[8] http://en.wikipedia.org/wiki/List_of_Internet_top-level_domains

results in Section 4.4.

4.2.3 Spell-checker-based normalisation

Spelling errors or typos (SPERR) are a common characteristic in any user-generated loosely moderated dataset. In order to reduce this category of OOVs, we employ automatic spelling correction to replace the erroneous spellings with their most probable correct alternative. For the spell-checking task we used a combination of two off-the-shelf spelling correction toolkits. Using the 'After the Deadline toolkit' (AtD)[9] as our primary spell-checker, we also used a Java wrapper on Google's spellchecking API[10] to supplement the AtD spell checking results. 'After the Deadline' offers an open-source authoring aid offering contextual spell-checking, real word error detection as well as grammar checking for different languages through the use of large language models. We only use the contextual spelling suggestions from the toolkit to handle our spelling correction requirements. The off-the-shelf version of the spell-checking toolkit works with bigram language models trained on data from Wikipedia[11] and project Gutenberg[12] to identify spelling errors and provide context-sensitive suggestions. In addition to the language models it also uses a spell-checker dictionary comprising about 125,000 words collected from public domain word lists[13] and different blogging sites on the web.

While the off-the-shelf version of the spell-checkers works well for most of the spelling errors in general-purpose English, it flags a lot of 'in-domain' (technical) words in the Symantec datasets we use for our experiments. Hence we adapt the spell-checker to the domain. This is achieved by supplementing the existing language models in the spell-checker with 'in-domain' data at our disposal. Further details of the adaptation task and the performance of the spell-checker is presented in Section 4.3.3. The adaptation of the spell-checker helped us to eliminate most of

[9] http://open.afterthedeadline.com/

[10] http://www.google.com/tbproxy/spell?lang=en&hl=en

[11] http://en.wikipedia.org

[12] http://www.gutenberg.org/

[13] http://wordlist.sourceforge.net/

the false positives flagged by the original unadapted spell-checker. The errors flagged by the spell-checker are replaced with the highest ranking suggestion from the spell-checker. Since the spelling errors are OOVs, by correcting them we aim to map them to existing words in the parallel training data thus improving the coverage in the translations.[14] The spelling corrections are applied only to the testsets to ensure a reduction in the number of spelling error-based OOVs.

4.2.4 Supplementary Data Selection

To take care of the VAL tokens which are valid words but absent in the training data, we explore techniques of mining supplementary data to improve the chances of successfully translating these tokens. We use the following freely available parallel data collections as potential sources of supplementary data:

1. Europarl (Koehn, 2005): Parallel corpus comprising of the proceedings of the European Parliament.

2. News Commentary Corpus: Released as a part of the WMT 2011 Translation Task.[15]

3. OpenOffice Corpus: Parallel documentation of the Office package from OpenOffice.org, released as part of the OPUS corpus (Tiedemann, 2009).

4. KDE4 Corpus: A parallel corpus of the KDE4 localisation files released as part of OPUS.

5. PHP Corpus: Parallel corpus generated from multilingual PHP manuals also released as part of OPUS.

6. OpenSubtitles2011 Corpus:[16] A collection of documents released as part of OPUS.

[14] In some cases the correct spelling offered by the spell-checker may not appear in the training data, but those cases are left out of the scope of current normalisation techniques and may be handled by data selection techniques (cf. Section 4.2.4)

[15] http://www.statmt.org/wmt11/translation-task.html

[16] http://www.opensubtitles.org/

7. EMEA Corpus: A parallel corpus from the European Medical Agency also released as part of OPUS corpus.

To select relevant parallel data, we query each of the parallel corpora with the VAL OOV words and add sentence pairs containing the OOVs into the existing 'in-domain' parallel corpora. During the selection process, the number of parallel sentences selected for any particular OOV item is restricted to a threshold of 500 for En–De and 67 for En–Fr. This is done to limit the size of the selected 'out-of-domain' supplementary data such that it does not exceed the size of the TM-based (in-domain) training data. The target sentences of the selected parallel data are also added to the language model to ensure language model adaptation.

Corpora	**En–De**			**En–Fr**		
	Selected Sent.#	Total Sent.#	OOV (%)	Selected Sent.#	Total Sent.#	OOV (%)
OpenOffice	1,006	41,939	1.52	879	37,289	1.59
KDE4	8,568	220,905	15.09	8,170	206,741	14.97
PHP	531	39310	1.17	730	41,563	1.72
OpenSubs	449,697	4,649,247	79.54	462,083	12,483,718	88.80
Europarl	241,395	1,721,980	59.70	155,747	1,809,563	61.55
News-Com.	40,731	135,758	44.44	35,062	115,085	43.69
EMEA	88,642	1,098,635	13.37	37,835	1,083,669	13.83
TOTAL	830,570		87.55	700,506		92.13

Table 4.2: Number of Sentences selected from supplementary training data and percentage of category-4 OOVs covered by each selection

Table 4.2 reports the number of sentence pairs selected from each of the different supplementary datasets alongside the total number of sentences in each corpus. The table also reports the percentage of OOVs covered by the selections. In total the selection process covers 87.55% and 92.13% of VAL OOVs for the En–De and En–Fr language pairs respectively. As is apparent from the OOV coverage percentages Europarl, Open-Subtitles and News Commentary feature the maximum coverage of this category of OOVs amongst all the different supplementary corpora.

4.2.5 OOV Tokens Unsuitable for Translation

The last remaining category of OOVs (NTR) represents tokens for which translation was usually not necessary. Most of these comprised product or service names, names of the forum users or numeric tokens. This class of tokens is not explicitly handled since due to their absence from the training data (and hence from the phrase table), they would be preserved during the translation process in the standard SMT setup.

4.3 Experimental Setup

In this section we present the details of the different datasets used in our experiments along with a measure of their OOV rates. We describe in detail the adaptation process of third-party automatic spell-checkers used to handle SPERR category of OOVs and a detailed account of the different experiments performed.

4.3.1 Datasets

The primary training data for training our baseline SMT models consist of En-De and En–Fr bilingual datasets in the form of Symantec TMs. In contrast to these bilingual datasets, monolingual forum posts collected from the German and French Symantec online forums are also available. These datasets represent the target domain of our experiments and hence are used for training the language models. However the small size of the German and French forum data prompts us to use them in combination with the target side of the TM-based training data for language model training to ensure better coverage. In addition, we also have a large collection of posts from the original Symantec English forums acquired over a period of two years which is used to compute the initial list of OOVs with respect to the training data. It is on this particular list the OOV categorisation is performed. Table 4.3 reports the amount of data used (in terms of number of sentences) and the average sentence length (ASL) for all our experiments.

As reported in Table 4.3, we use two different testsets, for our experiments.

	dataset	**En–De**			**En–Fr**		
		Sent. Count	**En ASL**	**De ASL**	**Sent. Count**	**En ASL**	**Fr ASL**
Bi-text	Training	832,723	12.86	12.99	702,267	12.42	14.86
	devset	500	16.14	16.02	500	16.14	17.86
	Generic testset	2,022	13.49	12.68	2,022	13.49	15.52
	Noisy testset	600	19.95	20.48	600	19.95	23.73

	Forum Data	**Sent. Count**	**ASL**
Mono-lingual	English	1,129,749	12.48
	German	42,521	11.78
	French	41,283	14.82

Table 4.3: Number of sentences and average sentence length for training development, test and monolingual forum datasets

The first one (generic testset) was randomly chosen from the English forum data and hence represented general 'forum-style' content. However, in comparison to the general normalisation techniques, we are particularly interested in the effect of spell-checking on the translation quality of forum content. Therefore, a second testset (noisy testset) was selectively chosen such that it simulated a higher proportion of spelling-errors (one spelling-error in every two sentences), which may be typical of some forum posts. Once the generic testset was randomly selected from the English forum data, the entire remaining data was checked using an automatic spell-checker. A set was randomly selected from these flagged sentences followed by a manual review to ensure that the testset consisted of valid spelling-errors and few false flags. Both the testsets were manually translated following basic guidelines for quality assurance. The randomly chosen devset was translated using Google Translate,[17] and manually post-edited by professional translators following guidelines[18] for achieving 'good enough quality'.

Table 4.4 depicts the percentage of the OOV word categories in the English forum data and the two testsets with respect to the En–De and En–Fr TM-based source datasets. The category-based percentages are computed by considering the count of OOV tokens in the category divided by the total number of OOV tokens in the data. Comparing the category-wise percentage figures on the two testsets (generic

[17] http://translate.google.com/

[18] http://www.translationautomation.com/machine-translation-post-editing-guidelines.html

OOV Type	En–De			En–Fr		
	Forum (%)	Generic (%)	Noisy (%)	Forum (%)	Generic (%)	Noisy (%)
MASK	25.82	21.33	10.61	25.47	19.29	10.33
FW	8.89	4.11	2.23	8.76	3.71	2.17
SPERR	10.59	9.48	54.62	10.45	8.57	53.17
VAL	6.38	14.06	12.84	6.74	14.57	13.50
NTR	48.32	47.87	19.69	48.58	51.00	20.83

Table 4.4: Category-based percentage of OOVs in the English forum and two test datasets. The numbers in each column represent percentage figures

and noisy) clearly show the distribution of the categories in generic testset is similar to that of the English forum data. The noisy testset shows a higher percentage of SPERR tokens as it had been consciously designed to have high spelling-error density. The figures also depict the relative importance of the specific OOV categories in forum style data with non-translatable (NTR) and maskable tokens (MASK) covering nearly 75% of the OOV tokens. The actual counts of MASK, FW and SPERR tokens are same for each dataset across both language pairs (since we translate from English, and the English dev and test set is same for En–De and En–Fr), but the difference in percentage is due to the total number of OOV tokens being different for each language pair. The VAL, NTR and the total OOV tokens are however dependent on the training data (which are different for En–De and En–Fr) and hence are different across language pairs for each dataset.

4.3.2 Pre-Processing and Post-Processing

Prior to training, all the bilingual and monolingual data are subjected to tokenisation and lowercasing using the standard Moses pre-processing scripts. As elaborated in Section 4.2 we use the regex-based normalisation, fused-word splitting and spell-checking techniques on the test data in this phase. Since we replace the MASK category tokens with unique place-holders (cf. Table 4.1) with an objective of reducing a multi-word token into a single one, the standard Moses tokeniser is modified to ensure that the place-holder tokens themselves are not tokenised. The pre-processing step also involves the process of creating a map file for storing the mapping between the actual tokens and the place-holders. In the post-processing step, this map file is used to replace the place-holders with the corresponding tokens. For target sentences having multiple place-holders of the same type, the corresponding actual tokens are replaced in the same monotonic order in which they appeared in the source. Furthermore, the forum content is subjected to basic cleanup in the pre-processing step, which involve removing smilies, emoticons, junk characters and repeated uses of punctuation symbols (E.g. '!!!!!!!' or '............' or '——————-').

4.3.3 Spell-Checker Adaptation

Observing the effect of spelling corrections is one of our additional objectives in this phase of experiments. But as already stated in Section 4.2.3, the off-the-shelf version of the spell-checker is trained on generic datasets which results in flagging a large number of false positives (valid 'in-domain' words) when used directly on our domain-specific datasets. Hence adapting the spell-checking toolkit to improve the quality of spell-checking plays an important role in our experiments.

Since our datasets are primarily from the IT domain, a large number of in-domain words are flagged by the off-the-shelf version of the tool. In order to avoid that, we use the source side of our parallel training data based on Symantec TMs along with a glossary of domain-specific words to enhance the existing spell-checker dictionary. Additionally the TM-based training data is used along with the existing general domain data to retrain domain-specific language models for the spell-checker. These techniques essentially help in adapting the spell-checker to the IT domain resulting in fewer false positives during the spell-checking process. In order to further improve the coverage of the AtD spell-checker, we use a second pass of spell-checking based on the Google Spell-checker. The Google spell-checker is an online service and hence cannot be domain-adapted specifically to our domain. Hence, in this case a list of possible false-positives are used to ensure that they are not automatically corrected by the spell-checker.

In order to evaluate the spell-checking configurations, we must consider two aspects of the task. The first aspect is the flagging performance of the spell-checker which indicates how many of the valid spelling errors it can detect. The second aspect is the suggestion accuracy which indicates if the suggestion provided by the tool is correct in the context. Hence we evaluate both these aspects using three different metrics. (i) Flagging Precision (P_f) which measures the accuracy of spelling error detection. (ii) Flagging Recall (R_f) which measures the coverage of the detection (how many errors are flagged by the spell-checker out of total errors) and (iii) Suggestion Accuracy (A_s) which indicate the proportion of correct suggestions provided by the spell-checker. Since the spell-checkers used in our experiments return multiple suggestions for each error detected, we only consider the first suggestion in our experiments. The metrics are formally defined as per equation (4.1).

$$P_f = \frac{f_t}{f_t+f_f} \qquad R_f = \frac{f_t}{f_a} \qquad A_s = \frac{s_c}{f_t} \tag{4.1}$$

where f_t represents the number of correct flagging, f_f indicates the number of incorrect flaggings or false positives, f_a indicates the number of actual spelling errors present in the testset and s_c represents the number of correct suggestions. Table 4.5 presents the precision and recall values for the different configurations of the spell-checkers used on the noisy testset. For the spell-checker evaluation, we just report the results on the noisy testset since it has a higher density of spelling errors (compared to generic testset) and has been designed specifically for evaluating spell-checking performance on translation.

Configuration	P_f (%)	R_f(%)	A_s(%)
AtD	64.37	67.40	91.63
Adapted AtD	92.42	95.6	92.46
Ada-AtD+Google	92.64	98.75	97.14

Table 4.5: Spell-checker Flagging Precision and Recall and Suggestion Accuracy on noisy testset

The figures in the table clearly show that while the off-the-shelf version of the AtD toolkit has a high suggestion accuracy, the flagging precision is considerably low, which is primarily due to the large number of false flags it generates. Adapting the spell-checker to the domain drastically improves the flagging precision and recall by reducing the number of false flags while slightly improving the accuracy. Finally combining the adapted AtD with Google spell-checker further improves the suggestion accuracy and flagging recall as additional errors missed by the adapted AtD toolkit are flagged and corrected by the Google spell-checker. The flagging precision however has a minor improvement in this case.

4.3.4 Experiments

In order to evaluate the effect of each of the normalisation techniques on the translation quality of the testset sentences, we start with a baseline SMT models trained on the unnormalised TM-based training data. The actual experiments were carried out in five different phases, each focussing on reducing one category of OOV words mentioned in Section 4.2. For the baseline model, the TM-based training data as well as the monolingual forum data are subjected to basic clean-up such as dropping empty lines and very long sentences (more than 100 tokens). This model is tested with both the testsets subjected to similar clean-up procedure as the training data to generate the baseline translation scores.

Subsequently both the testsets are then subjected to the following adaptations in a cumulative step-by-step manner:

1. Regex: Regular Expression based normalisation for the reduction of MASK OOVs.

2. Wrd-Split: Heuristic-based tokenisation for handling FW OOVs.

3. Spell-Chk: Off-the-shelf spell-checking for reducing SPERR.

4. Adapted-Spell-Chk (Ada SpChk): Spell-checking using domain adapted spell-checkers to reduce false positive flags.

5. Sup-data: Supplementary data selection and addition to enrich existing models to reduce VAL OOVs.

6. Regex+Sup-data: Data normalised by regular expression-based masking translated using models enhanced with selected supplementary datasets.

In the first step of experiments we perform regular expression-based masking on the entire TM-based training data, the monolingual forum data as well as on the source side of the two testsets. The regular expressions are developed on the basis of the MASK category tokens identified on the English forum data. These regular expressions are then used to mask all such tokens both in the training and test data. Since different URLs or Paths of Registry entries get normalised to similar place-holders both in the test and training data, the subsequent differences between the two is minimised. While masking the test data, a mapping is maintained between the place-holders and the original tokens which is later used to replace the place-holders in the translated sentences.

In the second step, the FW category tokens are targeted using the fused-word splitting scripts developed on the basis of FW category tokens identified on the English forum data. The scripts split up multiple words fused inadvertently to two or more separate valid words thereby improving the chances of them being translated properly. The scripts are designed to specifically avoid tokens like file names or numbers which naturally may have a period character within them. Since fused-words is a characteristic of the user-generated forum content, this normalisation is only applied on source side of the testsets.

The third and fourth steps of normalisation involves the use of spell-checking software to handle spelling errors. Automatic spell-checkers identify spelling-errors on the testsets and replace them with the first option from the suggestion list. This enables the most of the SPERR tokens to be mapped to valid words in the training data thus allowing their proper translation. If however a corrected SPERR token does not occur in the training data then we rely on the next adaptation step (supplementary data selection) to handle them. The spell-checkers are initially used in their standard configuration for the task. However the phenomenon of flagging valid domain-specific words as spelling errors prompt us to adapt the spell-checker to domain-specific data. This adaptation is guided by the performance of the spell-checkers on the SPERR tokens identified on the English forum data. We perform two separate set of experiments, one using unadapted spell-checker and the other using the adapted version of the same to observe their effect on translation quality. Similar to the Wrd-Split phase this normalisation is only performed on the source side of the testsets considering the fact that spelling-errors are characteristic of user-generated forum content.

The fifth phase involves enhancing baseline translation models with supplementary training material to allow translations of the VAL category tokens. The supplementary datasets are generated by querying the different parallel datasets described in Section 4.2.4 using a list of VAL category tokens generated from the English forum data. Once the parallel datasets are acquired, they are added to the in-domain TM-based training data and the translation and language models are re-estimated on the combined data. This allows a significant portion of the VAL tokens identified from the English forum data to be present in the enhanced phrase-tables thereby allowing their translations. The testset data is simply translated with this enhanced model to observe the effect of data selection on the translation quality.

While each the first five steps involves different normalisation techniques the final step (Regex+Sup) does not involve any specific normalisation, but is rather performed to investigate the effect of supplementary data selection on regex-based normalised testsets without any of the other normalisations. Considerable manual effort is required in the development and testing of each of the supplementary techniques. In comparison the supplementary data selection method is fully automatic and requires minimal manual effort. The objective of the last step of experiments is actually to compare the effect of supplementary data selection to normalisation on the translation quality of forum data with different degrees of noise.

4.4 Results and Analysis

Table 4.6 shows the different automatic evaluation metric scores for translations subject to each category of normalisation and supplementary data selection along with the percentage of OOV word reduction they result in, for both testsets and language pairs in our experiments. The last row (Regex+Sup-data) for each language pair in the table reports the results for translating only regular expression-based normalised testsets (without the other normalisations) using supplementary training data enhanced models. * denote statistically significant improvement of BLEU scores over the scores in previous row. Best scores in each set are in bold. The '+' sign in some of the normalisation columns indicate that the technique is used in addition to the techniques in previous rows.

	Normal-isation	Generic Test			Noisy Test		
		OOV	BLEU	METEOR	OOV	BLEU	METEOR
En–De	Baseline		25.98	43.91		21.32	38.25
	Regex	21.33	*26.53	44.09	10.10	21.63	38.50
	+ Wrd-Split	3.48	26.59	44.14	2.05	*21.68	38.55
	+ Spell-Chk	7.27	26.78	44.31	33.73	*22.50	38.88
	+ Ada-SpCk	4.58	26.92	45.01	18.66	*23.17	40.12
	+ Sup-data	12.16	***27.88**	**45.62**	11.99	***23.78**	**40.86**
	Regex+Sup-data	33.49	27.45	45.34	22.09	23.01	40.03
En–Fr	Baseline		34.14	52.34		30.27	47.37
	Regex	19.29	*34.80	52.78	9.83	30.65	47.96
	+ Wrd-Split	3.14	34.89	52.86	2.00	*30.77	47.89
	+ Spell-Chk	6.57	35.10	53.08	32.83	*31.60	48.79
	+ Ada-SpCk	4.14	35.33	53.22	18.17	*32.28	49.45
	+ Sup-data	12.71	***36.67**	**54.41**	12.17	***33.39**	**50.01**
	Regex+Sup-data	32.00	35.55	53.61	22.00	31.96	48.96

Table 4.6: Translation results using normalisation and supplementary data selection.

As the results in Table 4.6 show, regular expression-based normalisation results in a 0.55 absolute BLEU points improvement in En–De translations and a 0.66 absolute BLEU points improvement for En–Fr translations on generic testset. The METEOR score improvements for the generic testset are 0.18 absolute and 0.44 absolute points for En–De and En–Fr translations, respectively. For noisy testset, the improvements are 0.31 and 0.38 absolute BLEU points (0.25 and 0.59 METEOR points) for En–De and En–Fr, respectively. While the generic testset BLEU improvements are statistically significant at p=0.05 level using bootstrap resampling (Koehn, 2004), the noisy testset improvements are not statistically significant. The reason behind this may be attributed to the larger percentage of MASK tokens in the generic test compared to the noisy testset. The number of OOV tokens is reduced by 135 counts on the generic testset and 59 counts on the noisy testset. The improvements result from the fact that this normalisation helps to maintain intra-word ordering within MASK tokens and it does not translate the constituent words (since, the constituent words are replaced by the mask, they never pass through the SMT engine). The METEOR scores show similar trends of improvement across both testsets further confirming the translation quality improvement due to regex masking.

Using the fused word splitting technique on the regex-processed testsets, we observe minor improvements both for generic and noisy testsets over the previous normalisation scores, for both En–De and En–Fr translations. None of the improvements in this phase are statistically significant at the p=0.05 level. The reason for the marginal improvement becomes apparent observing the low percentage of OOV's (Table 4.6) reduced by this mechanism. This technique results in the reduction of the OOV count by 22 and 12 tokens for the generic and noisy testsets respectively.

As expected, handling the spelling errors using spell-checkers has a profound effect on the reduction of OOV words for the noisy testset with high density of spelling-errors. Using the adapted spell-checker on this testset, we achieve a total improvement of 1.49 absolute BLEU points for En–De and 1.51 absolute BLEU points for En–Fr translations. The METEOR score improvements are by 1.57 and 1.56 absolute points for En–De and En–Fr translations, respectively. This corresponds to a total reduction (combining reductions for unadapted and adapted spell-checking) of 316 OOVs for both En–De and En–Fr testsets. The overall improvement of using spell-checkers over the previous normalisation results are statistically significant at the p=0.05 level. However, for the generic testset, with spelling-error density reflecting that of average forum data, the improvements are much lower. While these improvements are not statistically significant they correspond to a reduction of 75 OOV tokens for both En–De and En–Fr testsets, respectively. The variation in the degree of improvement is found to be proportional to the quantity of SPERR tokens in the testsets, incorporating spelling corrections improves the quality of forum data translations in general. Hence this set of experiments allow us to achieve our additional objective of observing the effect of spelling correction on translation quality.

The fourth phase of experiments, where different supplementary parallel data resources are mined, results in further reduction of the OOV rates and improvements in translation quality. The data selection guided by VAL OOV tokens improves the scores by 0.96 and 1.34 absolute BLEU points (0.61 and 1.11 METEOR points) for En–De and En–Fr translations respectively on the generic testset. For the noisy testset the improvement figures are 0.61 and 1.11 absolute BLEU points (0.74 and 0.56 METEOR points) for En–De and En–Fr translations, respectively, over the previous normalisation results. All these improvements are statistically significant at the p=0.05 level. Furthermore, this technique further reduces the number of OOVs by 77 and 70 counts for the generic and noisy En–De testsets, respectively. The corresponding reductions on the En–Fr testsets are 89 and 73 counts for the generic and noisy testsets, respectively. Like in the case of the other normalisation techniques, the METEOR scores reflect the same trend of improvements across both language pairs and testsets. Studying the relative improvements between the generic and noisy testsets we observe that the improvements in the generic testset is higher than in the noisy testset for both language pairs. Hence clearly, supplementary data selection improves the generic testset more than it improves on the noisy testset. While one of the reasons for this variation is in the different percentages of VAL OOVs reduced in each testset, the other reason could be attributed to the amount of parallel (and monolingual) sentences added to the training data.

In Table 4.6, the translation quality scores in the 'Sup-data' row indicates the additive effect of the different normalisation techniques as well as supplementary data acquisition measures. Combining all the techniques results in statistically significant overall improvements of 1.90 and 2.53 absolute BLEU points (1.71 and 2.07 METEOR points) over the baseline scores on the generic testset, and 2.46 and 3.12 absolute BLEU points (2.61 and 2.64 METEOR points) on the noisy testset, for En–De and En–Fr translations respectively. Translating the regex-masked testsets (without word splitting and spell-checking) with the supplementary data enhanced models, we aim to assess the impact of the supplementary data selection technique in contrast to that of the normalisation methods. The 'Regex+Sup-data' row in Table 4.6 indicates the translation quality scores achieved for these experiments. Comparing the scores in the 'Regex+Sup' row to those in the 'Ada-Spck' row in the table, we observe that for generic testsets, supplementary data selection alone provides slightly better (0.53 absolute BLEU on En–De and 0.22 absolute BLEU for En–Fr) translation quality compared to that achieved by all normalisation methods. For the noisy testset however, the scores are lower than the adapted spell-checking scores by 0.16 and 0.32 absolute BLEU points for En–De and En–Fr respectively. These results clearly show that for general forum data (with average spelling-error density), fully automatic supplementary training data acquisition can perform as well and sometimes better than semi-automatic normalisation although they target different types of OOVs. However, for noisy data, normalisation complemented with supplementary data selection leads to much better translation scores than just using supplementary data selection.

4.5 Manual Evaluation

In addition to the automatic evaluation using evaluation metrics in Section 4.4, we further conduct a manual evaluation of the translated sentences to better understand the reasons for quality improvement. A number of translations from three systems: baseline, full normalisation (using all the normalisation techniques without supplementary data selection) and normalisation + supplementary data selection are compared manually by professional evaluators and scored. This section presents the details of our manual evaluation experiment.

4.5.1 Experimental Setup

A random selection of 100 translations (50 from generic and 50 from noisy for En–De and 48 from generic and 52 from noisy for En–Fr) from the three sources are used in the manual evaluation for both language pairs. We use three independent professional evaluators for each language pair, who are native speakers of German or French with good English skills. The evaluators were provided with the source, reference and hypothesis translations from the three systems, although they were agnostic of the actual systems (baseline, normalisation, normalisation+supplementary data selection)[19] producing the hypothesis translations. The evaluation is performed on the basis of two five-point scales representing *fluency* and *adequacy* (LDC, 2002). The adequacy scale conveys the amount of meaning conveyed in the hypothesis translation in comparison to the reference translation. The five-point fluency scale on the other hand, indicates the closeness of the translation to natural text in the target language (French or German). The details of the scale are presented in Table 4.7. In addition to marking each of the translations with a fluency and adequacy rating, the evaluators were further requested to provide a reason for the improvement or deterioration of the translation quality. The last column in the table 4.7 present the different reasons specified for our experimentation. The detailed guidelines for the manual evaluation experiment are presented in the appendix (cf. Appendix C).

4.5.2 Manual Evaluation Results

As previously mentioned in Chapter 3, we use Fleiss' Kappa measure (Fleiss, 1971) to assess the reliability of the the agreement between different evaluators. Table 4.8 reports the Fleiss Kappa values in Fluency ($Kappa_{Fl}$) and Adequacy ($Kappa_{Ad}$)

[19] although the system outputs were always presented in the same order

Fluency	Adequacy	Reasons
5=Flawless German/French	5=All	Better/Poor Translation of OOV words
4=Good German/French	4=Most	Better/Poor Word Order
3=Non-native German/French	3=Much	Better/Poor Lexical Selection
2=Disfluent German/French	2=Little	Other Reasons
1=Incomprehensible	1=None	

Table 4.7: Adequacy and Fluency Scales for Human Evaluation of MT

ratings for both the testsets and language pairs. The range of all the kappa values presented in the table are between the 'good' (0.41-0.6) and 'fair' (0.21-0.4) ranges (Landis and Koch, 1977) indicating a reasonable agreement between the three human evaluators for both the metrics (Fluency and Adequacy). As per the scores in Table 4.8 the agreement for fluency is higher than for adequacy, and the agreement for German is higher than that for French for most configurations. This therefore confirms the reliability of the manual evaluation experiment.

Lang. Pair	System	Test-1		Test-2	
		$Kappa_{Fl}$	$Kappa_{Ad}$	$Kappa_{Fl}$	$Kappa_{Ad}$
En–De	Baseline	0.65	0.48	0.63	0.33
	Norm	0.55	0.49	0.59	0.31
	Norm+Supp	0.56	0.39	0.54	0.24
En–Fr	Baseline	0.43	0.38	0.42	0.29
	Norm	0.33	0.31	0.46	0.37
	Norm+Supp	0.35	0.26	0.42	0.40

Table 4.8: Fleiss Kappa values for different datasets in Manual Evaluation

The average fluency (Avg-Fl) and adequacy (Avg- Ad) scores along with the corresponding BLEU scores of both the datasets and language pairs are reported in Table 4.9. The average fluency or adequacy ratings are computed by simply adding all the ratings from three different evaluators for individual system, testset and language pair and taking an arithmetic mean of the values.

Comparing the average adequacy and fluency ratings of the human evaluators for the three systems confirms our conclusion in Section 4.4 that normalisation improves translation quality over the baseline (by a narrow margin for En–De generic testset) and using supplementary data selection with normalisation improves the

Lang. Pair	**System**	**Generic testset**			**Noisy testset**		
		BLEU	**Avg-Fl**	**Avg-Ad**	**BLEU**	**Avg-Fl**	**Avg-Ad**
En–De	Baseline	20.07	2.28	2.81	19.68	1.86	2.29
	Norm	24.63	2.29	2.81	23.73	1.93	2.36
	Norm+Supp	28.51	2.51	3.00	26.06	1.99	2.48
En–Fr	Baseline	25.60	2.62	3.19	24.64	2.25	2.85
	Norm	27.93	2.69	3.23	27.75	2.43	3.03
	Norm+Supp	32.05	2.99	3.50	30.43	2.68	3.28

Table 4.9: BLEU scores, Average Fluency and Adequacy Ratings and Fleiss Kappa values for Manual Evaluation

results further. Normalisation (Norm) improves the average fluency by 0.43% (0.01 absolute) over the baseline for En–De translation on generic testset. The corresponding improvements for noisy testset translations are 3.63% (0.07 absolute) and 3.06% (0.07 absolute) for average fluency and adequacy, respectively. Furthermore, for En–Fr translations, we find normalisation improving average fluency by 2.67% (0.07 absolute) and 8.0% (0.18 absolute) for generic and noisy test translation ratings, respectively. In the case of average adequacy, the improvements are 1.25% (0.04 absolute) and 6.32% (0.18 absolute) for generic and noisy testsets, respectively. The improvements clearly show that the noisy test having a higher density of noise, gains much more from normalisation than the generic noise-density testset. Hence, this finding confirms our observation on automatic evaluation of translation scores, that normalisation is much more effective in improving translation quality for noisy datasets.

Comparing the average fluency ratings of supplementary data selection (Norm+Supp) over that of only normalisation (Norm) we observe an improvement of 9.61% relative (0.22 absolute) and 3.11% relative (0.06 absolute) on the generic and noisy En–De testsets, respectively. The improvements for the En–Fr generic and noisy testsets are 11.15% relative (0.3 absolute) and 10.29% relative (0.25 absolute), respectively. The relative adequacy ratings also follow a similar trend of improvement across both datasets. Clearly the improvements for the generic testsets are slightly higher than those for the noisy testsets for this phase. Hence the manual evaluation further confirms our findings that supplementary data selection has a more profound effect on generic testsets compared to their noisy counterparts.

Finally if we compare the average fluency ratings of Norm+Supp to that of the baseline, we observe relative improvements of 10.09% and 6.99% for generic and noisy testsets, respectively on the En–De data. For En–Fr, the relative improvements are 14.12% and 19.11% on generic and noisy testsets, respectively. Assuming the effect of normalisation and data selection to be additive, it is evident that using only data selection is almost as effective as using normalisation+data selection (comparing improvements of 9.61% to 10.09% for En–De and 11.15% to that of 14.12% for En–Fr) for the generic datasets. For noisy datasets however, normalisation improvements are significant and hence data selection alone can never match the improvements achieved by Norm+Supp (improvements of 6.99% compared to 3.11% for En–De and 19.11% to 10.25% for En–Fr datasets). This corroborates our final finding in Section 4.4, that supplementary data selection is often sufficient enough to provide necessary improvements for generic testsets.

While the overall human evaluation scores confirm the findings observed using the automatic evaluation metrics, we additionally want to have a deeper insight into the reasons behind the improvement produced by each technique. Hence as a part of the task , the evaluators were asked to mark the reasons for translation improvement or deterioration for the following two scenarios:

- S1: Comparing the translations produced by normalisation to that of the baseline system

- S2: Comparing the translations produced by normalisation+Supplementary Data Selection to that of only normalisation.

Analysing these reasons helped us identify how each technique affected translation quality. Table 4.10 reports the number of better or worse translations and their corresponding category-wise breakup for both the scenarios (S1 and S2), both testsets (generic and noisy) and both language pairs. Since we use three evaluators for the task, each set of sentence has three sets of scores (Fluency and Adequacy).

Lang. Pair	Status	Reason	S1				S2			
			Generic		Noisy		Generic		Noisy	
				Total		Total		Total		Total
En–De	Better	Better OOV Handling	2	12	3	17	1	21	1	22
		Better Word Order	2		2		1		2	
		Better Lexical Selection	12		15		20		22	
	Worse	Poor OOV Handling	1	16	0	6	1	6	0	8
		Poor Word Order	2		1		0		0	
		Poor Lexical Selection	14		6		8		8	
	Same			22		27		23		20
En–Fr	Better	Better OOV Handling	6	17	9	25	1	28	26	29
		Better Word Order	3		9		11		14	
		Better Lexical Selection	12		18		26		25	
	Worse	Poor OOV Handling	1	7	0	12	0	6	0	6
		Poor Word Order	2		3		1		2	
		Poor Lexical Selection	5		12		5		6	
	Same			24		15		14		17

Table 4.10: Categorical Distribution of Manual Analysis Observations

Hence the notion of ‘Better’ or ‘Worse’ translation is achieved by majority voting. A translation is considered to be better only when at least two out of three evaluators have provided higher ratings to it than its competitor. However, in particular cases where two evaluators have marked two translations to be of similar quality and the third one has marked one better with specific reasons, we consider the outcome of the third evaluator as final thereby considering it a better translation.

Observing the total number of ‘better’ or ‘worse’ translations for both the scenarios and datasets, we notice (unsurprisingly) that for the first scenario, which evaluates the effect of normalisation, the number of better translations are much higher for the noisy testset compared to the generic one. The overall number of better translations is higher in S2 compared to S1, which also confirms that both data selection and normalisation have an additive effect on translation quality. While S1 showcases the effect of normalisation only, S2 compares the effect of data selection plus normalisation to that of only normalisation, thereby attempting to isolate the effect of data selection only. Directly comparing the number of better German translation for the two scenarios, we find an improvement in 9 sentences for the generic test compared to 5 in the noisy testset. For the French translations again, the improvement figures are 11 and 4 for the generic and noisy testsets, respectively. This clearly shows that the effect of S2 on the generic test is bigger compared to its effect on the noisy testset. This re-affirms our conclusion that data selection alone works as well as normalisation + data selection for generic testsets.

Finally looking into the category-wise distribution of the improvements suggests that lexical selection is the most profound reason for improvement across both datasets and language pairs although the effect is much more profound in S2 compared to S1. For the German translations in S2 we find that supplementary data selection has not resulted in better OOV translation (we only find 1 case of better OOV handling compared to 26 in the French translations). This is in contrast to our findings from the automatic evaluation phase. The reason for this could be the misjudgement of the evaluators in identifying the improvements due to better OOV handling and better lexical selection. Overall, supplementary data selection in general drives better lexical selection in translations leading to improvement in overall quality.

4.6 Observation

During the automatic evaluation of our experiments as well as in the manual evaluation phase, we have observed the effect of different normalisation and supplementary data selection techniques on the translation quality of the forum content data with different degrees of noise. The results in Table 4.6 clearly indicate that normalisation techniques have an overall positive effect on the translation quality of forum data. We use three different kinds of normalisation techniques: regular-expression masking, fused-word splitting and spell-checking each addressing specific categories of OOV tokens, providing cumulative improvements in translation quality. In order to clearly illustrate how each techniques improve the translation quality we present a set of examples selected from the testsets (both generic and noisy), one for each normalisation technique aimed at a specific category of OOV tokens (c.f.

Section 4.2).

Src	*5 . click on the folder button and navigate to* ***c : \documents and settings \all users \application data*** *\and select the carbonite folder*
Ref	*5. klicken sie auf die ordnerschaltfläche und öffnen sie den ordner “* ***c : \documents and settings \all users \application data \carbonite*** *”*
Baseline	*5. klicken sie auf den ordner “ und navigieren sie zu* ***c : \dokumente und einstellungen \alle benutzer \anwendungsdaten \*** *und wählen sie die carbonite ordner*
Masked	*5. klicken sie auf die schaltfläche “ und wechseln sie zum ordner* ⟨**path_ph**⟩ *und wählen sie die carbonite ordners*
Regex	*5. klicken sie auf die schaltfläche “ und wechseln sie zum ordner* ***c : \documents and settings \all users \application data*** *und wählen sie die carbonite ordners*

Table 4.11: Regex Masking Example

Table 4.11 shows how regular-expression based masking helps better translate a sentence with MASK tokens. The token which is actually a Windows path entry is highlighted in the source sentence. The reference shows that the expected translation should not have the path elements translated in any manner. But the baseline translation model translates some parts of the path entry to their corresponding German translations. Using our regex-based masking technique we mask the path entry with a place-holder ⟨*path_ph*⟩. This place-holder is treated as a single token and is not broken up by the tokeniser. Finally after translation using the Regex-masked system, the place-holder is replaced with the original path token, thereby producing the final translation in row 5. While this example is based on a path type token, the same mechanism works for the other types of OOV tokens categorised as MASK (cf. Table 4.1).

Src	*re : nis09 did not detect 8 threats & 23 infected* ***objects.and*** *16 suspicious objects ?*
Ref	*re : nis09 n’ a pas détecté 8 menaces , 23* ***objets infectés et*** *16 objets suspects ?*
Baseline	*re : nis09 n’ a pas détecter 8 menaces et 23 infecté* ***objects.and*** *16 les objets ?*
Wrd-Split	*re : nis09 n’ a pas détecter 8 menaces et 23* ***infecté objets . et*** *16 les objets ?*

Table 4.12: Fused-word Splitting Example

In Table 4.12, we observe that the source sentence has a fused-word token *objects.and.* The baseline system has no entry for the fused-word in its phrase-table and hence considers it to be OOV and leaves the same untranslated. However, when the *Wrd-splitting* script is used on this sentence, the fused word is split into two valid entries *objects . and* which is easily translated by the same translation models as *objets . et.* This example clearly shows how wrd-splitting enables translation of the FW tokens by breaking them up at the punctuation marks.

Src	*and no for* ***somthing completly*** *different .*
Ref	*und nun zu* ***etwas völlig*** *anderem .*
Baseline	*und keine für* ***somthing completly*** *anders .*
Spck	*und nicht für* ***etwas völlig*** *anders .*

Table 4.13: Spell-Checker Example

Table 4.13 presents an example of a sentence with SPERR tokens as highlighted in the source sentence.[20] The misspelled words *somthing completly* do not occur in the phrase table of the baseline model and hence it is unable to translate them leaving them as is in the baseline translation. Automatic spelling correction allows us to change these tokens to valid words *something completely* which are present in the translation model. This provides their proper translations as is shown in the *Spck* row in the table. Clearly the *Spck* translation matches the reference translation better than that of the baseline translation thereby improving the translation quality.

Finally, Table 4.14 presents a source sentence with a VAL token. The word *disappointed* is not in the training data and hence the baseline system is unable to translate the same leaving it untranslated in the Baseline translation (row 3). Using supplementary data selection results in enhancing the baseline translation model thereby allowing the word and its translation to be added to the phrase table. Using this enhanced model to translate the sentence allows for a successful translation of the VAL token to its French counterpart, *déçu* as is highlighted in the last row of the table. This technique therefore widens the coverage of the baseline models to improve translation quality.

As is evident from the examples, the normalisation techniques discussed in this

[20] This sentence is selected from the noisy dataset

Src	*pretty **disappointed** with nis parental control not blocking websites on blocked list as well as through their category of websites to block .*
Ref	*je suis assez **déçu** que le contrôle parental de nis ne bloque pas les sites web figurant dans la liste bloqués aussi bien que ceux de la catégorie des sites web à bloquer .*
Baseline	*assez **disappointed** avec contrôle parental de nis pas le blocage de sites web sur liste bloqués ainsi que par l' intermédiaire de leur catégorie de sites web à bloquer .*
Sup	*assez **déçu** de contrôle parental de nis pas le blocage de sites web sur liste bloqués ainsi que dans leur catégorie de sites web à bloquer .*

Table 4.14: Supplementary Data Selection Example

chapter do work towards better translations for sentences with specific OOV types. However, the relative densities of each type leads to varied improvements in translation quality. Moreover, normalisation techniques are domain and content-specific and involve considerable manual effort in their development and design. On the other hand the supplementary data selection techniques discussed in Section 4.2.4 are fully automatic and require little manual intervention. Figure 4.2 shows the relative variations in translation quality (measured in terms of BLEU metric) between the baseline model, the model using all normalisation techniques cumulatively, the model using both normalisation and supplementary data selection to complement each other and the model using only supplementary data selection on both testsets and language pairs.

The results in Figure 4.2 clearly indicate that normalisation + Supplementary Data selection is the best configuration for both testsets and language pairs. However, comparing the translation quality provided by only normalisation techniques (Regex+Wrd-Split+Spell-Check) to that of only supplementary data selection (Regex+Supp), we observe that for generic testsets with lower degree of noise, supplementary data selection is slightly better than just using normalisation and slightly worse than normalisation + data selection. We have already observed from the scores in Table 4.6 that this variation is not statistically significant for the generic testset (Test 1). For Test-2 with higher degree of noise, normalisation clearly helps and provides better scores than using just supplementary data selection. For such testsets, using normalisation and data selection to complement each other leads to a

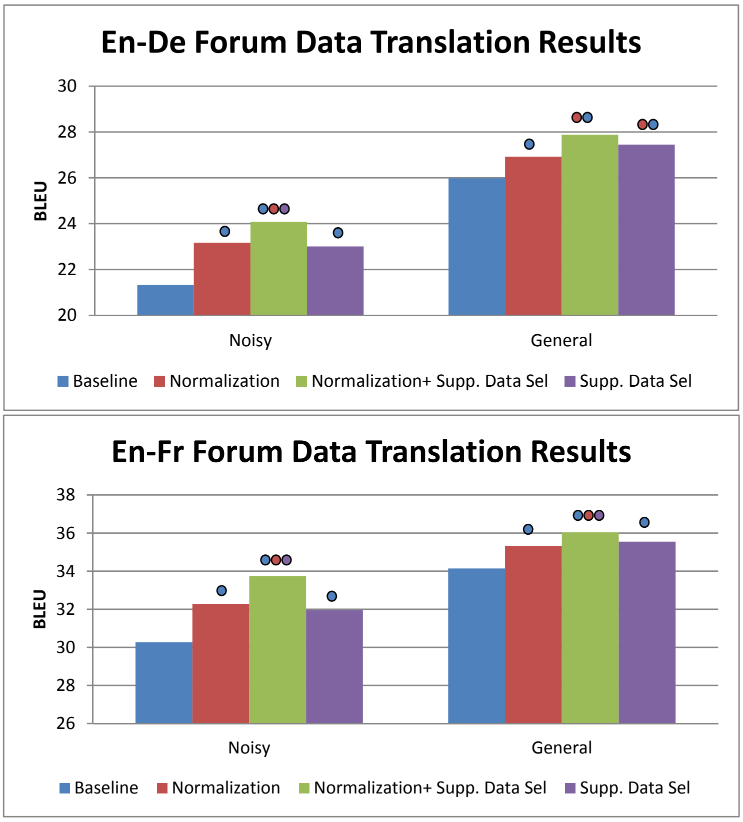


Figure 4.2: Comparing BLEU scores generated by baseline, only normalisation, normalisation and data selection and only data selection techniques

statistically significant improvement in translation quality. Considering the manual effort involved in normalisation, we observe that it is helpful only for noisy posts in the forum data. For comparatively less noisy posts, simple data selection works as well as combining normalisation with data selection.

4.7 Summary

Both normalisation techniques and supplementary data acquisition techniques used in our experiments show significant improvements in translation quality when applied to forum-content translation. We have observed the effect of these two categories of techniques in both additive and contrastive scenarios. Our findings based on both automatic and manual evaluation methods show that (i) normalisation and supplementary training material selection techniques can be complementary, (ii) for general forum data, fully automatic supplementary training data acquisition can perform as well or sometimes better than semi-automatic normalisation (although tackling different types of OOVs) and (iii) for noisy data, normalisation is really effective and cannot directly be matched only by supplementary data selection.

In addition to investigating the effect of normalisation on user-generated forum content, the experimental findings in this chapter clearly show that normalisation techniques are effective adaptation tools in a scenario where in-domain training data is unavailable and related-domain data needs to used to train models. Finally we present the second research question (**RQ2**) introduced in Chapter 1 which formed the actual motivation of the experiments in this chapter:

> **RQ2:***In a scenario where the target domain is different from the training domain, how effective are normalisation and data selection methods towards the improvement of translation quality?*

Considering the summary of the observations presented previously, our experiments in this chapter have succeeded in providing a conclusive answer to **RQ2**.

4.7.1 Contributions

The main contributions of this chapter are as follows:

- We have successfully used different normalisation techniques to adapt translation models trained on enterprise content to translate user-generated forum data

- We have shown the effect of data selection and normalisation for additive as well as contrastive scenarios, on forum-content data with different degrees of noise. For generic forum content, our findings have shown that supplementary data selection techniques work nearly as well as normalisation and data selection methods, while for noisy content normalisation is really effective.

- We have successfully adapted an off-the-shelf spell-checker for the technical domain and applied its results to improve translation quality of forum data.

- We have presented a supplementary data selection method guided by a list of OOV words and have shown it to effectively improve system performance.

Considering the success of data selection techniques, in the next chapter we focus our efforts on combining supplementary and in-domain data at the various component levels of an SMT system. We use a mixture modelling framework to combine multiple datasets both for translation and language models and observe their comparative performance on the translation of forum content data.

Chapter 5

Domain Adaptation by Component-level Mixture Modelling

Chapter 4 introduced the adaptation challenges involved in using SMT systems trained on corporate TM-based data for user-generated web forum data translation. The differences between the training data (corporate TMs) and the target domain (forum data) were quantified in terms of the number of out-of-vocabulary (OOV) tokens. The OOVs were broadly categorised into different categories depending on their characteristics and different normalisation and data selection techniques were developed for each OOV category. Using both the techniques additively had an overall positive impact on the translation quality. Moreover, comparing the exclusive effects of normalisation and data selection in our experiments revealed that the data selection was nearly as effective, and sometimes even better than normalisation for the given task. The success of data selection as an adaptation approach in the current setting prompts us to investigate deeper into the use of this technique.

The general approach of adaptation by supplementary data selection has two main aspects: (i) the criteria used to select data from an out-of-domain or a related domain data source and (ii) the method of combining the selected data to the existing in-domain data. In the previous chapter, we used a particular category of OOVs as a criterion for supplementary data selection and used simple concatenation of the selected data and in-domain data as the combination method. We also observed the effect of an overall system-level combination of SMT systems in the experiments presented in Chapter 3. Consequently, in this chapter we investigate the effect of this combination at a more fine-grained level. In contrast to combination at the system level or at the data level, we combine in-domain and out-of-domain selected data at the individual component-level of a standard SMT setup. Using the mixture model adaptation framework (Hastie et al., 2001), we combine individual translation models as well as language models (LMs) trained on in-domain and out-of-domain datasets to create a combined model SMT system.

The scenario for the adaptation experiments presented in this chapter are exactly the same as in Chapter 4. We aim to translate user-generated content from the Symantec web-forums using translation models and LMs trained on Symantec TM data. The corresponding out-of-domain models are trained on supplementary data selected from three different freely available parallel data resources. In contrast to the OOV-guided data selection method of the previous chapter, a perplexity-based ranking and thresholding technique is used for the data selection. We use two different variants of mixture adaptation: linear mixture adaptation and log-linear mixture adaptation to combine the in-domain and out-domain component models in our experiments. These two variants are used to separately adapt both the translation model and language model components of the respective in-domain and out-of-domain SMT systems. Different sets of experiments are conducted to compare the effect of both translation and language model adaptation as well as linear and log-linear adaptation on the translation quality of forum data. Furthermore, in order to compare the effect of mixture adaptation to that of data concatenation, we present a set of experiments where a single model is trained on a concatenation of the in-domain and out-of-domain corpora. All the experiments are conducted in the English-to-German (En–De) and English-to-French (En–Fr) language directions. Experimental results reveal that for the current task, LM adaptation is more effective than translation model adaptation across both language pairs and datasets. Additionally, linear interpolation of both translation models and LMs performs better than both the log-linear as well as the concatenated data setting.

The remainder of the chapter is organised as follows: Section 5.1 presents the motivation for our work and a brief background. Section 5.2 details the concatenation, linear and log-linear interpolation methods used in the experiments. Section 5.3 describes the datasets used along with an account of the different experiments conducted. Section 5.4 presents the results and analysis of the different experiments, followed by general observations in Section 5.5. The chapter concludes with a summary of the findings in Section 5.6.

5.1 Motivation

The experimental findings in Chapter 4 have clearly shown (i) that using SMT models trained on TM-based content to translate forum data requires domain adaptation in some form, and (ii) that data selection from supplementary parallel data sources is particularly useful in this regard. However, the data selected from supplementary sources need to be combined with the in-domain data in an effective way to extract the maximum advantage out of them. This forms the primary hypothesis for our experiments reported in this chapter. In addition to the data-level combination tried out in Chapter 4, we investigate the effects of model- or component-level combination between in- and out-of-domain data in our forum data translation setting. The requirement for model combination at the different component levels of an SMT system motivated us to use the widely adopted mixture models as the combination framework in our experiments. We compare the effects of linear and log-linear interpolation methods in combining models in addition to the straightforward method of data concatenation. Additionally, comparing the effect of LM adaptation to that of translation model adaptation in the current setting also forms a secondary motivation for the experiments reported in this chapter.

Data selection is a particularly popular domain adaptation technique in SMT, and different approaches to data or model combination have been proposed in the literature (Hasan and Ney, 2005; Foster and Kuhn, 2007; Yamamoto and Sumita, 2008; Lavergne et al., 2011; Sennrich, 2012a). The experiments reported in this chapter are broadly based on those of Foster and Kuhn (2007), using the same linear and log-linear mixture modelling to combine translation and language models. However, the primary difference lies in the nature of the test and development sets used in the experiments. While Foster and Kuhn (2007) use development and testsets from a mixture of the different training domains, in our case the corresponding datasets come from the target domain (user forums), which is quite different from the training data. Compared to the distance-based mixture weight estimation used in Foster and Kuhn (2007), we use expectation maximisation (EM) (Dempster et al., 1977) on LMs built on the target side of the training data (in-domain and out-of-domain) to estimate the mixture weights with respect to the development set. Furthermore, the mixture model experiments– especially the log-linear interpolation experiments reported by Foster and Kuhn (2007)– were carried out on the Portage phrase-based SMT system (Ueffing et al., 2007b), whereas we use Moses (Koehn et al., 2007) to carry out our experiments.

5.2 Approach

Faced with the task of adapting an SMT system trained on Symantec TM data to better translate forum content, we present a mixture model-based adaptation approach to address the domain adaptation task. This approach can be summarised using the following general steps:

1. Select a relevant portion of the supplementary parallel training data.

2. Train different component models (both translation and language models) on in-domain as well as selected supplementary datasets.

3. Weight each component model according to its fit with the target domain of the task (user forum content).

4. Combine the weighted component models into a single global model and use it to translate forum content.

In the following sections, we elaborate on each aspect of the steps in more detail.

5.2.1 Relevant Supplementary Data Selection

While additional training data is known to boost the performance of an SMT system (e.g. Sennrich (2012b)), adding a huge amount of out-of-domain data (with respect to the in-domain data size) may not always be the best choice for a domain-specific system (Axelrod et al., 2011). Hence relevant data selection is an important aspect in all data selection-based approaches to domain adaptation and forms the first step in our adaptation process. In the current setting, in order to select relevant data, each sentence pair in the supplementary dataset is ranked according to their 'closeness' to the target domain. As a measure of closeness, we use the *perplexity* of the source sentence with respect to an LM in the target domain (cf. Hildebrand et al. (2005)). The perplexity of a sentence s with empirical n-gram distribution p given a language model q is represented as in (5.1):

$$PP(s|p,q) = 2^{-\sum_x p(x) log q(x)} = 2^{H(p,q)} \tag{5.1}$$

where $H(p,q)$ is known as the *cross-entropy* between p and q. A sentence with a low perplexity value implies low cross entropy between the language model distribution and the empirical n-gram distribution of the sentence, thereby suggesting the closeness of the sentence to the LM. Therefore, sorting each sentence-pair in terms of its perplexity value with respect to a forum data LM implies that the sentences closest to the target domain appear at the top. Finally an empirical threshold value is used to select only a section of the closest sentence pairs for adaptation. While this technique allows the selection of parallel dataset for translation model adaptation, the target side of the selected supplementary dataset is also used for LM adaptation.

5.2.2 Component Models

As previously stated, we use mixture adaptation to separately adapt both translation and language models in our experiments. Hence, individual models are created on the Symantec TM in-domain training data and the selected supplementary training data prior to the combination. Using the standard phrase-based SMT (PBSMT) approach (Koehn et al., 2003), source–target phrase pairs consistent with the word alignment information are extracted from the sentence-aligned parallel training data. The extracted source–target phrase pairs along with their feature values are stored in a phrase table which constitutes the translation model in a standard PBSMT setup (cf. Chapter 2). All these features along with the features from a reordering model (used to model the reordering of the phrases in the target sentences) and LM probabilities are combined in a log-linear framework in order to assign a score to each translation option during decoding. Finally the best translation option (the one with the highest score) is chosen as the translation for an input sentence. Formally this task can be expressed as in (5.2):

$$\hat{e} = \arg\max_{e} \sum_{i=1}^{K} \lambda_i h_i(f, e) \tag{5.2}$$

where, $h_i(f, e)$ denotes the features from different components used in translating the source sentence f into the target sentence e. K is the number of features used and λ_i are the corresponding weights of the features. These feature weights (λ_i) were estimated using a discriminative training method known as Minimum Error Rate Training (MERT) (Och, 2003), on a held-out development dataset (devset).

The in-domain LMs are created using a combination of actual forum content and the target side of the TM data. In contrast the target side of the supplementary data is used to train the out-of-domain LMs. While in translation model adaptation we usually combine two models (one in-domain and the other out-of-domain), for LMs we usually use a combination of three models: a target domain model (forums), an in-domain model (the target side of the Symantec TM) and an out-of-domain model (the target side of the supplementary data).

5.2.3 Linear and Log-linear Interpolation of Models

In order to combine the individual component models, we used two combination frameworks in mixture modelling: linear mixtures and log-linear mixtures. Individual translation or language models are linearly interpolated using the formula in (5.3):

$$p(x|h) = \sum_{s} \lambda_s p_s(x|h) \tag{5.3}$$

where $p(x|h)$ is one of the features in the LM or the translation model, $p_s(x|h)$ is the same feature trained on the individual training resource s, and λ_s is the corresponding weight of the particular resource, all of which sum up to 1. These mixture weights are only used to combine the feature values from individual resources into a single combined feature value and do not directly participate in the global log-linear combination represented by equation (5.2).

As an alternative to linear interpolation, log-linear interpolation provides model combination in the form of a global model using the formula in (5.4):

$$p(x|h) = \prod_{s} p_s(x|h)^{\alpha_s} \tag{5.4}$$

where α_s is the log-linear mixture weight for the feature $p_s(x|h)$ which is a part of a model (translation or language model) trained on the training resource s (in-domain or out-of-domain datasets). In contrast to the linear mixture weights, the log-linear mixture weights are global weights just like the other feature weights specified in equation (5.2). This difference between the linear and log-linear interpolation methods provides a distinct advantage to the latter. Using the MERT algorithm, the log-linear mixture weights can be directly optimised to maximise translation quality as measured in terms of automatic evaluation metrics. In contrast, the linear mixture weights can not directly be optimised using MERT (since it assumes only a 'flat' log-linear model). Hence setting the linear mixture weights requires optimisation with respect to monolingual metrics of closeness (perplexity in our case) to the target domain.

5.2.4 Learning Mixture Weights

Since MERT cannot be used to set and optimise the linear mixture weights, we use a work-around for setting them in the current context. In order to set the mixture weights for linear interpolation of LMs, we use the EM algorithm to optimise the maximum likelihood of the language models with respect to the target side of the development set. Initially all models are uniformly weighted and the EM algorithm iteratively optimises the weights until a predefined convergence criterion is met. In order to estimate mixture weights for the translation models, individual LMs are created on the target side of the parallel training data for both in-domain and out-of-domain datasets. The mixture weights for these LMs are then estimated using the same EM-based setup used to estimate the LM mixture weights, against the target side of the development set. Finally these weights are used to combine the feature values in the respective phrase tables to generate a linearly interpolated translation model.

As previously stated, the log-linear mixture weights are estimated by directly using multiple component models (from each of the resources) in the standard PB-SMT setup and running MERT to maximise translation performance in terms of BLEU automatic evaluation metric (Papineni et al., 2002) on the held-out development set. In contrast to our method of setting the linear mixture weights this technique has an added advantage in the fact that the weights are optimised not in terms of fitness to the target domain, but directly in terms of translation quality. However, using multiple phrase tables and LMs greatly increases the number of features to be optimised, thus hindering MERT's ability of converging on an optimal set of weights (Chiang et al., 2009). In order to address the problem of sub-optimal weight setting by MERT, we re-run our log-linear mixture experiments using the MIRA algorithm (Crammer et al., 2006) as an alternative to MERT. The details of the MIRA experiment results along with their comparison to MERT are presented in Section 5.4.

As already mentioned in Chapter 2 (cf. Section 24), we use the IRSTLM (Federico et al., 2008) toolkit for all our LM training purposes. Specialised scripts associated with this toolkit are used to compute sentence-level perplexity scores on the forum data LM as well as estimating linear mixture weights using EM on the devset. However, in order to save the linearly interpolated LMs into a single combined LM, we use the weighted model mixing mechanism in the SRILM toolkit (Stolcke, 2002), as IRSTLM does not provide this particular feature.

5.3 Experimental Setup

In this section we present the details of the different datasets used in our experiments along with the tools and techniques used to create the translation and language models. We also describe the different set of experiments performed to evaluate the effect of different combination strategies and component adaptation on translation quality.

5.3.1 Datasets

The in-domain training data for our SMT models consist of En–De and En–Fr bilingual datasets in the form of Symantec TMs. In addition to the bilingual datasets, we also have small monolingual collections of actual forum posts in both German and French. Despite being from the actual target domain, these datasets are monolingual and hence only useful for language modelling. However, the small size of the German and French forum data prompts us to use them in combination with the target side of the TM-based training data for training the in-domain LMs. We also have a large collection of posts from the original Symantec English forums acquired over a period of two years which we use to create the LM on the basis of which the supplementary datasets are ranked. In addition to the in-domain training data, we use the following three freely available parallel corpora as the source of supplementary datasets in the current set of experiments:

- Europarl (Koehn, 2005) (EP): Parallel corpus comprising the proceedings of the European Parliament.
- News Commentary Corpus (NC): Released as a part of the WMT 2011 Translation Task.[1]
- OpenSubtitles2011 Corpus (OPS):[2] A collection of documents released as part of OPUS.

Table 5.1 reports the number of sentences in all the datasets across both language pairs along with the average sentence length (ASL) in the source and target corpora for all datasets. Note that out of the different supplementary datasets mentioned in Chapter 4, we only use the ones which report highest OOV coverage (cf. Page 106).

The development (dev) and the testsets reported in the Table 5.1, are essentially derived from the generic noise density-based testset used in Chapter 4 (Section 4.3.1). This dataset comprises sentences randomly selected from the English forum data and their corresponding translations generated by professional translators. Following the experiments in Chapter 4, this dataset was subjected to basic cleanup using some of the normalisation techniques presented in Chapter 4 followed by a manual review.[3] Finally this dataset is randomly split into dev and testset sentences for the experiments in this chapter.

[1] http://www.statmt.org/wmt11/translation-task.html

[2] http://www.opensubtitles.org/

[3] This cleanup explains the difference in sentence counts between generic testset in Chapter 4 (2022) and the number of sentences in the dataset presented in this chapter (1000+1031 = 2031)

	dataset	En–De			En–Fr		
		Sent. Count	**En ASL**	**De ASL**	**Sent. Count**	**En ASL**	**Fr ASL**
Bi-text	Training	832,723	12.86	12.99	702,267	12.42	14.86
	devset	1,000	12.91	12.20	1,000	12.91	14.99
	testset	1,031	12.75	11.99	1,031	12.75	14.69
Supp. Data	EP	1,721,980	27.48	26.11	1,809,563	27.34	30.35
	NC	135,758	24.34	24.98	115,085	24.79	29.06
	OPS	4,649,247	7.61	7.16	12,483,718	8.61	8.17
Mono-lingual	**Forum Data**	**Sent. Count**			**ASL**		
	English	1,129,749			12.48		
	German	42,521			11.78		
	French	41,283			14.82		

Table 5.1: Number of sentences and average sentence length for in-domain, supplementary data and monolingual forum datasets

5.3.2 Pre-Processing and Post-Processing

Before training, all the bilingual and monolingual data are subjected to tokenisation and lowercasing using the standard Moses pre-processing scripts. We also use the regular expression-based masking technique presented in Chapter 4 for masking URLs, path entries, registry entries, dates and IP addresses as a part of the pre-processing step primarily on the forum content and the source side of the dev and testsets. Since these tokens are replaced with unique place-holders with the objective of reducing a multi-word token into a single one, the standard Moses tokeniser is also modified to ensure that it does not tokenise the place-holder tokens. The pre-processing step also creates the map file for storing the mapping between actual tokens and place-holders which is utilised in the post-processing step to replace the masks in the same monotonic order in which they appeared in the source side of the testsets.

5.3.3 Data Selection

Prior to conducting the mixture model experiments, we have to select relevant parts of the supplementary data for adaptation. Section 5.2.1 explains our approach based on ranking individual sentence pairs of the supplementary datasets using perplexity on a target domain LM. In order to rank the sentence pairs, the perplexity of every source sentence in the supplementary dataset is computed against a LM trained on the monolingual English forum data. Eventually the sentences are sorted according to their perplexity values to place the sentences 'closest' to the target domain at the top of the ranking. However, selecting only a relevant portion from such a ranked list requires the selection of a threshold perplexity value under which all sentences could be considered relevant to our target context. In order to maintain a balance (in terms of number of sentences) between the amount of in-domain data and the selected supplementary datasets, we chose a perplexity threshold value such that the number of selected sentences was limited by the size of the in-domain data (Symantec TMs). Since the number of sentences in the EP and OPS corpus is much greater than that of the Symantec TMs (cf. Table 5.1), the thresholding technique is applied only to them. In contrast, the entire NC corpora is used for adaptation in our experiments. In addition to individually using each of the supplementary datasets for adaptation, we also used a combination (CMB) of all three selected datasets for both translation model and LM adaptation. Table 5.2 presents the threshold values, number of selected sentences and the ASL for all four resources.

Data	**En–De**				**En–Fr**			
	Thr.	**Sent. Count**	**En ASL**	**De ASL**	**Thr.**	**Sent. Count**	**En A.S.L**	**Fr ASL**
EP	8.8	832,651	32.55	30.13	7.2	702,171	28.16	30.96
NC		135,758	24.34	24.98		115,085	24.79	29.06
OPS	4.6	832,704	8.12	7.32	2.4	702,262	9.45	8.47
CMB		1,801,113	20.63	19.19		1,519,618	20.42	19.26

Table 5.2: Threshold value, number of sentences and average sentence length of selected supplementary data

5.3.4 Unadapted Baseline Model

The baseline model used in our experiments is a standard Moses-based SMT system trained only on the in-domain datasets we had at our disposal. The translation model was trained on the bilingual Symantec TM data, while the LM was estimated on the concatenation of monolingual forum data and the target side of the bilingual TM data. Considering our objective of observing the effect of model adaptation on forum data translation, the baseline model is deliberately kept free of any adaptation using any of the selected supplementary datasets.

5.3.5 Language Model Adaptation

For LM adaptation in the current scenario, there are three different sources of training data: the monolingual forum data, the target side of the bilingual TM data and the target side of the selected supplementary data. Hence for the adaptation experiments, we use the following three configurations:

1. Conc: An LM trained on the monolingual data generated by simple concatenation of all the three different sources of training data.

2. Linmix: Individual models trained on each of the resources combined using linear mixture modelling.

3. Logmix: Individual models trained on each resource combined using log-linear mixture modelling.

Training the Conc model involves the simple routine of concatenating all three sources of data and training a single LM on it. Using this configuration on the CMB dataset, we concatenate the data from all three supplementary resources along with the forum data and target side of the TM data and train a single model on the same.

In the Linmix configuration, we use the linear interpolation weights estimated using the technique described in Section 5.2.4 to combine individual LMs trained on each of the independent data sources. Hence for each of the EP, OPS and NC datasets, we perform a linear interpolation of three LMs trained on forum data, target side of TM data and target side of selected supplementary data, respectively. However, for the CMB dataset, we perform a linear interpolation between five different LMs comprising the forum model, the TM model and individual model based on the EP, NC and OPS datasets.

For the Logmix configuration we use the individual LMs trained on each of the in-domain and the supplementary data sources directly in the configuration file of the Moses SMT decoder. When provided with multiple LMs, the Moses decoder treats the probabilities from each of the LMs as individual feature values which are eventually combined using the log-linear combination presented in equation (5.2). As previously stated, the log-linear mixture weights for each of the LMs are estimated by running MERT on the devsets.

5.3.6 Translation Model Adaptation

Similar to LM adaptation, our translation model adaptation experiments also use the same three configurations (Conc, Linmix and Logmix) for combining the in-domain and out-of-domain phrase tables. In the Conc configuration, the TM-based in-domain training data is simply concatenated to the selected out-of-domain supplementary data, and a single phrase table is estimated on the combined data. Using the Conc configuration, the combination happens at the data level and thus is fairly simple to implement. In contrast, the Linmix or the Logmix configurations attempt the combination at the model level, thereby making the implementation somewhat more complex.

Mixture adaptation of the translation model aims at combining the phrase tables generated from the in-domain and the supplementary datasets. Since we use Moses to train our translation models, the phrase tables generated contain the following 5 feature values:

1. Inverse phrase translation probability: $\phi(s|t)$

2. Inverse lexical weight: $lex(s|t)$

3. Direct phrase translation probabilities: $\phi(t|s)$

4. Direct lexical weight: $lex(t|s)$

5. Phrase penalty: (always $exp(1) = 2.718$)

Linear mixture adaptation of phrase tables in the current context thus involves linear interpolation of each of these feature values from the in-domain and the supplementary phrase tables using formula (5.3) to generate a combined phrase table. However, since the two phrase tables are independently estimated on different datasets, there are only a few phrase pairs common to both tables. For the phrase pairs which do not occur in all the phrase tables (involved in the mixture), the linear interpolation implementation in equation 5.3 assumes feature values of 0 in the non-occurring phrase table. Strictly speaking, for phrase pairs occurring in a single table should ideally render the phrase translation probabilities undefined. For example, if we are trying to merge $\phi(t|s)$ and the source phrase s is missing from one phrase table then ideally the direct phrase translation probability is undefined (instead of 0). To avoid this deficiency we implemented a weight re-normalisation technique along the lines of Sennrich (2012b) for our linear interpolation experiments. However, the re-normalisation method is not strongly motivated and does not significantly affect the eventual BLEU scores (Sennrich, 2012b). Hence, it could safely be ignored in future implementations of linear interpolation.

Using the log-linear mixture adaptation (Logmix) on phrase tables poses a slightly different problem. As stated in Section 5.2.4, we utilise the Moses decoder's ability to accommodate multiple phrase tables using the multiple decoding path functionality (Koehn and Schroeder, 2007) for log-linear mixture adaptation. However, the Moses decoder allows different configurations for handling multiple phrase tables:

- **Both**: In this configuration, all constituent phrase tables are used to score a particular translation option by combining the translation options using a weighted log-linear combination. However, this requires all the constituent phrase tables to have the same phrase pairs. If a phrase pair is not contained in one of the tables, it is ignored for both scoring and decoding.

- **Either**: This configuration allows a translation option to be scored by any one of the constituent phrase tables. For a phrase-pair common to multiple phrase tables, separate translation options are created for each occurrence, but with different scores.

Clearly in order to achieve actual log-linear mixture adaptation, we have to use the *both* configuration in the current setting. However, since the phrase tables are trained on different data sources, a large majority of the phrase pairs are not shared between the phrase tables. In order to overcome this issue, we copy such phrase pairs to all the constituent phrase tables while re-distributing their probability masses uniformly between the phrase pair in every phrase table. Hence when combining only a single supplementary data source (EP, NC or OPS) with the in-domain phrase table, for each non-shared phrase pair its probability mass is halved in each of the tables. In the case where we use a combination of all the three supplementary sources, the probability masses are quartered. It is to be noted that this operation is only carried out on the Inverse and Direct phrase translation probabilities. The lexical weights, not being true probabilities,[4] are exempted from this operation and are copied 'as is' to the different constituent phrase tables. This process forces each phrase pair to be present in all the constituent phrase tables, thus allowing us to use the *both* configuration for log-linear mixture adaptation.

5.3.7 Experiments

The primary motivation for the experiments in this chapter is to investigate the effect of mixture adaptation on the different component levels of an SMT system. In addition, we also want to identify the individual effects of both translation and LM adaptation on the translation quality of the forum data. Sections 5.3.5 and 5.3.6 present the three different adaptation configurations (Conc, Linmix, Logmix) we use for our experiments for LM and translation model adaptation, respectively. Hence we broadly divide our experiments into three different phases (Phase-1, 2 and 3) each using one (out of the three) particular adaptation setting for translation model

[4] Lexical weights are computed as an average of word translation probabilities considering the most popular alignment.

adaptation. Within each phase, we further use three different adaptation settings for the LM adaptation. Hence for each supplementary dataset under consideration, we perform nine different sets of experiments for every possible combination of translation model and language model adaptation settings. Figure 5.1 depicts the different phases and scenario for our experiments.

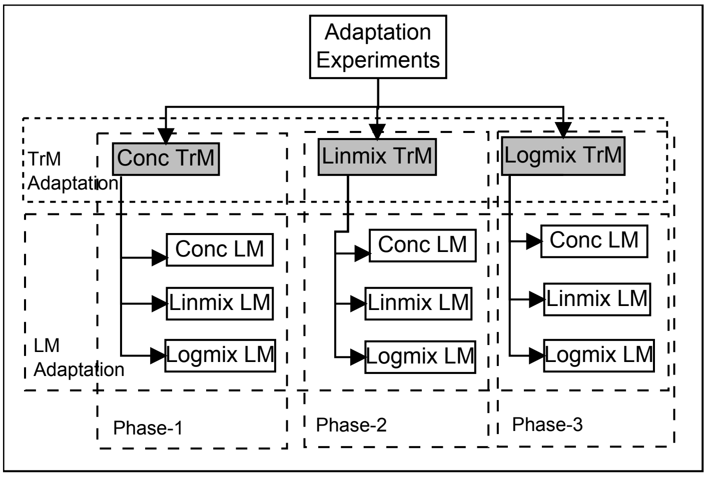


Figure 5.1: Translation (TrM) and Language Model (LM) Mixture Adaptation Experiments

After combining the adapted translation and language models, each setup is tuned using MERT on the devset and used to translate the testset across both language pairs.

5.4 Results and Analysis

In this section, we present the results of our adaptation experiments in four different tables, one for each of the supplementary datasets under consideration. Table 5.3 presents the translation quality metric scores for each of the nine different adaptation settings using EP as the supplementary data source. The first row indicates the unadapted baseline scores, while the best scores for each phase (i.e. for each translation model adaptation setting) are in bold. Statistical significance of the BLEU scores at the p=0.05 level, computed using the bootstrap resampling method (Koehn, 2004), are also marked in the table with $*, \dagger, \ddagger$ indicating statistically significant improvement over the baseline, concatenated language models and log-linear mixture of language models, respectively. In addition to the nine adaptation configurations, we present an additional set of LogMix experiments tuned using the MIRA algorithm (Crammer et al., 2006) to address the issue of sub-optimal weight estimation by MERT in large parameter settings. The last three rows in Table 5.3 present the scores obtained on the MIRA tuned LogMix TrM setting.

Adapt. Setting		**En–De**		**En–Fr**	
TrM	**LM**	**BLEU**	**METEOR**	**BLEU**	**METEOR**
Baseline		21.80	39.78	31.65	50.83
Conc	Conc	∗‡22.88	40.86	∗32.63	51.68
	Linmix	∗‡**23.03**	**41.21**	∗†‡**33.24**	**52.38**
	Logmix	22.10	39.85	∗32.40	51.42
Linmix	Conc	∗‡22.90	40.93	∗32.73	51.60
	Linmix	∗‡**23.30**	**41.25**	∗†‡**33.33**	**52.03**
	Logmix	∗22.32	39.96	∗32.47	51.18
Logmix	Conc	22.23	40.07	∗‡32.21	51.17
	Linmix	∗†‡**22.83**	**41.09**	∗†‡**33.04**	**52.13**
	Logmix	21.98	39.60	31.03	49.43
MIRA	Conc	22.74	40.66	31.91	50.93
	Linmix	∗‡**22.78**	**40.91**	∗†‡**32.69**	**51.83**
	Logmix	22.14	39.84	31.56	49.94

Table 5.3: Mixture Adaptation Results using EP as the supplementary data source.

The results in Table 5.3 clearly show that augmenting the in-domain dataset with the selected supplementary data improves the translation quality over the baseline model. In the first phase of experiments, using Conc adaptation on translation models, we observe that all three LM adaptation settings provide statistically significant improvements over the baseline scores. Using a Conc LM results in improvements of 1.08 and 0.98 absolute BLEU points (1.08 and 0.85 METEOR points) for the En–De and En–Fr testsets, respectively. Using a Linmix LM adaptation setting we observe even better improvements of 1.23 and 1.59 absolute BLEU points (1.43 and 1.55 METEOR points) for the En–De and En–Fr testsets, respectively, over the baseline translation scores. The Logmix adaptation of LMs also improves translation quality over the baseline, although the degree of improvement is lower compared to Conc

LM or Linmix LM. Both the Linmix and Conc improvements over the baseline are statistically significant at p=0.05 level using bootstrap resampling (Koehn, 2004). Comparing the Conc, Linmix and Logmix scores reveals that Linmix is the best-performing system, outperforming the other two systems and the Logmix scores are the worst among the three settings. The METEOR scores reveal the same trend as the BLEU scores, with Linmix LM adaptation providing the best scores in every phase.

Since the Logmix adaptation setting combines language model probabilities using a weighted product model (cf. 5.4), a phrase having a low LM probability negatively affects its overall translation score in the decoding process, thereby causing the decoder to prefer other phrase pairs over it. This is particularly a problem for those target phrases which occur in only one of the LMs. Considering the different sources our constituent LMs are trained on, such phrases are a majority in our experiments thereby leading to poor performance of Logmix LMs. In contrast, using the Linmix adaptation computes a weighted average of the phrase probabilities from individual language models and hence is less susceptible to outliers and data sparseness. In contrast, the Conc model trains a single language model on the concatenated data and hence is free from the effect of data sparseness, but still suffers the ill effects of statistical outliers. This explains the better performance of Linmix adaptation over the other two models. Furthermore, the better performance of Linmix over Logmix LM adaptation settings seems to confirm the findings reported in Foster and Kuhn (2007) and Lavergne et al. (2011).

Observing the results of the second and third phase of experiments using both linear and log-linear mixed translation model adaptations, we observe a similar trend of the Linmix LM outperforming the two other adaptation settings. Moreover, when comparing the translation model adaptation effect between the experiments in three phases, we again find the Linmix adaptation setting to perform the best among the three adaptation techniques. While all three translation model adaptations outperform the baseline scores, the Linmix adaptation provides the best scores, followed by the Conc model with again the Logmix setting performing the worst. This is slightly surprising since the Logmix weights are set by running MERT which maximises BLEU on the devset, while the Linmix weights are set by optimising maximum likelihood on the target side of the devset. Hence we would expect Logmix adaptation to perform better in the current setting. However, in the tuning phase, MERT was observed to iterate to the default iteration limit (30) in order to complete, rather than converging automatically by maximising BLEU in most of the experiments. Along with the poor performance of Logmix models, this observation strongly suggests the inability of the MERT algorithm to converge on an optimal set of weights for a reasonably large number of parameters (Chiang et al., 2009).[5] In some cases (e.g. En–Fr Logmix TrM, Linmix LM setting) however, we observe that the log-linear mixture of translation models performs almost as well as the other models with the difference in scores not being statistically significant.

In order to address the issue of sub-optimal weight estimation by MERT, we use MIRA as an alternative tuning algorithm to tune the log-linear mixture weights for the third phase (Logmix TrM adaptation) of our experiments. Comparing the MIRA scores to the MERT scores for the En–De translations we observe an improvement of 0.51 and 0.16 absolute BLEU points, for Conc and Logmix LM adaptation settings, respectively. However, the MIRA scores for the Linmix LM is found to be poorer than the MERT scores although the difference is statistically non significant. For the En–Fr translations, we find both the Conc and Linmix scores provided by MIRA to be slightly poorer than the corresponding MERT scores with the difference being statistically insignificant. However for the Logmix LM adaptation setting we observe a statistically significant improvement of 0.53 absolute BLEU points and 0.51 absolute METEOR points. Observing the relative variation in MERT and MIRA scores clearly show that both these tuning methods produce comparable scores in the current setting. Moreover, the general trend of the Linmix LM adaptation out-

[5] For the Logmix setting we have two translation models, two reordering models and three LMs, having in total of 27 parameters for MERT to optimise.

performing the other two adaptation settings is also maintained in the experiments using MIRA tuning. While MIRA addresses the convergence issues of the MERT algorithm, it does not change the overall trend of the results, thus indicating the weakness of the log-linear adaptation in the current setting.

Overall, a linear mixture-adapted TrM along with a linear mixture adapted LM is found to be the best-performing system, providing statistically significant improvements of 1.5 and 1.68 BLEU points (1.47 and 1.2 METEOR points) improvement over the unadapted baseline for En–De and En–Fr translations, respectively. All these improvements are statistically significant at p=0.05 level. Moreover, the effect of language model adaptation on translation quality is found to be more profound than that of translation model adaptation (1.75% relative improvement for LM adaptation compared to 1.17% relative improvement for TrM adaptation). This could be attributed to the fact that our target domain (forums) is different from the in-domain data (Symantec TMs) more in terms of style rather than actual content, so that the style of the translations are more affected by the LMs than the translation models.

Adapt. Setting		**En–De**		**En–Fr**	
TrM	**LM**	**BLEU**	**METEOR**	**BLEU**	**METEOR**
Baseline		21.80	39.78	31.65	50.83
Conc	Conc	∗23.04	40.94	∗‡32.51	51.31
	Linmix	∗‡**23.42**	**41.40**	∗†‡**33.04**	**52.22**
	Logmix	∗22.67	40.21	32.01	50.97
Linmix	Conc	∗23.09	40.61	∗32.53	51.60
	Linmix	∗**23.49**	**41.45**	∗†‡**33.16**	**51.92**
	Logmix	∗23.07	41.02	∗32.32	51.18
Logmix	Conc	∗22.63	40.12	∗‡32.36	51.45
	Linmix	∗†‡**23.40**	**41.26**	∗†‡**33.07**	**51.71**
	Logmix	22.20	39.57	28.50	48.84
MIRA	Conc	∗22.87	41.07	32.20	51.36
	Linmix	∗‡**23.32**	**41.53**	∗†‡**32.99**	**52.05**
	Logmix	22.48	40.06	31.12	49.33

Table 5.4: Mixture Adaptation Result using OPS as the supplementary data source.

Table 5.4 presents the mixture adaptation results using OPS as the supplementary data source. $*, \dagger\ddagger$ denote statistically significant improvements on Baseline, Conc and LogMix BLEU scores, respectively. Compared to the results in Table 5.3, we observe similar trends in the translation results across different translation and LM adaptation settings and language pairs. Within every phase pertaining to one translation model adaptation setting, Linmix adaptation of LMs performs best in terms of translation quality (for both METEOR and BLEU scores). Across different translation model adaptation settings, the Linmix adaptation is again the best-performing system. Using OPS as the supplementary source, the best combination (Linmix translation model with Linmix LM) achieves improvements of 1.69 and 1.51 absolute BLEU points over the baseline translations for En–De and En–Fr, respectively. The METEOR scores show improvements of 1.67 and 1.09 points over the baseline En–De and En–Fr translations, respectively. All these improvements are statistically significant at the p=0.05 level. Although METEOR scores improve with the same trend as the BLEU scores, the range of improvements is lower for METEOR which can be attributed to the 'near-matching' capability of METEOR.

Using MIRA to tune the weights of the log-linear combination of translation models provide improvements over the corresponding MERT scores in some (for Logmix LM adaptation) cases but the improvements are mostly statistically non significant. However, the translation scores generated using MIRA preserve the same trend as the other phases, with Linmix LM adaptation providing the best scores among the three adaptation settings.

Comparing the improvement figures observed by using OPS and EP as supplementary data source for adaptation, reveal nearly similar levels of relative improvements (6.88% using EP compared to 7.75% using OPS) for En–De translations, although improvements are slightly more visible in OPS. For the En–Fr translations however, the trend is reversed with the EP improvements being more profound (5.31% relative) compared to OPS (4.77%). Data selection improves the translation quality for two major reasons– (i) better coverage, i.e. handling of out-of-vocabulary words, and (ii) better lexical selection due to richer statistics. Counting the number of errors in the German hypothesis translations provided by the EP and OPS data selection methods, we find 4419 and 4264 errors due to lexical selection, respectively.[6] On the other hand, the number of errors due to missing words (this includes OOVs as well as general words) are 971 for the EP hypothesis and 1017 for the OPS hypothesis. Hence, although the data selected from EP corpus accounts for fewer missing words, the OPS translations provide better lexical selection. The size of the EP data is clearly greater (having a higher value of ASL) than that of the OPS dataset, which can be the cause of lesser missing words. In contrast the OPS data being stylistically more similar to the target domain[7] accounts for better lexical selection. For the En–Fr translations however, the counts of missing word errors are 1160 and 1228 for the EP and OPS hypotheses, respectively. The lexical selection error counts are 4486 and 4480 for EP and OPS, respectively. For the French translations, EP provides better translation quality simply due to less missing words (the number of lexical errors being nearly comparable).

Adapt. Setting		**En–De**		**En–Fr**	
TrM	**LM**	**BLEU**	**METEOR**	**BLEU**	**METEOR**
Baseline		21.80	39.78	31.65	50.83
Conc	Conc	21.91	39.93	∗‡32.38	51.38
	Linmix	∗†‡**22.46**	**40.59**	∗‡**32.44**	**51.39**
	Logmix	21.80	39.39	31.62	50.48
Linmix	Conc	∗22.33	40.32	∗‡32.50	51.36
	Linmix	∗‡**22.63**	**40.82**	∗‡**32.81**	**51.82**
	Logmix	22.04	39.63	31.98	50.52
Logmix	Conc	22.27	40.44	∗‡32.49	51.20
	Linmix	∗‡**22.59**	**40.86**	∗‡**32.78**	**51.63**
	Logmix	21.87	39.55	31.09	49.75
MIRA	Conc	22.32	41.04	31.92	50.97
	Linmix	∗‡**22.80**	**41.06**	∗†‡**32.55**	**51.74**
	Logmix	21.99	39.91	31.63	50.17

Table 5.5: Mixture Adaptation Result using NC as the supplementary data source.

The results in Table 5.5 which present the adaptation scores using NC as the sup-

[6] We used the hjerson toolkit (http://www.dfki.de/ mapo02/hjerson/) to measure the count of errors with respect to the reference translations.

[7] The OPS data comprises film subtitles and hence is a greater source of colloquialisms and informal content, that is also the characteristics of forums.

plementary source, again demonstrate the same trend as is observed in the results presented in Tables 5.3 and 5.4. $*, \dagger\ddagger$ denote statistically significant improvements on Baseline, Conc and LogMix BLEU scores, respectively. The Linmix adaptation setting for LM outperforms the other settings in terms of translation scores in all three phases. Furthermore, the Linmix translation model adaptation produces the best overall scores. The best-performing system is again the Linmix translation model adaptation used with Linmix LM adaptation, which improves over the baseline translation scores by 0.83 and 1.16 absolute BLEU points for En–De and En–Fr translations, respectively. The corresponding METEOR improvements are 1.04 and 1.16 points for En–De and En–Fr translations respectively. While both these improvements are statistically significant, the relative improvements are considerably lower than those observed for EP and OPS. The reason for this can attributed to the smaller size of the selected data using NC as the supplementary data source in comparison to the others. Observing the relative variation of the METEOR scores further confirms the better performance of the Linmix LM adaptation setting in each phase. Across the phases, the METEOR scores show Linmix TrM adaptation to be the most successful, while the Logmix TrM scores are better than the Conc TrM scores for both language pairs.

The same relative trend is found in the set of Logmix experiments tuned using MIRA instead of MERT. However, unlike the previous cases, the Linmix LM adaptation setting tuned using MIRA provides the best scores among all the experiments even outperforming the Linmix TrM and Linmix LM combination for En–De translations. However, the difference is statistically non-significant in the present scenario.

Further comparing the error rates for data selection from the NC corpora for En–De translations, we observe 955 missing words errors and 4461 lexical errors. Although the missing word errors are fewer compared to OPS (1017), and EP (971), the number of lexical errors for the NC corpus is much greater compared to the counts in EP (4419) and OPS (4264). For the En–Fr translations, we observe a similar trend with NC having more missing word errors (1177) than EP (1160), but less missing words than OPS (1228). The lexical selection errors for NC (4567) are much larger than the corresponding counts in EP (4486) and OPS (4480). Hence the low improvements observed using selected data from NC corpus can be attributed to more lexical errors in the translations. Since the size of the NC corpus is much smaller than both EP and OPS, the additional selected data does not enhance the lexical choice sufficiently, thus leading to smaller improvements.

Finally Table 5.6 presents the mixture adaptation scores using all three datasets together as the source of supplementary data. While the translation model and LM adaptation trends remain the same, we observe that the Logmix scores are considerably poorer than in the other two phases. In contrast to combining two phrase tables in the previous experiments, here we combine four different phrase tables, thus dramatically increasing the feature space. In this larger parameter setting MERT does a poor job of setting the optimal log-linear mixture weights, which brings down the scores. These experiments clearly show, therefore, that log-linear combination is not very scalable when the number of models increases to more than two. However, the Conc and the Linmix adaptation setting scores show that they are not affected by this scalability problem. The best-performing system is again a combination of Linmix translation models and Linmix language models providing statistically significant improvements of 1.86 and 2.09 absolute BLEU points over the baseline translation scores for En–De and En–Fr translations, respectively. The METEOR scores follow the exact same trend of improvements with improvements of 2.37 and 1.89 points for En–De and En–Fr translations, respectively.

Using the CMB dataset as the supplementary source, the difference in the MERT and MIRA scores are much more prominent, compared to the previous experiments. The MIRA scores are found to consistently outperform the MERT scores for both language pairs. Even though the differences are not statistically significant, the consistently better performance of MIRA in the current setting can be attributed to its ability to converge in large parameter settings. The drastic increase in the feature space negatively affects the translation quality of Logmix TrM adaptation setting using MERT, while MIRA is able to generate better scores in the same setting. However, despite the better performance of MIRA, the Logmix scores still fail to match the scores obtained using the other adaptation settings with Linmix performing the best among all adaptation settings.

Adapt. Setting		**En–De**		**En–Fr**	
TrM	**LM**	**BLEU**	**METEOR**	**BLEU**	**METEOR**
Baseline		21.80	39.78	31.65	50.83
Conc	Conc	∗23.04	41.02	30.11	49.73
	Linmix	∗**23.07**	**41.11**	∗†‡**33.54**	**52.48**
	Logmix	∗22.66	39.81	†32.09	50.51
Linmix	Conc	∗23.24	41.16	30.45	50.06
	Linmix	∗‡**23.66**	**42.15**	∗†‡**33.74**	**52.72**
	Logmix	∗22.84	40.47	∗†32.56	51.20
Logmix	Conc	∗‡22.63	40.37	29.57	49.49
	Linmix	∗‡**22.68**	**40.82**	†‡**31.97**	**50.64**
	Logmix	21.97	39.27	†28.87	48.53
MIRA	Conc	∗‡22.68	40.62	28.33	48.92
	Linmix	∗‡**22.99**	**41.04**	†‡**32.10**	**51.24**
	Logmix	22.19	39.36	30.99	49.04

Table 5.6: Mixture Adaptation Results using CMB (EP+OPS+NC) as the supplementary data source.

Comparing the improvements provided by the best-performing systems using different datasets, clearly indicate that using CMB data provides the best improvements of the lot. For En–De translations, the CMB data has 941 errors due to missing words and 4198 lexical errors, which are both lower than the counts from other systems. For En–Fr translations, the missing error count is 1149 and lexical error count is 4440, which are again the lowest of all settings. Using the CMB data not only improves coverage of the system, but also improves general lexical selection of the models. The reason behind this is obvious, since CMB data comprises data selected from all the other datasets.

Observing the overall trends in Table 5.3, Table 5.4, Table 5.5 and Table 5.6, we can conclude that linear mixture adaptation is more successful in model combination compared to concatenation at the data-level and log-linear mixture adaptation at least for the current setting. Moreover, LM adaptation is found to affect the translation quality more profoundly than translation model adaptation across all the different datasets and language pairs.

5.5 Observation

The experimental results revealed that supplementary data selection using a simple perplexity-based ranking and thresholding is effective and improves the translation quality with respect to an unadapted baseline system. This improvement was statistically significant across all of the three supplementary datasets and their combinations. Table 5.7 presents an example of the effect additional supplementary data has on translation quality for both language pairs. The adapted translations are generated by a system with Linmix translation and LM adaptation setting (since it is the best performing system) using EP as the supplementary dataset.

	En–De Translation	**En–Fr Translation**
Src	*if you find a good product , please* ***let*** *me* ***know*** *.*	***i am using*** *a trial nis 2010 .*
Ref	*falls sie ein gutes produkt finden ,* ***lassen sie*** *es mich* ***wissen*** *.*	***j' utilise*** *une version d' évaluation de nis 2010 .*
Baseline	*wenn sie ein gutes produkt finden ,* ***teilen sie*** *mir .*	***je suis à l' aide*** *d' une version d' essai de nis 2010 .*
Adapted	*wenn sie ein gutes produkt finden ,* ***lassen sie*** *es mich* ***wissen*** *.*	***j' utilise*** *une version d' évaluation de nis 2010 .*

Table 5.7: Effect of Supplementary data selection on translation quality

Looking at the En–De example translations, we observe that while the baseline system generates a translation with a missing verb (*wissen*), the translation provided by the adapted system is more complete and closer to the reference. Considering the En–Fr example, we find that the adapted system translates the phrase *i am using* into the appropriate translation *utilise*. In contrast the baseline system simply concatenates translations of *i am* (*je suis*) and *using* (*à l' aide d'*) leading to a less appropriate translation of the phrase. Both these examples clearly show that adding supplementary data selection to the existing in-domain baseline model improves translation quality over unadapted baseline and this improvement is irrespective of the kind of combination technique used. However, depending upon the mode of combination (data-level vs. model-level, linear vs. log-linear), the improvements vary considerably. Overall these results seem to reconfirm our findings from Chapter 4 about the positive impact of supplementary data selection as an adaptation method.

The main objective of the chapter was to compare the effect of component-level combination in domain adaptation of SMT systems. Our experiments revealed that linear interpolation works better than both concatenation or log-linear interpolation across different supplementary datasets and this improvement was observed both for language model and translation model adaptation. The log-linear mixture model suffers from the issue of sub-optimal weights being set by the MERT algorithm in a large parameter setting and hence is not scalable to the combination of more than two models. Apart from the issue of sub-optimal weight setting, log-linear combination by nature is susceptible to statistical outliers and data sparsity (Sennrich, 2012b). In contrast linear mixtures compute a weighted average of the sub-component features to produce a combined model and are much more robust to the issues of data sparsity and outliers. The concatenation approach combines the datasets instead of the models and hence involves the process of retraining from scratch which is costly both in terms of time and computation. Secondly, since a single component model is created using this method, there is no notion of relative weighting depending upon the source's fit with respect to the target domain. Furthermore, both the concatenation approach and the log-linear mixture adaptation approach were not found to be scalable when combining more than two models.

Table 5.8 presents another set of examples to highlight the actual effect of different LM adaptation techniques on translation quality. The three translations presented in Table 5.8 are generated by a Linmix translation model combined with Conc, Linmix and Logmix LM adaptation settings, respectively, using EP as the

	En–De Translation	**En–Fr Translation**
Src	*i have four (4) trojan horses : svnmgr , wingpr , sqcmgr , and symnotifywnd on my computer and my norton 2008 is not detecting them therefore , they* ***cannot be dealt with*** *.*	*a lot of people have had problems with* ***avg*** *8 too !*
Ref	*ich habe vier (4) trojaner : svnmgr , wingpr , sqcmgr und symnotifywnd auf meinen computer und mein norton 2008 erkennt sie nicht . daher können sie* ***nicht behandelt werden*** *.*	*de nombreuses personnes ont également eu des problèmes avec* ***avg*** *8 !*
Conc	*ich habe vier (4) trojanische pferde : svnmgr , wingpr , sqcmgr und symnotifywnd auf meinen computer und mein norton 2008 ist nicht erkennen sie daher , sie* ***behandelt werden kann****.*	*beaucoup de gens ont eu des problèmes avec* ***moyenne*** *8 trop !*
Linmix	*ich habe vier (4) trojaner : svnmgr , wingpr , sqcmgr und symnotifywnd auf meinen computer und mein norton 2008 ist sie daher nicht erkennt , können sie* ***nicht behandelt werden*** *.*	*un grand nombre de personnes ont eu des problèmes avec* ***avg*** *8 trop !*
Logmix	*ich habe vier (4) trojanische pferde : svnmgr , wingpr , sqcmgr und symnotifywnd auf meinem computer und mein norton 2008 ist nicht erkennen sie sie also* ***behandelt werden kann*** *.*	*un grand nombre de personnes ont eu des problèmes avec* ***moy*** *. 8 trop !*

Table 5.8: Effect of LM Adaptation on translation quality

supplementary dataset. Considering the En–De translations from three LM adapted systems reveal that only the Linmix system is able to maintain the actual meaning of the source sentence in the translation. Both Conc and Logmix translations are missing the negation on the latter part of the sentence (the translation of *they cannot be dealt with*) thereby changing the meaning completely. Additionally, the Conc translation is slightly better than the Logmix one in terms of fluency. As for the French translations, we find that the Linmix model handles the translation of the word *avg* much better than the other two adaptation settings. In the context of the source sentence *avg* is a product name, but the Conc translation considers it to be the abbreviated form of *average*, thereby translating it to *moyenne*, while the Logmix translation provides the abbreviated form of the French word *moy.* as the translation. The example clearly depicts the better performance of the Linmix adaptation setting over the other two modes of LM adaptation.

The secondary objective of our experiments was to observe the relative impact of language model adaptation compared to translation model adaptation for the current task of forum data translation. The experimental results presented in Section 5.4 clearly indicate that LM adaptation has a more significant effect on translation quality than translation model adaptation. Both the in-domain training data from Symantec TMs and the target domain forum data from the Symantec web forums are about the same products and services, but as elaborated in the previously (cf. Section 2.5.2), the difference is more in terms of style rather than content. Since a LM contributes more to the style of the translations produced, it is evident that LM adaptation has a more significant effect on translation quality. In order to compare the relative effects of LM adaptation to that of translation model adaptation on the translation quality, we present two example sentences from the testset along with their translations in Table 5.9. The first two of the three translations presented in the example are generated by a Conc translation model combined with a Conc LM and a Linmix LM respectively. The third translation is generated by a Linmix translation model, Conc LM combination. Therefore, comparing the first two translations provide an estimate of the LM adaptation effect, while comparing the third with the first one provides the effect of translation model adaptation.

	En–De Translation	**En–Fr Translation**
Src	*you can also restore files , etc* ***from*** *here .*	*i checked the* ***virus definitions they were*** *current as of 9 / 18 .*
Ref	*sie können außerdem dateien usw.* ***von*** *hier aus wiederherstellen .*	*j' ai vérifié les* ***définitions de virus : elles étaient*** *à jour , datées du 9 / 18 .*
ConcTrM + ConcLM	*sie können auch dateien wiederherstellen , usw. hier .*	*j' ai vérifié les* ***définitions de virus dont ils ont été*** *actuelles en date de 9 / 18 .*
ConcTrM + LinLM	*sie können auch dateien usw.* ***von*** *hier wiederherstellen .*	*j' ai vérifié les* ***définitions de virus , ils ont été*** *à jour du 9 / 18 .*
LinTrm + ConcLM	*sie können auch dateien wiederherstellen , usw.* ***von*** *hier .*	*j' ai vérifié* ***les définitions de virus dont ils ont été*** *actuel en date du 9 / 18 .*

Table 5.9: Relative effects of translation model and LM adaptation on translation quality

Observing the German translations in Table 5.9 we observe that the second translation (ConcTrM+LinLM) maintains the meaning of the source sentence correctly compared to the first translation (ConcTrM + ConcLM). In contrast, the third translation although having a wrong sentence structure, is slightly better than the first translation due to the inclusion of a preposition (*von*). Analysing the French translation we observe a similar trend with the second translation better handling the implied structure of the source sentence by introducing the implied comma (*de virus , ils ont*) in the translation. In contrast the third translation, although quite similar to the first one uses a better translation of the preposition *of* (*du* being better than *de* in the current context) compared to the first one. Both these examples illustrate our finding regarding the LM adaptation effect being more profound than the translation model adaptation.

Finally we used MIRA to address the issue of sub-optimal weight estimation by MERT in the large parameter settings especially for the log-linear adaptation of translation models. However, our experiments revealed that in most cases MERT and MIRA provided comparable performances in terms of translation quality when combining two sets of models. MIRA is found to perform consistently better only in the set of experiments where more than two component models (translation or language model) were combined. However even in that case, the improvement in translation quality provided by MIRA is statistically non significant in comparison to the MERT scores.

The set of graphs in Figure 5.2 presents the relative improvements over the baseline for language model adaptation and translation model adaptation. The language model adaptation graphs present the best scores within the three different phases of experiments, i.e. the experiments using a Linmix translation model. Similarly, the translation model adaptation experiments also present the best three scores in every phase using Linmix language model adaptation. The coloured circles above the histogram denote a statistically significant improvement over the other scores.

Comparing the relative variation of the translation scores pertaining to trans-

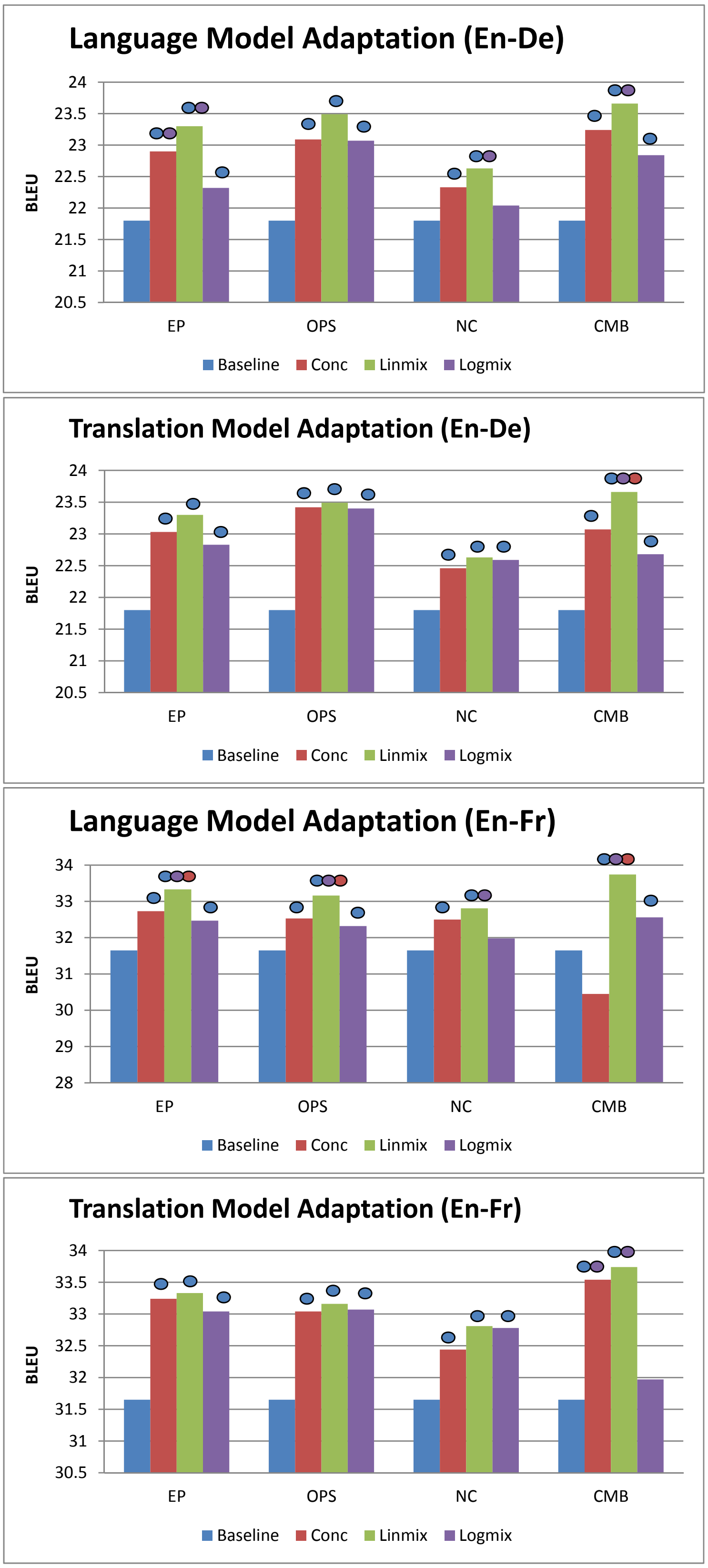


Figure 5.2: Comparing BLEU scores generated by baseline, Conc, Linmix and Logmix techniques for translation model and language model adaptations

lation model adaptation and language model adaptation further confirms our observation that language model adaptation affects the translation quality more than translation model adaptation. Furthermore, the figures also reveal that the best improvements are obtained by using the CMB dataset as the supplementary source, while the NC dataset produces the smallest improvements over the baseline scores.

5.6 Summary

In this chapter we used a technique of perplexity-based supplementary data selection in order to adapt in-domain SMT models to translate Symantec web forum content. We also used three contrastive combination techniques in order to combine the supplementary data or the models built on them with their in-domain counterpart. Our experiments reveal that supplementary data selection using the perplexity ranking method improves translation quality over an unadapted baseline. Out of the different combinations, linear interpolation achieves the best translation scores, while in most cases log-linear interpolation performs the worst. Furthermore, comparing the translation scores achieved by translation model and language model adaptation, we find that the latter is more effective in improving translation quality in the current setting. Finally we revisit the third research question (**RQ3**) presented in Chapter 1, which formed the initial motivation for the experiments in this chapter:

> **RQ3:***How can multiple models be adapted at different component levels of an SMT system and what is the effect of component-level adaptation on translation quality?*

Analysing the observations presented in Section 5.5 clearly shows that our experiments in this chapter and the associated findings provide a conclusive answer to **RQ3**.

5.6.1 Contributions

The main contributions of this chapter are as follows:

- We have successfully used a perplexity-ranking based data selection method to adapt an SMT system trained on corporate content, to translate user-generated forum data.

- We have implemented linear mixture and log-linear mixture adaptation using Moses as the SMT framework for component-level combination and shown its effectiveness in domain adaptation.

- Our experiments have shown linear mixture adaptation to be the most successful method of model combination as it outperforms both data concatenation and log-linear combination.

- We have also compared the relative effects of language model and translation model adaptation concluding that the former is more effective in improving translation quality in the current scenario.

The experiments in this chapter reconfirm the effectiveness of data selection methods in domain adaptation for translating forum content. However, this chapter focussed more on the data combination aspects of the approach using existing methods of data selection. In the next chapter we focus on developing a novel approach of data selection based on translation quality maximisation and present its effect on the current adaptation scenario.

Chapter 6

Translation Quality-Based Supplementary Data Selection

The experiments in Chapter 5 focussed on using supplementary data selection as a domain adaptation approach for the task of user-generated forum content translation. We identified two major aspects of data selection approaches in the previous chapter: (i) selecting sub-parts of the out-of-domain corpora which are relevant to the adaptation task at hand and (ii) using a specialised strategy to combine them with the in-domain data or models. Using a simple LM perplexity-based criterion (Hildebrand et al., 2005; Banerjee et al., 2011b) for 'relevant' data selection, we focussed on the second aspect of the approach in Chapter 5. Investigating different combination strategies (concatenation, linear or log-linear interpolation) at different levels of granularity (data- and model-level), our experiments revealed that while the rate of improvement varied for different combination techniques, data selection improved the translation quality generally over an unadapted baseline irrespective of the combination technique used. Motivated by the success of data selection in general, in this chapter we shift our focus to the first aspect of the approach, i.e. selection of 'relevant' data from out-of-domain corpora for adaptation in the current scenario.

With the dominance of the SMT paradigm in MT, the availability of freely available parallel corpora on the web has flourished, especially in the past decade. While some such corpora comprise data from wide-coverage domains such as politics[1] or news,[2] others are based on much more focused and narrower domains such as medical texts[3] or software manuals.[4] Using additional data for training SMT systems is strongly motivated by the data sparseness issue of the component models. Using larger amounts of training data leads to more robust estimations of word alignments, lexical and phrase translation probabilities as well as language model probabilities (Sennrich, 2012b). However, there is a significant side-effect of using additional data in this regard which is not necessarily positive. Domain specificity, or the lack of it in the training data, adds to the problem of ambiguity in translations (Axelrod et al., 2011). For example, the word *Windows* generally translates to *Fenster* in German, but in the context of the IT domain (which is also our target domain), it mostly refers to the Windows operating system. Using generic out-of-domain corpora as additional data to supplement the domain-specific models might shift the translation probability of *Windows* towards *Fenster* instead of the actually intended *Windows*, thus resulting in improper translations.[5] Adding supplementary data from 'out-of-domain' corpora tends to mitigate the data sparsity problem thus improving translation quality. At the same time, however, it accentuates the ambiguity issue in the component models, thereby potentially reducing translation quality. In order to maximise the improvements (by addressing sparsity) and minimise the deterioration (due to ambiguity), 'relevant' data selection from out-of-domain data is extremely important for these approaches. In the previous chapter we used perplexity with respect to a target domain language model as a measure of relevance. In this chapter we focus on an alternative strategy based on actual translation quality to select relevant data from supplementary corpora.

[1] http://www.statmt.org/europarl/

[2] http://www.statmt.org/wmt11/translation-task.html#download

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

[4] http://opus.lingfil.uu.se/PHP.php

[5] This problem is more acute due to the practise of lowercasing the training data prior to translation.

Given the TM-based domain-specific baseline model and a general-domain supplementary dataset, we iteratively select batches of sentences from the supplementary dataset and add them to the in-domain translation model of the baseline system and evaluate the translation quality in terms of automatic evaluation metrics on a development set. A batch is approved for addition to the baseline model only upon improvement over the baseline evaluation metric scores. In order to incrementally and rapidly retrain the translation model with each additional batch of sentences, a translation model is estimated for each batch under consideration in isolation and subsequently merged with the existing larger translation model using a phrase-table merging mechanism (Sennrich, 2012b). Prior to the iterative batch selection process, the supplementary training data is ranked using perplexity with respect to a forum data language model (similar to our approach in Chapter 5). This technique allows the selection of batches of sentence pairs from the supplementary data with perplexity scores within a close range.

Using the same scenario for domain adaptation as in the last two chapters (translation of forum content), we use the models trained on Symantec TM datasets as the 'in-domain' models while three different freely available parallel corpora (again the same as used in Chapter 5) are used as supplementary datasets. We conduct experiments for English–French (En–Fr) and English–German (En–De). Our experiments show that when incorporated into the baseline translation model, the selected supplementary datasets consistently improve translation quality over the baseline performance, for different supplementary data resources. Comparing our method of data selection with existing approaches (Koehn and Schroeder, 2007; Foster et al., 2010; Axelrod et al., 2011) confirms the superiority of our technique in terms of translation quality improvement. In addition to the data selection, we develop a phrase table merging technique as an efficient alternative to established methods of model combination. We compare our technique of model combination to the traditional approach of static retraining, use of multiple translation models (Koehn and Schroeder, 2007) as well as mixture modelling with linear interpolation (Foster and

Kuhn, 2007) to find that our technique performs at least as well as most of the other established techniques in terms of translation quality.

While the translation quality-based data selection technique performs well in the experiments presented in this chapter, there is a risk that the approach may overfit on the small development sets used (small development sets are a typical situation in real-life domain adaptation scenarios). In particular, this can happen if the set is not 'fully' representative of the target domain in question. Hence the evaluation during the iterative data selection phase should ideally be carried out for multiple development sets and the intersection of the selected datasets (from each run) should be used. However, generating multiple development sets for a given target domain is prohibitively expensive given the considerable manual effort involved. To alleviate this issue, the source data of the development set selected for the set of experiments reported here is randomly chosen from a large collection of the target domain, and proper care is taken to preserve the characteristics of the target domain during the manual translation of the development set (cf. Chapter 2, Page 46).

The rest of the chapter is organised as follows: Section 6.1 presents the motivation behind our approach. Section 6.2 introduces our approach of data selection and phrase-table merging. Section 6.3 presents the experimental setup for the different comparative approaches, including our own. Section 6.4 presents the results and analysis followed by observations in Section 6.5.

6.1 Motivation

Similar to the two previous chapters, the primary motivation of the data selection experiments reported in this chapter is the improvement of translation quality of forum content using models trained on Symantec TM data. Our experiments in the last two chapters have confirmed the importance of supplementary data selection as an adaptation approach in the current setting. However, in the previous chapters we used established techniques of data selection based on perplexity or OOV reduction.

Although widely used in literature as a measure of relevance, perplexity reduction does not often correlate with translation quality improvement (Axelrod, 2006). This motivates us to develop an alternative data selection technique based directly on translation quality improvement, which is our end objective.

The idea of supplementary data selection from related or unrelated domains to boost the performance of sparse 'in-domain' models has been widely practised in domain adaptation of SMT (Eck et al., 2004; Hildebrand et al., 2005; Foster et al., 2010; Daume III and Jagarlamudi, 2011; Banerjee et al., 2012a). Axelrod et al. (2011) present a technique of using the difference in cross-entropy of the supplementary sentence pairs on 'in-domain' and 'out-of-domain' datasets for ranking and selection by thresholding on the ranked dataset. While all these techniques use varied measures of relevance, they are driven by the eventual goal of translation quality improvement on the basis of an assumption that the relevance metric correlates well with translation quality. This correlation might hold true for some cases (e.g. OOV rate) but fail for some others (e.g. perplexity). Accordingly, we take a more direct approach to the problem by using translation quality itself as the measure of relevance thus avoiding altogether the issue of correlation between relevance metric and translation quality. To the best of our knowledge this is a novel approach and is one of the main contributions of this thesis.

The use of translation quality to select supplementary data involves combining parts of out-of-domain data with the in-domain models and iteratively retraining the component models. In order to address the issue of scalability, we resort to incremental training by proposing a phrase-table merging mechanism that is used to merge smaller phrase-tables estimated on incremental batches of supplementary data with the existing in-domain phrase-table. Incremental updates of translation models have been attempted using a stepwise online expectation maximisation algorithm (Cappé and Moulines, 2009) for stream-based translation models (Levenberg et al., 2010) or using suffix arrays (Callison-Burch et al., 2005) to store the source–target alignments in memory. Hardt and Elming (2010) reported an incremental training approach based on a local phrase-table and a smaller incremental phrase table in order to incorporate post-edit modifications to an existing SMT system. The two tables were combined using the multiple translation model handling capability of Moses. Our approach differs from these methods primarily in how we update translation model probabilities. The domain-specific aspect of our experimental setup allows us to avoid costly incremental alignment estimations. We rely on independently computed domain-specific alignments on data from each domain as word-alignments are known to benefit from domain-specific overfitting (Gao et al., 2011). Furthermore, our approach enables merging independent translation models estimated on different domain-specific word/phrase alignments, thus providing an alternative to existing model combination techniques. While simple concatenation of in-domain and out-domain data prior to (re-)training is a commonly used (but costly) technique (Foster et al., 2010), multiple phrase-tables (one per domain) can directly be combined using the decoder (Koehn and Schroeder, 2007), or interpolated using linear or log-linear weighted combination using mixture modelling (Foster and Kuhn, 2007; Banerjee et al., 2011b). Our phrase-table merging technique is motivated by the linear interpolation-based approach, but differs in our use of phrase counts to merge multiple phrase pairs. As far as we are aware, such a combination technique has not been presented in the literature to date.

6.2 Approach

This section details our translation quality-based data selection method and the phrase-table merging technique which is developed to assist in the incremental training of the phrase-tables.

6.2.1 Batching Sentence Pairs in Supplementary Data

The primary objective of our experiments is to identify those sentence pairs in the 'out-of-domain' supplementary datasets, which when incorporated into the 'in-domain' model, would improve translation performance. Ideally, for every sentence pair in the supplementary datasets, a new translation model needs to be retrained and its performance evaluated. A sentence is suitable for selection only when its inclusion improves the translation quality of the baseline system. However, it is not clear whether inclusion of a single sentence can improve the quality of a large existing translation model, unless it covers specific OOV translations. However, in order to manage the scalability of this approach, instead of evaluating individual sentence pairs, we group a number of them together in every iteration, with optimal grouping determined empirically. In addition, updating any sizeable model with a single sentence pair is unlikely to produce any measurable changes in overall translation output. The supplementary datasets are initially ranked according to their perplexity with respect to a language model estimated on the English user forum dataset. In every iteration, we collect sentence pairs whose perplexity lies within a small predefined range (to be supplied by the user as an input parameter). For our experiments reported in this chapter, we use an *ad-hoc* value of 1 for the range, although a further detailed investigation on the effect of the range size on data selection is planned future work. Since perplexity is used as a measure of 'closeness' with respect to the target domain, all pairs in the selected batch having perplexity within a small range (with a value of 1) ensures uniform closeness of all sentences within the group to the target domain. This assumption helps to reduce the scale of the iterative selection from the total number of sentences to the number of groups of sentences in the supplementary dataset.

6.2.2 Selection Algorithm

To decide whether a particular batch of supplementary sentence pairs is suitable for improving translation quality, we use the process outlined in Algorithm 1. The algorithm starts with a baseline translation model BL, a baseline translation score sc_0, a perplexity range r and a supplementary dataset comprising source and target sentence pairs along with their perplexity score. Source and target sentence pairs

Algorithm 1 Supplementary data batch selection for translation performance improvement

```
Require: $BL \leftarrow$ Baseline Model, $sc_0 \leftarrow$ Baseline Score,
Require: $sup \leftarrow \{pp_i, src_i, trg_i\}$, $r \leftarrow$ Perplexity Range;
 1: $itn \leftarrow 1$; $step \leftarrow r$;
 2: $b_{itn} \leftarrow \{\}; i \leftarrow 1$;
 3: while $not(EOF(sup))$ do
 4:    if $pp_i \leq step$ then
 5:       $b_{itn} \leftarrow b_{itn} \cup \{src_i, trg_i\}$; $i = i + 1$;
 6:    else
 7:       $model_{itn} \leftarrow train\_model\{b_{itn}\} \cup BL$;
 8:       $sc_{itn} = evaluate\_on\_dev\{model_{itn}\}$;
 9:       if $sc_{itn} \geq sc_0$ then
10:          $BL \leftarrow model_{itn}; sc_0 \leftarrow sc_{itn}$;
11:       end if
12:       $itn = itn + 1$;
13:       $step \leftarrow step + r$; $b_{itn} \leftarrow \emptyset$
14:    end if
15: end while
```

are batched into a group (lines 4-6) as long as their perplexity values fall below the specified range. Once the batch is selected, a new translation model is trained on the batch, and the batch model is merged with the baseline model to generate an updated model $model_{itn}$ (line 7). The updated model is then used to evaluate the development set using automatic evaluation metrics (line 8) and generates a new translation score sc_{itn}. The algorithm tests whether the new score is better than the previous baseline score (line 9), and if found better updates the baseline model and score with the current model and score value in the iteration. Eventually the perplexity range is extended to the next step, and the batch is cleared in order to accommodate the next batch of sentences (line 13). This process runs as long as there are no more batches to process. Selected batches are accumulated to produce the supplementary dataset used for adaptation. Since the batches are ordered according to perplexity-based similarity with the target domain, the algorithm makes it increasingly harder for a batch to get into the final selection as (i) later batches are less similar to the targeted domain and (ii) they need to improve on a steadily improving baseline. Therefore, the algorithm implements the intuition that only those parts of the generic supplementary data are selected which are good enough to generate better translation quality on the development set.

A generic SMT system is usually comprised of three different statistical components: translation model (TrM), language model (LM) and a lexical reordering model (RoM). Algorithm 1 is general enough to handle updates in all these components. However, in this chapter we only report experiments with TrM and RoM model updates and use statically trained language models (cf. Section 6.2.5).

6.2.3 Phrase-table Merging

Training a TrM for an SMT system comprises three main steps: (i) estimating word and phrase alignments from sentence-aligned parallel training data, (ii) computing lexical probabilities or word translation probabilities between source and target data and (iii) estimating a phrase-table with the 7 different features:

- Inverse ($lex(s|t)$) and direct ($lex(t|s)$) lexical weights.
- Inverse ($\phi(s|t)$) and direct ($\phi(t|s)$) phrase translation probabilities.
- Source ($c(s)$) and target ($c(t)$) phrase counts.
- Phrase penalty (always $exp(1) = 2.718$).

Ideally for every iteration step, the selected batch of supplementary sentence pairs should be combined with the 'in-domain' training data of the baseline model and a new model should be estimated. Considering the computational cost involved in full retraining, clearly this is not feasible in an iterative framework. In order to facilitate an incremental approach we develop a set of techniques to avoid full retraining by estimating a model only on the small incremental batch and then merging the models with the existing baseline models.

Word alignment estimation is the most computationally expensive process in TrM training. Thus to avoid re-estimation of word alignments in every iteration, we once and for all pre-compute the word alignments on the entire supplementary dataset and use this in every iteration. This not only reduces the estimation overhead but also addresses the issue of having poor word alignments due to small amounts of parallel data in every iteration. Word alignments are known to benefit from domain-specific over fitting (Gao et al., 2011) which motivated us to keep our 'in-domain' (computed on Symantec TM data) and 'out-domain' (computed on the supplementary dataset) word alignments separate from each other.[6] Hence the phrase pairs extracted for each domain (Symantec TMs or the supplementary datasets) are only based on domain-specific word alignments estimated on the respective corpora.

In order to achieve lexical table merging, the standard lexical tables[7] are augmented with the source and target word counts (in addition to lexical probabilities). Once new lexical tables are created on the selected batch, the baseline lexical tables are scanned for shared entries and the corresponding probabilities are updated using the formulae in (6.1):

$$\begin{aligned} lex_{merged}(e|f) &= lex_{bl}(e|f) \times \tfrac{c_{bl}(f)}{c_{bl}(f)+c_{inc}(f)} + lex_{inc}(e|f) \times \tfrac{c_{inc}(f)}{c_{bl}(f)+c_{inc}(f)} \\ lex_{merged}(f|e) &= lex_{bl}(f|e) \times \tfrac{c_{bl}(e)}{c_{bl}(e)+c_{inc}(e)} + lex_{inc}(f|e) \times \tfrac{c_{inc}(e)}{c_{bl}(e)+c_{inc}(e)} \end{aligned} \tag{6.1}$$

where $lex_{bl}, c_{bl}, lex_{inc}$ and c_{inc} indicate the baseline lexical probability, baseline word count, incremental lexical probability and incremental word count, respectively. e and f indicate the source and target words in this context. Entries which are not shared between the base model and the batch lexical tables are simply added to the new merged lexical table. Equation (6.1) emulates the lexical probabilities which would result from full retraining.

Once the lexical tables have been updated, the phrase-table estimation is completed on the batch data using the merged lexical tables. Being estimated on the merged lexical table, the inverse and direct lexical weights are already up-to-date in the new phrase-table, so only the remaining probabilities and counts need to be updated. In a similar approach to the lexical table merging strategy, every entry in the new (incremental) batch phrase-table is compared against the older (baseline)

[6] We did experiment with combined alignment models (i.e models trained on both TM+Supplementary data), but the results were slightly poorer than using domain-specific alignments.

[7] By standard we refer to the Moses lexical table structure.

phrase-table, and the shared phrase pairs are updated by the formulae in (6.2):

$$\begin{aligned}\phi_{merged}(e|f) &= \phi_{bl}(e|f) \times \tfrac{c_{bl}(f)}{c_{bl}(f)+c_{inc}(f)} + \phi_{inc}(e|f) \times \tfrac{c_{inc}(f)}{c_{bl}(f)+c_{inc}(f)} \\ \phi_{merged}(f|e) &= \phi_{bl}(f|e) \times \tfrac{c_{bl}(e)}{c_{bl}(e)+c_{inc}(e)} + \phi_{inc}(f|e) \times \tfrac{c_{inc}(e)}{c_{bl}(e)+c_{inc}(e)}\end{aligned} \tag{6.2}$$

where $\phi_{bl}, c_{bl}, \phi_{inc}$ and c_{inc} indicate the baseline phrase translation probability, baseline phrase count, incremental phrase translation probability and incremental phrase count, respectively. e and f indicate the source and target phrases in the context. Entries which are not shared are simply copied to the merged phrase-table. Again the updates applied to the inverse and direct translation probabilities (in equation (6.2)) are motivated by the aim to approximate the probabilities which would ideally have been generated by full retraining.

Using these merging techniques, we are able to merge the smaller incremental models with the larger baseline models to simulate the full retraining effect. Furthermore, since the actual training only happens on the smaller batches of selected data, it is computationally much faster than full retraining at every step. Note that (6.1) and (6.2) ensure that the updated lex_{merged} and ϕ_{merged} are true probabilities such that the conditions 0$\leq$ lex_{merged} $\leq$1 and 0$\leq$ ϕ_{merged} $\leq$1 hold true and both probabilities sum to 1.

The combination strategy used in equations (6.1) and (6.2) are quite similar to the weighted linear interpolation of the incremental and baseline probabilities, but the weights are based on relative frequencies of word/phrase pairs in the respective datasets. Hence, in contrast to standard linear interpolation, this method does not require a separate weight estimation technique. Furthermore, in contrast to the uniform weights (a single weight per resource for all phrase/word probabilities) normally used in linear interpolation, this technique uses different weights (based on frequency) for each individual phrase or word probability it combines.

6.2.4 Reordering Model Merging

While the basic idea behind phrase-table merging could also be applied to the reordering model, we choose a simpler option for re-ordering model updates. Once a new RoM is computed on the selected batch of supplementary data, every entry in it is compared to the baseline RoM, and only new entries are added to it to generate a merged RoM. For the shared entries the reordering probabilities are retained as in the baseline model. Not only does this allow faster merging of reordering models but also ensures that for shared entries, 'in-domain' reordering is preferred over the 'out-of-domain' ones.

6.2.5 Language Models

As already stated, we use statically trained LMs for all our experiments reported in this chapter. All of these LMs used are 5-gram models with modified Kneser-Ney smoothing (Kneser and Ney, 1995) and interpolated back-off. With such models adding a single n-gram into an existing model affects the probabilities and back-off values of all existing n-grams in the model. Hence incremental merging of LMs cannot be achieved as easily as in the case of TrMs. Accordingly, in the current set of experiments we use statically estimated interpolated LMs. We estimate three different 5-gram LMs on monolingual German and French user forum data, the target side of the entire TM data and supplementary datasets, respectively. We then combine them using linear interpolation. The interpolation weights are estimated by running expectation maximisation (EM) (Dempster et al., 1977) on the target side of the development set using the same technique as described in Chapter 5 (cf. page 135).

6.3 Experimental Setup

In this section, we introduce the datasets along with the specific techniques used in our experiments. We also present the experimental setups for comparing our data selection and model merging techniques with established techniques reported in the literature.

6.3.1 Datasets

The training data for our baseline systems consists of En–De and En–Fr bilingual datasets in the form of Symantec TMs. Monolingual Symantec forum posts in German and French along with the target side of the TM training data serve as language modelling data. In addition, we also have a sizeable amount of English forum data which is used to create the language model with respect to which the supplementary datasets are ranked. The development (dev) and testsets are randomly selected from this English forum dataset, ensuring that they are representative of the forum data in terms of different statistics (cf. Chapter 2, page 46), and manually translated by professional translators. Similar to the experiments in Chapter 5, we use the same three freely available datasets as supplementary sources of data in this chapter:

1. Europarl (Koehn, 2005) version 6 (EP): a parallel corpus comprising of the proceedings of the European Parliament.

2. News Commentary Corpus (NC): released as a part of the WMT 2011 Translation Task.[8]

3. OpenSubtitles2011 Corpus (OPS):[9] a collection of documents released as part of the OPUS corpus (Tiedemann, 2009).

In addition to these datasets, we also use a combined version (CMB) of the three supplementary datasets as the fourth supplementary resource in our experiments. Table 6.1 reports the number of sentences in the different datasets along with the average sentence length (ASL.) used in all our experiments.

Although we use the same supplementary datasets as was used in the previous chapter, the combined dataset (CMB) is used in a slightly different way in our experiments in this chapter. In Chapter 5, the CMB dataset comprised a simple concatenation of the selected sub-parts of the individual datasets. In contrast, the CMB dataset used in this chapter comprises a concatenation of the full datasets prior

[8] http://www.statmt.org/wmt11/translation-task.html

[9] http://www.opensubtitles.org/

	dataset	**En–De**			**En–Fr**		
		Sent. Count	**En ASL**	**De ASL**	**Sent. Count**	**En ASL**	**Fr ASL**
Bi-text	Training	832,723	12.86	12.99	702,267	12.42	14.86
	devset	1,000	12.91	12.20	1,000	12.91	14.99
	testset	1,031	12.75	11.99	1,031	12.75	14.69
Supp. Data	EP	1,721,980	27.48	26.11	1,809,563	27.34	30.35
	NC	135,758	24.34	24.98	115,085	24.79	29.06
	OPS	4,649,247	7.61	7.16	12,483,718	8.61	8.17
	CMB	6,506,985	13.22	12.54	14,408,366	11.09	11.13
Mono-lingual	**Forum Data**	**Sent. Count**			**ASL**		
	English	1,129,749			12.48		
	German	42,521			11.78		
	French	41,283			14.82		

Table 6.1: Number of sentences and average sentence length for in-domain, supplementary data and monolingual forum datasets

to any selection. Since all our supplementary datasets are sorted according to their perplexity, the CMB dataset is generated by concatenating all the other datasets and re-sorting on their sentence-level perplexity values. This process ensures that all the sentences from the different supplementary datasets which are closest to the target domain appear at the top of the CMB dataset. The primary objective of using the CMB dataset in our experiments is to highlight the success of our technique in directly selecting relevant data from a mixture of multiple datasets.

6.3.2 Experiments

The main goal of the experiments reported in this chapter is data selection from supplementary parallel training data for domain adaptation. In order to evaluate the effect of our data selection technique, we compare our method with other established methods reported in the literature. Furthermore, as combination techniques are an important aspect of data selection-based adaptation experiments, we also compare different existing mechanisms to combine the selected data with in-domain data.

Baseline

Prior to running the incremental data selection experiments, the baseline TrMs are estimated on the 'in-domain' (Symantec TM) datasets. The standard Moses training scripts are modified to augment the actual word counts (both source and target words) to the existing lexical table format. The scoring mechanism of Moses is adjusted to handle the variation in the lexical table formats. This modified version of the training scripts is then used to estimate the baseline translation model only on the Symantec TM data. The reordering model is estimated on the same data using the standard mechanism as described in Chapter 2.

As already stated, the language models used in our baseline models are statically interpolated language models. Four different interpolated LMs, one pertaining to the target side of each supplementary data resource, are estimated using the technique reported in Section 6.2.5. For experiments with a particular supplementary dataset, we used the respective interpolated LMs for the baseline as well as all other models. Therefore, the baselines for each set of experiments (for every supplementary dataset) have the same translation model but different language models. The GIZA++ alignments for each of the supplementary datasets are pre-computed and used in the iterative setup.

Data Selection Experiments

To evaluate the quality of our translation quality-based data selection, we compare the following four data selection techniques:

1. Full: The naive approach of using the full data for adaptation.
2. PP: Data selection by ranking the supplementary data using perplexity with respect to the target domain and thresholding (Foster and Kuhn, 2007).
3. PPD: Using the difference in cross-entropy between in-domain and out-of-domain datasets to rank supplementary data followed by thresholding (Axelrod et al., 2011).

4. TQS: Our approach of translation quality-based data selection (cf. Section 6.2).

The Full technique is not a data selection technique *per se*, as it refers to the practice of using the entire supplementary data for adaptation. Although it is not a particularly popular approach in the domain adaptation literature, we use it to primarily highlight the importance of relevant data selection in the context of domain adaptation of SMT systems.

The second method of data selection (PP) based on perplexity ranking of source sentences has already been discussed in detail in Chapter 5. In order to rank the supplementary dataset sentences by perplexity, we use an LM trained on the English forum data as the target domain LM. For each sentence on the source side of the supplementary dataset, its perplexity is computed on this target domain LM according to the formula in (6.3).

$$PP(s|p,q) = 2^{-\sum_x p(x) log q(x)} = 2^{H(p,q)} \tag{6.3}$$

where s denotes the source sentence in the supplementary dataset, p denotes the empirical n-gram distribution in the sentence and q represents the target LM. $H(p,q)$ is the cross-entropy between p and q. Once the perplexity values are computed the sentences are sorted accordingly, thereby ensuring that the sentences which are closest to the target domain (i.e. those that have the lowest perplexity score) appear at the top. The data selection is eventually performed by selecting the top-N sentences from the ranked corpora. However, in our approach in Chapter 5, the value of N was decided such that the selected data did not exceed the size of the in-domain corpora. In the experiments in this chapter, the value of N is decided by the number of sentences selected by our TQS method for fair comparison.

Following the technique presented in Axelrod et al. (2011), the cross-entropy-based ranking (PPD) approach requires an out-of-domain LM in addition to the existing in-domain LM. Based on the original technique by Moore and Lewis (2010), the source-side sentences in the supplementary training data are ranked using the formula in (6.4):

$$H_i(s) - H_o(s) \tag{6.4}$$

where H_i and H_o represent the cross entropy of the sentence s with respect to the in-domain and out-of-domain LMs, respectively. In contrast to the PP approach, this technique biases towards the sentences which are *like* the in-domain corpus and at the same time *unlike* the average of the out-of-domain corpora. While this formulation was monolingual, Axelrod et al. (2011) proposed a bilingual extension to the formulation for application in parallel data selection. Hence the differences in cross-entropy as presented in equation (6.4) are added together for both the source and the target sentences to rank individual sentence pairs in the supplementary datasets using the formula in (6.5):

$$[H_{i-src}(s) - H_{o-src}(s)] + [H_{i-trg}(s) - H_{o-trg}(s)] \tag{6.5}$$

An out-of-domain LM is built on a randomly selected sub-sample of the supplementary training data having the same number of sentences and the same vocabulary as the in-domain LM.[10] A similar set of in-domain and out-of-domain LMs are also built on the target language side using the German and French forum datasets for in-domain LMs and random samples from supplementary datasets as the out-of-domain LMs. Eventually, each supplementary data sentence pair is ranked according to the formula in (6.5). As in the case of PP, the sentence-pairs are sorted by these scores and the lowest-scoring sentences are selected by using a thresholding value, which is set by the number of sentences selected by the TQS method.

The sentences selected using our TQS technique are selected in batches using the approach described in Section 6.2.2. The particular evaluation metric used to evaluate the quality of a batch on the devset is BLEU (Papineni et al., 2002), although the algorithm is generic enough to allow the use of any other evaluation metric for the

[10] The same vocabulary size of the in-domain and out-of-domain LMs is according to the experimental approach described by Moore and Lewis (2010).

same purpose. In order to speed up the translation process in the iterative framework, we utilise the multi-threaded feature of the Moses decoder. Furthermore, the merged phrase-table and the reordering models are filtered using the source side of the devset to reduce memory requirements as well as ensure faster decoding. While the other two ranking techniques require the selection of a thresholding value to select an appropriate subset of the supplementary data for adaptation, TQS is designed to automatically select a subset of the same. Accordingly, we use the number of sentences selected by the TQS method as the thresholding value for PP and PPD selection schemes.

Data Combination Experiments

Once the supplementary data is selected, this data needs to be combined with the in-domain training data for adaptation. Our experiments in Chapter 5 have already highlighted that the simple method of concatenating out-of-domain data is not always the best possible combination approach in the present scenario. Therefore, we investigate four configurations of data or model combination based on existing methods in the SMT literature.

1. Conc: the naive approach of concatenating the selected data with the in-domain data and retraining the SMT model from scratch.

2. Multiple phrase-table (MPT): creating separate phrase-tables for the in-domain and the selected data and using the multiple decoding path feature of the Moses decoder (Koehn and Schroeder, 2007).

3. Linear Interpolation (Linmix): using a weighted linear interpolation to combine the individual phrase-tables (Foster and Kuhn, 2007).

4. PTM: using the phrase-table merging technique reported in this chapter (cf. Section 6.2.3).

In the concatenation approach (Conc), the selected supplementary data is added to the in-domain training data and a new translation model is retrained from scratch.

This model is then tuned using the devset and finally tested using the testset to reveal the effect of adaptation. We have already observed the effect of this adaptation in Chapter 5 for the different datasets and have found it to work better than log-linear combination in most of the cases.

The Multiple phrase-table (MPT) approach requires training separate phrase-tables on the in-domain and selected data and combining them using the multiple decoding path feature of the Moses decoder. As previously stated, Moses allows the combination of multiple phrase-tables under two different configurations. (i) the *Both* configuration wherein the decoder scores every translation option from both phrase-tables. This configuration requires all phrase pairs to be shared between both phrase-tables; (ii) the *Either* configuration where translation options are scored from either of the phrase-tables. In the current scenario, both phrase-tables do not share all the phrase pairs (they are trained on different datasets). Hence we use the *Either* option to combine the two phrase-tables in this combination method. The weights of the features in each phrase-table are set by running MERT (Och, 2003) on the devset in order to optimise BLEU scores.

In the linear interpolation approach (Linmix), the two phrase-tables are combined using weights in a linear interpolation scheme. In order to learn the interpolation weights, LMs are constructed on the target side of the in-domain training set and the selected supplementary data. These LMs are then interpolated using EM on the target side of the devset to learn the optimal mixture weights. These weights are subsequently used to combine the individual feature values for every phrase pair from two phrase-tables using the formula in (6.6).

$$p_{linmix}(s|t) = \lambda p_{in}(s|t) + (1 - \lambda)p_{out}(s|t) \tag{6.6}$$

where $p_{in}(s|t)$ and $p_{out}(s|t)$ are the feature values of individual phrase pairs from the in-domain and out-of-domain phrase-tables, respectively. λ is the tunable weight between 0 and 1. Similar to the other combination techniques, after Linmix combination, the phrase-tables are subjected to MERT to set the feature weights before being tested on the testsets.

The phrase-table merging (PTM) technique outlined in Section 6.2 was originally developed to rapidly combine incremental and baseline translation models to aid our iterative data selection method. However, here we use it as an alternative technique to combine the in-domain and out-of-domain phrase-tables. While the basic idea behind this technique is similar to that of linear interpolation, in our technique each feature is weighted according to its frequency in the respective phrase-tables which is in contrast to using a global weight for every feature in Linmix. Following model combination, all the models are tuned using MERT on the devset.

6.4 Results and Analysis

As stated in Section 6.3.2, the incremental data selection process is performed by evaluating translation quality in terms of BLEU scores on the devset data. Hence we first present the scores achieved on the devset using our data selection method. Table 6.2 reports the baseline scores, the best scores and the number of sentence pairs selected during the process of incremental data selection on the devset. $*$ indicates statistically significant improvement, best scores are in bold. Alongside the number of selected sentences, the percentage figures indicate the proportion of the selected sentences with respect to the entire size of the supplementary datasets as reported in Table 6.1. Note that the BLEU scores reported in this table are all non-MERT scores, as these are the scores achieved directly using our data selection method in the iterative framework, prior to running MERT. The supplementary data is combined with the baseline model using the PTM method as outlined in Section 6.2.

The scores in Table 6.2 clearly show the improvements observed on the devset for both language pairs across all supplementary datasets. The improvements obtained using the Europarl (EP), Open-Subtitles (OPS) and the combined (CMB) corpus

	Model	EP	OPS	NC	CMB
En–De	Baseline	22.97	22.94	22.91	23.09
	Best	***24.17**	***24.33**	**23.34**	***24.82**
	Sent Count (%)	663,127 38.51%	1,464,798 31.51%	15,473 11.39%	1,707,104 26.23%
En–Fr	Baseline	31.33	31.72	31.16	31.61
	Best	***31.85**	***32.77**	**31.34**	***32.92**
	Sent Count (%)	368,777 20.38%	1,869,765 14.98%	14,511 12.61%	4,336,949 30.10%

Table 6.2: BLEU scores on the devset using incremental TrM updates and number of sentences selected.

as supplementary data sources are statistically significant at the p=0.05 level using bootstrap resampling (Koehn, 2004) for both language pairs. However, the improvements using the News-Commentary (NC) corpora as the supplementary dataset are not statistically significant over the baseline scores. Compared to the improvements obtained on the other two sets, NC improvements are much lower, which could be attributed to the smaller size of the corpus and hence consequentially the smaller size of the selected dataset. The CMB dataset provides the highest improvements in both language pairs. Considering that the CMB dataset comprises the combination of most relevant (having lowest perplexity) sentence pairs from all the other datasets, this result is unsurprising. Note that the number of selected sentences as reported in Table 6.2 for each supplementary dataset are used as the threshold values for data selection in the PP and PPD ranking methods.

6.4.1 Data Selection Results

As the primary objective of our approach is data selection from supplementary sources, we first report the results of our data selection methods in comparison to the other data selection techniques described in Section 6.3.2. In this phase, the selected supplementary data is concatenated with the in-domain training data to train new translation models which are then tuned using MERT on the devset. Table 6.3 reports the BLEU and METEOR scores for the different data selection techniques on the testset. Note that the results reported in Table 6.3 use simple data concatenation as the method for combination. ∗, †, ‡, § indicates statistically significant improvement in BLEU over baseline, PP, PPD and Full datasets, respectively. Best scores are in bold.

	System	**EP**		**OPS**		**NC**		**CMB**	
		BLEU	METEOR	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR
En–De	Base	21.98	40.64	22.56	40.75	22.10	40.03	22.43	40.37
	PP	∗22.69	40.94	∗23.03	40.98	22.24	40.50	∗23.03	41.58
	PPD	∗22.80	41.23	∗23.14	41.23	22.34	40.75	∗23.13	42.38
	Full	∗22.58	41.09	22.67	40.90	22.20	40.36	22.41	40.53
	TQS	∗§**23.10**	**41.45**	∗§**23.50**	**41.66**	**22.47**	**40.75**	∗§**23.50**	**42.79**
En–Fr	Base	31.87	51.04	32.52	51.38	31.82	51.04	32.39	51.23
	PP	∗32.73	51.76	∗33.18	52.05	§32.28	51.36	∗33.02	52.30
	PPD	∗§33.03	52.29	∗33.26	52.10	∗§32.38	51.42	∗§33.46	52.82
	Full	32.39	51.59	32.96	52.25	31.59	50.79	32.59	52.28
	TQS	∗§†‡**33.58**	**52.41**	∗§**33.56**	**52.47**	∗§**32.56**	**51.55**	∗§†‡**34.04**	**53.04**

Table 6.3: Testset BLEU and METEOR scores using four data selection methods.

The scores reported in Table 6.3 show that adding *Full* supplementary datasets to the in-domain training data improve translation quality scores over the baseline in nearly all cases. The quality actually deteriorates over the baseline by an insignificant amount when adding the *Full* NC data to the En–Fr in-domain training data . We see a similar minor drop on addition of the *Full* CMB data to En–De in-domain data, but the METEOR scores show an improvement. In both cases the drops are not statistically significant. The additional data addresses data sparsity issues in the in-domain training data, thereby improving the translation quality in some cases. In other cases, however, the additional out-of-domain data also increases ambiguity, thus leading to poorer scores.

Looking at the evaluation scores provided by the actual data selection methods (PP, PPD and TQS) in Table 6.3, we see improvements over the baseline as well as the Full data addition scores. Since all these data selection methods aim at selecting only a relevant part of the supplementary data, the selected data addresses the sparsity issue but at the same time keeps the ambiguity issue under control. As a result we observe statistically significant BLEU improvements at the p=0.05 level using most of the supplementary datasets across both language pairs. The METEOR scores also reveal similar improvements for all the supplementary data selection techniques. The only exception to statistically significant BLEU improvements over the baseline occurs when using the NC dataset for En–De. This can be attributed to the fact that the amount of supplementary data selected from the NC corpus is the lowest among all data selections from all other corpora (cf. Table 6.2).

Comparing the translation quality scores between PP, PPD and TQS, we observe that while the PPD scores are slightly better than the PP scores, the TQS method performs best, consistently improving over the other two data selection methods in terms of BLEU. The METEOR scores follow a similar trend with TQS providing the best scores among the different data selection methods. Both PP and PPD methods select only those sentence pairs which are closest to the target domain in terms of perplexity or cross-entropy. In contrast, the TQS method selects only those batches of sentence pairs for which translation quality improves. Since perplexity reduction on selected data is known to have no correlation with translation quality improvements (when same data is added for training TrMs) (Axelrod, 2006), the TQS methods selects sentence pairs that are more relevant to the context and the task at hand. Hence, while for PP and PPD methods, all of the top N sentence pairs are selected, the TQS method rejects some of the batches as they cause to degrade the translation quality. This distinction allows the TQS method to provide better translation quality compared to the other data selection techniques.

When using EP as the supplementary corpus the TQS method provides improvements of 1.12 and 1.71 absolute BLEU points (0.81 and 1.37 METEOR points) over the baseline scores for En–De and En–Fr translations, respectively. We observe similar improvements over both language directions using OPS, NC and CMB datasets. While the EP, OPS and CMB improvements are statistically significant at p=0.05 level for both language pairs, for NC only the En–Fr improvement is statistically significant. The METEOR scores show similar trends of improvement or deterioration as the BLEU scores for nearly all the datasets and language pairs although the range is different. The statistical significance of the improvements over the baseline as well as the *Full* scores clearly indicate the success of TQS as a method of data selection in the current context.

6.4.2 Data Combination Results

The results reported in Table 6.3 use the Conc approach (cf. Section 6.3.2) to combine the additional data with the in-domain dataset. However, combining in-domain and out-of-domain datasets using this approach may not always lead to the best results as is evident from the literature (Foster and Kuhn, 2007; Banerjee et al., 2011b), as well as our experiments in the previous chapter. Hence in the second phase of our experiments we compare the translation quality achieved by using the different combination methods explained in Section 6.3.2. Since the data selected by the TQS method was the best-performing dataset as per Table 6.3, we report the results of the different data combination experiments using this particular set only. Table 6.4 reports the effect of different data combination methods on translation score using data selected by the TQS method. † indicates statistically significant improvement in BLEU over MPT methods. Best scores are in bold.

	Syst-em	**EP**		**OPS**		**NC**		**CMB**	
		BLEU	METEOR	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR
En–De	Conc	23.10	41.45	23.50	41.66	22.47	**40.75**	23.50	41.58
	MPT	23.15	41.46	23.25	41.30	21.75	40.22	22.75	40.55
	PTM	23.17	41.58	23.78	41.73	22.58	40.60	23.28	41.39
	Linmix	**23.23**	**41.58**	†**23.80**	**41.94**	†**22.66**	40.70	†**23.72**	**41.93**
En–Fr	Conc	33.58	52.41	33.56	52.25	32.56	51.55	34.04	53.04
	MPT	33.31	52.22	33.34	52.31	32.20	51.13	33.87	**53.08**
	PTM	33.30	52.29	33.71	52.48	32.66	51.43	34.05	52.74
	Linmix	**33.75**	**52.74**	†**33.84**	**52.87**	†**32.79**	**51.74**	**34.15**	52.63

Table 6.4: Testset BLEU and METEOR scores using different data combination methods.

The translation quality scores in Table 6.4 confirm our assumption that concatenation is not always the best option to combine multiple datasets. The results show weighted linear interpolation (Linmix) to outperform the other approaches, but the difference is statistically insignificant. Multiple phrase-tables (MPT) are found to work better than Conc in some cases (EP datasets for En–De and En–Fr) but in most cases is poorer than all the other methods. Weighted linear interpolation (Linmix) is known to work well in multi-domain phrase-table combination (Banerjee et al., 2011b) and our experiments in this chapter again confirm this assumption. Interestingly, using our phrase-table merging method (PTM) for model combination seems to work reasonably well for all the different datasets and language pairs. While it does not outperform the Linmix technique, it certainly performs on a par with the other combination techniques, the differences being statistically insignificant in most cases. The METEOR scores following a similar trend as the BLEU scores further confirm the findings in this phase of our experiments.

Using the MPT configuration has a major advantage over the Conc approach in keeping the in-domain and out-of-domain phrase-tables separate. While this can really be an effective choice in some cases, this model has a larger number of parameters which are difficult to optimise using MERT (Chiang et al., 2009).[11] The linear interpolation mechanism avoids the large parameter setting by combining features from multiple tables into a single table. Moreover, Linmix weighs each phrase-pair belonging to different phrase-tables according to their fit to the target domain. Hence phrase-pairs belonging to in-domain phrase-tables are weighted higher than those from the out-of-domain phrase-tables, for shared phrase pairs. For the phrase pairs which are not shared, they are added ‘as-is’, improving the coverage of the model. These factors contribute towards the slightly better performance of Linmix over the other combination methods. However, this requires the estimation of the interpolation weights and it is not very straightforward to optimise the linear weights directly in terms of translation quality. In contrast to the Linmix method which uses

[11] Experiments using an alternative tuning technique – MIRA – did not provide necessary and consistent improvements in the previous chapter, so we did not use them for the experiments in this chapter.

global weights for all phrase pairs, the PTM method uses different weights based on the frequency of occurrence of the phrase pairs in each corpus. This avoids the problem of linear interpolation-based weight optimisation as well as the large parameter setting. In our experimental setting this method performs nearly as well as Linmix, thus making it a viable alternative method of data combination.

6.4.3 Combining Data Selection and Model Combination

The results in Table 6.4 indicate that linear interpolation of phrase-tables provides the best scores among different data combination techniques at least for the datasets under consideration. Hence in the final phase we present the results on different data selection methods using linear interpolated mixture models as the combination technique in Table 6.5. †, ‡, § indicate statistically significant BLEU improvements over PP, PPD and Full scores, best scores are in bold.

	System	**EP**		**OPS**		**NC**		**CMB**	
		BLEU	METEOR	BLEU	METEOR	BLEU	METEOR	BLEU	METEOR
En–De	Base	21.98	40.64	22.56	40.75	22.10	40.03	22.43	40.37
	PP	∗22.96	41.23	∗23.13	41.23	22.33	40.65	∗§23.38	41.68
	PPD	∗23.05	41.24	∗23.26	41.43	22.41	40.53	∗§23.48	41.55
	Full	22.73	41.07	22.83	40.69	22.25	**41.00**	22.76	40.99
	TQS	∗§**23.23**	**41.58**	∗†‡§**23.80**	**41.94**	∗**22.66**	40.70	∗§**23.72**	**41.93**
En–Fr	Base	31.87	51.04	32.52	51.38	31.82	51.04	32.39	51.23
	PP	∗33.00	52.08	∗33.25	52.09	∗32.41	51.32	∗§33.87	52.63
	PPD	∗33.29	52.33	∗33.32	51.90	∗§32.62	51.73	∗§33.92	52.82
	Full	∗32.80	51.81	33.01	52.30	31.96	51.24	32.70	51.41
	TQS	∗†§**33.75**	**52.74**	∗†‡§**33.84**	**52.87**	∗§**32.79**	**51.74**	∗§**34.15**	**52.96**

Table 6.5: Testset BLEU and METEOR scores with Linmix as combination method.

Using linear interpolation to combine the models built on different datasets results in a more-or-less uniform improvement in all translation quality scores for all datasets and language directions when compared to the results in Table 6.3. As in the case of using concatenation, the data selected using the TQS method provides statistically significant improvements over the baseline scores as well as those using the *Full* dataset for almost all of the datasets. Note that using the NC corpora as the supplementary dataset also results in statistically significant improvements over the baseline scores for both language pairs in this phase of our experiments. Furthermore, the TQS scores are now significantly better than both PP and PPD scores for the En–Fr translation on both EP and the OPS datasets and for the En–De translations on the OPS dataset. Overall, using Linmix as the method of combination resulted in TQS providing a higher degree of improvement over the baseline scores when compared to the improvements reported in Table 6.3. As per the results in Table 6.5, the TQS method improves over the baseline by 1.25, 1.24, 0.56 and 1.29 absolute BLEU points for En–De translations using EP, OPS, NC and CMB datasets, respectively. For the En–Fr datasets the improvement figures are 1.88, 1.32, 0.97 and 1.76 absolute BLEU points for EP, OPS, NC and CMB datasets, respectively. As in the previous cases, the METEOR scores show a similar trend of improvement with respect to the BLEU scores, but the range of improvements varies.

The overall results in Tables 6.3 and 6.5 strongly suggest the success of data selection as an adaptation technique compared to the simple choice of full data selection. Selectively adding supplementary data widens the coverage of the translation models by reducing the number of untranslated words in the translations. Additionally it also provides richer lexical translation probabilities for some phrases and words which, although present in the baseline models, were sparsely represented. This leads to better lexical selection in the final translations leading to an improvement in translation quality. Furthermore, our experiments have empirically shown that our translation quality-based data selection method consistently outperforms perplexity ranking-based data selection approaches irrespective of the combination technique employed. The phrase-table merging technique which was developed originally to support incremental training in the iterative framework, is found to perform at least as well as existing techniques for model or data combination as per the results in Table 6.3.

6.5 Observations

The experimental results across different tables in the previous section clearly highlight the usefulness of additional data in improving translation quality. As previously observed in chapters 4 and 5 of this thesis, additional data results in richer estimation of the model statistics on which the TrM is based. Since translation quality correlates strongly with the frequency of words in the training corpora (Koehn and Knight, 2001), larger amounts of training data leads to better translation quality. However, the effect of additional data is not always positive on translation quality especially for domain-specific corpora (Sennrich, 2012b). Adding a lot of out-of-domain data to an existing in-domain corpus tends to overwhelm the 'in-domain' characteristics of the model statistics, thus leading to the problem of ambiguity. Comparing the translation quality scores obtained using data selection methods (PP, PPD or TQS) to those generated by using the entire supplementary dataset (Full) clearly supports this observation. Data selection methods tend to select only relevant parts of the supplementary data such that their incorporation makes the model statistics richer while preserving the important inherent characteristics of the model. As a result, such data selection methods improve translation quality but minimise deterioration due to ambiguity. We present an example of the translations of a single sentence from our testsets produced by the baseline, *Full* and TQS data selection methods to further explain these differences in quality.

Considering the German translation example in Table 6.6, we observe that the baseline system is unable to translate the phrase *yeah, i thought* and renders it as is. The *Full* system, which incorporates the entire supplementary data, is able to translate the particular phrase due to improved coverage of the models. However, the increased generalisation of the model results in translating the domain-specific term *quick scan* to *schnelle prüfung* which is clearly not the appropriate translation in the context. Finally considering the translation provided by the TQS method which selects only a part of the full data, it improves on both the initial phrase

	En–De Translation	**En–Fr Translation**
Src	***yeah , i thought*** *a* ***quick scan*** *may take place , but certainly not a full system scan , like in version 1 .*	*re :* ***trojan*** *- major problems : brastk.exe*
Ref	***ja , ich denke*** *, dass ein* ***quick scan*** *ausgeführt wird , auf keinen fall ein vollständiger systemscan , wie in version 1.*	*re :* ***trojan*** *- problèmes majeurs : brastk.exe*
Bl.	***yeah , i thought quick scan*** *kann stattfinden , aber sicherlich nicht einen vollständigen systemscan , wie in version 1 .*	*re :* ***trojan*** *- problème majeur : brastk.exe*
Full	***ja , ich dachte*** *, eine* ***schnelle prüfung*** *erfolgt , aber sicher kein vollständiger systemscan , wie in version 1 .*	*re :* ***cheval de troie*** *- problèmes majeurs : brastk.exe*
TQS	***ja , ich dachte*** *einen* ***quick scan*** *kann stattfinden , aber nicht wirklich ein vollständiger systemscan , wie in version 1 .*	*re :* ***trojan*** *- problèmes majeurs : brastk.exe*

Table 6.6: Comparative effect of baseline, full and relevant data selection on translation quality.

as well context-specific translation of *quick scan*. Similar behaviour is observed in the French translations where the baseline fails to properly translate the phrase *major problems*, losing the plural in the translation process. Using the *Full* dataset mitigates the plural issue, but translates *trojan* to *cheval de troie* which is a proper translation of the domain-specific term (*trojan*) but is not the most idiomatic on the French forum.[12] The relevant data selection method, however, maintains the domain-specific translation of *trojan* while also handling the plural issue in the generic part of the sentence. Both these examples clearly indicate the importance of additional data and the associated issues if 'relevant' data is not properly selected from an out-of-domain corpus.

Considering the advantage of data selection, the primary objective of this chapter was to introduce a novel data selection method that directly utilises translation quality (in terms of BLEU scores on the devset) to select relevant parts of the supplementary data. Using a phrase-table merging method, which simulates the effect

[12] Doing quick searches on the French forum return 18 discussions for "cheval de troie" vs 40 for trojan.

of full training, batches of supplementary data are added incrementally to the existing in-domain models and the combination is evaluated on the devset. Only those batches which improve translation quality over the previously set baseline scores are incorporated into the selected dataset. Experimental results in the previous section confirm that this translation quality-based data selection (TQS) provides better translation scores compared to the other existing data selection methods based on perplexity (PP) or cross-entropy (PPD). The PP and PPD methods use perplexity or cross-entropy as a measure of relevance to rank the supplementary datasets and selecting the top-ranking sentences from them. Both these methods work under the assumption that since the topmost sentences have low perplexity given the target domain, they would be relevant in translating the content from the same domain. However, in reality, perplexity reduction often has no correlation with translation quality improvement (Axelrod, 2006). Our approach works around this issue by using translation quality directly to estimate the relevance of a batch of supplementary sentences. Furthermore, our approach does not require estimation of a threshold value which is essential for data selection using the ranking methods. The example presented in Table 6.7 highlights the benefits of our data selection method over the other two comparable techniques.

The German translations in Table 6.7 obtained using the PP or PPD methods miss the translation of the word *experiencing*, thus generating improper translations of the source sentence. The translation provided by the TQS method generates the translation of the phrase *experiencing this* as *dieses auftritt*, thereby improving the structure and fluency of the translated sentence. Observing the French translations in the example, we find that both the PP and PPD translations actually provide an incorrect translation for the first part of the source sentence due to the presence of the word *ne* in the translations. The PPD translation provides a better translation (in terms of fluency) for the second part of the sentence in comparison to the PP translation. The TQS translation maintains the actual sense of the first part of the sentence as well as providing a fluent translation for the second part, thus at the

	En–De Translation	**En–Fr Translation**
Src	*am i the only person **experiencing this** and is there a solution at all ?*	***i can download** a setup file but it won ’t actually do anything other than tell me “ download already complete ” .*
Ref	*habe nur ich **dieses problem** und gibt es eine lösung ?*	***je peux télécharger** un fichier d’ installation , mais il se contente d’ afficher le message “ téléchargement déjà terminé ” .*
PP	*bin ich die einzige , **dies** und gibt es eine lösung auf alle ?*	***je ne peux télécharger** un fichier de configuration mais il ne fait pas quelque chose d’ autre que de me dire “ téléchargement déjà terminé ” .*
PPD	*bin ich die einzige person , **die dies** und gibt es eine lösung ?*	***je ne peux télécharger** un fichier d’ installation , mais il ne veut pas réellement faire autre chose que me dire “ téléchargement déjà terminé ” .*
TQS	*bin ich die einzige person **dieses auftritt** und gibt es eine lösung ?*	***je peux télécharger** un fichier d’ installation , mais il ne fait rien d’ autre que de me dire “ téléchargement déjà terminé ” .*

Table 6.7: Comparative effect of PP, PPD and TQS data selection methods on translation quality.

same time providing a better overall translation.

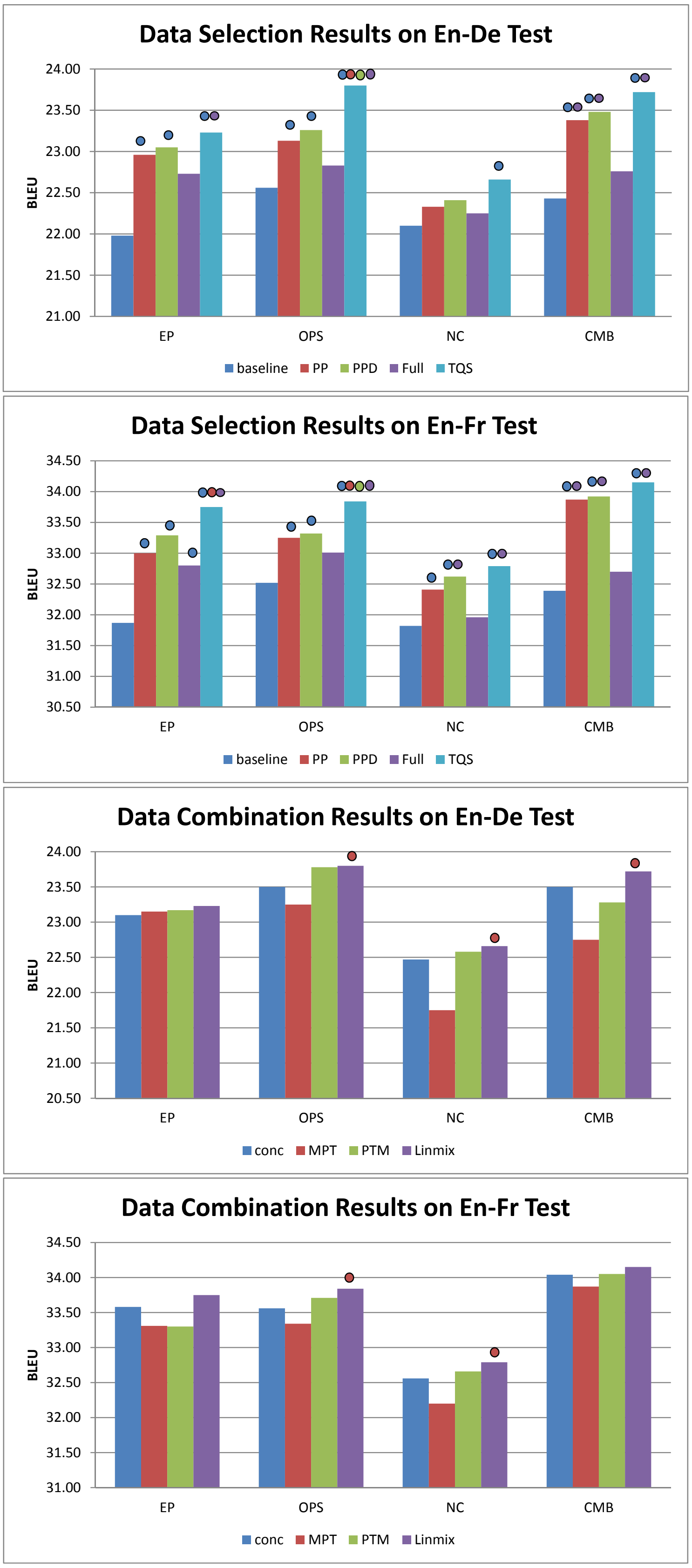
The second aspect of data-selection based domain adaptation approaches is the technique of combining the selected data to the existing in-domain dataset. Accordingly, we compared different combination techniques for combining the selected data with the in-domain data at the data level as well as the model level. For data-level combination we used the standard method of concatenating the in-domain and supplementary data together and training a new model on it (Conc). In addition, we used existing model-level combination techniques such as linear weighted interpolation (Linmix) and using multiple phrase-tables within the Moses decoder (MPT) in our experiments. Finally we used the phrase-table merging technique (PTM) to combine individual models trained on the supplementary and in-domain datasets. Based on the idea of weighted linear combination, this technique uses the frequency of the phrase pairs in the respective datasets to compute the weights used for the feature set combination. Our experiments in the previous section reveal that this technique in most cases provides nearly as good results as the others. Furthermore, this technique avoids the problem of large parameter settings of the MPT model as well as the issue of optimal weight estimation of the Linmix method. The set of graphs in Figure 6.1 show the relative variations of the different data selection and data combination methods for the four different supplementary datasets. Note that the graphs are drawn on the basis of the BLEU scores obtained by using Linmix as the combination method.

The relative variations of the BLEU scores in the graphs clearly highlight the superiority of the TQS method over other data selection methods for both language pairs and all supplementary datasets. The data combination graphs further indicate the minor variation in scores, with the PTM method performing nearly the same for most datasets and language pairs.

As previously mentioned in the introduction of the chapter, the TQS method depends on the devset to select batches of sentence pairs from the supplementary data. Since devsets are usually small in size, this technique may suffer from the issue of overfitting on the devset. This problem becomes quite acute when the devset is not representative of the target domain or the unseen testset. Hence ideally, this process should be carried out on multiple datasets, and only the intersection of the selected data from multiple runs be used for adaptation. For instances where in-domain data is sparse, we could resort to K-fold cross validation to address this issue. However, in our case, where parallel data from the target domain (forums) is not available at all, creating multiple devsets would be prohibitively expensive and time-consuming. Hence during the design of our devset, we ensured that it was representative of the target domain using a set of features described in Chapter 2 (cf. page 46).

6.6 Summary

In this chapter we have introduced a novel method of supplementary data selection for domain adaptation of SMT systems applied to the scenario of forum data trans-


Data Selection Results on En-De Test

Data Selection Results on En-Fr Test

Data Combination Results on En-De Test

Data Combination Results on En-Fr Test

Figure 6.1: Comparing BLEU scores generated by different data selection and combination techniques on the forum data.

lation. Sentence pairs are selected incrementally in batches from the supplementary bitext and added to the baseline system and evaluated in terms of BLEU scores on a development set. A batch is selected only if it results in improved BLEU scores. Once all the batches in a supplementary dataset are processed, the batches that pass the selection are accumulated to produce the selected parallel data for domain adaptation. When combined with the in-domain data, the data selected using this method is found to outperform other existing data selection methods in terms of translation quality on an unseen testset. This data selection method has the added advantage of automatically estimating the size of the selected supplementary data. This is in contrast to existing rank-based methods which require a threshold value for data selection.

In addition to the method of data selection, we also present a phrase-table merging technique that was developed to facilitate the iterative data selection. This technique is effectively used to combine multiple phrase-tables from different domains and performs on a par with other existing techniques in the field. The overall experiments using different data selection techniques in this chapter also confirm the effectiveness of relevant data selection (in contrast to using full datasets) for translating user-generated content using TM-based training data.

Finally we present the fourth and the final research question (**RQ4**), which forms the actual motivation for the techniques and experiments presented in this chapter:

> **RQ4:***How can translation quality be directly used to select relevant data from an out-of-domain corpus and effectively combine it with in-domain data to drive domain adaptation?*

The experimental findings in Section 6.4 and the subsequent observations in Section 6.5 strongly indicate that we have provided a conclusive answer to **RQ4** through the techniques introduced and experimented with in this chapter.

6.6.1 Contributions

The main contributions of this chapter are as follows:

- We have introduced a novel translation quality-based supplementary data selection technique for domain adaptation and successfully applied it to translate forum content using TM-based translation models.
- We have successfully shown our data selection technique to outperform existing methods in the literature.
- We have also introduced a new phrase-table merging technique based on frequency-based interpolation of feature values, which performs on a par with existing techniques.
- We have introduced an iterative framework for our data selection method using a unique scheme of batching similar sentence pairs.
- We have confirmed the importance of relevant data selection for domain adaptation of translation models by comparing the effect of different data selection methods to that of full data selection.

Chapter 7

Conclusion

In this thesis we have explored the effect of different domain adaptation techniques in Phrase-Based SMT (PBSMT) aimed at addressing two industrially relevant adaptation scenarios, specific to Symantec. Symantec being a global leader in security and storage and systems management, maintains a highly efficient and productive localisation workflow. Being an integral part of Symantec's localisation efforts, MT poses a series of challenges which have a high impact on the quality and efficiency of the process. Owing to the variety of products and services Symantec offers and needs to localise, domain adaptation of the MT systems presents itself as one of the most pertinent challenges. The first such adaptation scenario we address in this thesis concerns the translation of mixed-domain data in the presence of multiple domain-specific datasets.

While the first scenario captures one aspect of the domain adaptation spectrum, we investigate another aspect of the problem in the second scenario. The second scenario deals with handling translations for user-generated content in the absence of exact ''in-domain' training data. This is driven by the task of user-generated web-forum content translation for Symantec's Norton online forums. Web forums are rich sources of user-generated content, but are by nature monolingual. In absence of true 'in-domain' parallel training data, we utilise available parallel data from Symantec' internal documentation collection. Although the training and target domains are related (both are about Symantec products and services), there is a considerable difference in the nature of the two datasets. The training data is clean and quality controlled, conforming to internal controlled language guidelines, while the forum data is noisy, lightly moderated and somewhat irregular in terms of spelling and grammatical constructs. Moreover, the forum content, not being governed by any controlled language guidelines has a much wider and richer vocabulary compared to the training data. This stylistic and lexical difference between the training and target domain necessitates the use of domain adaptation approaches to facilitate good translation quality of the forums.

Faced with the task of addressing domain-adaptation problems for these two scenarios, we presented the background of the approaches and the specific research questions pertaining to each approach in Chapter 1 of this thesis. In Chapter 2, we presented a brief overview of the different paradigms in the MT literature focussing on the state-of-the-art PBSMT approach. The PBSMT paradigm is of primary interest in our work since all the SMT models used in this thesis are based on this approach. We presented a short description of the different components in a standard PBSMT setup, outlining the specific tools and software used in our experiments. Following this, we introduced the concept of domain adaptation in SMT which forms the central theme of this thesis. We reviewed existing approaches to domain adaptation reported in the literature both for generic NLP tasks as well as for SMT before aligning our research questions and scenarios to the state-of-the-art in domain adaptation research. In the following chapters of the thesis, we tackled the scenarios and the associated research questions one by one.

In Chapter 3, we presented our approach to tackle the first scenario of mixed-domain data translation. The primary motivation behind the experiments presented in this chapter was to develop a framework for translating mixed-domain data, such that it provided good quality translations, and at the same time was flexible to handle easy incorporation of new domains. We started by empirically confirming the assumption that PBSMT systems are particularly good at translating in-domain data, but the quality suffers when translating out-of-domain data (Haque et al., 2009). Based on this observation, we developed a technique to translate mixed-domain data by combining translations from two domain-specific systems. We used an automatic classifier to identify the domain of an input sentence and route it to the appropriate domain-specific model. In order to confirm the effectiveness of our approach, we compared it with existing techniques, such as data concatenation (Foster et al., 2010), model combination using the Moses decoder (Koehn and Schroeder, 2007) and pure system combination (Du et al., 2010). The experiments revealed that our classifier-based translation combination approach outperformed all the other techniques in terms of translation quality. Furthermore, by keeping the in-domain and out-of-domain SMT systems independent, our approach provided a flexible framework allowing easy addition or removal of domains.

The following three chapters (Chapters 4, 5 and 6) focussed on our efforts to address the second scenario of web forum data translation. The problem of forum-content translation, despite being a relevant one (Flournoy and Rueppel, 2010), has not received much attention in the SMT literature (Roturier and Bensadoun, 2011). The research presented in this thesis, concerning the domain-adaptation aspect of forum-content translation addressed this specific gap. One of the major challenges in translating forum content is the lack of parallel forum-style training data. In order to effectively use parallel corporate documentation to translate forum data, we started by quantifying the difference between the training data and the target domain in terms of OOV rates in Chapter 4. We investigated the nature of the OOV tokens generated on the forum content and classified them into four dominant categories. We developed different normalisation and data selection techniques to systematically reduce each category of OOVs, subsequently improving translation quality in every step. The novel technique of using domain-adapted automatic spell-checkers to address spelling errors was found to be particularly effective in this regard. Additionally we compared the individual effects of normalisation and data selection on the translation quality of forum data. The automatic evaluation results from our experiments, further supplemented by a full manual evaluation, showed the effectiveness of normalisation and data selection in translation quality improvement. We further showed that supplementary data selection alone was nearly as effective as using it in combination with normalisation for translating the posts with average noise density.

Observing the success of data selection as a domain adaptation technique in the current scenario, we investigated this technique further in Chapter 5. Using a LM perplexity-based data selection approach (Hildebrand et al., 2005), we focused on methods of combining in-domain and out-of-domain data together. Comparing the effect of corpus-level combination (concatenation of in-domain and out-of-domain data and training a single model on it), to more sophisticated model-level combination using linear or log-linear mixture adaptation (Hastie et al., 2001) we found that linear mixture adaptation was the best-performing combination method for the task. We also conducted a comparative analysis of the effect of component-level adaptation (translation model vs. language model) in our experiments which revealed that language model adaptation was more effective than translation model adaptation for the translation of forum content.

Finally in Chapter 6, motivated by the gap in the state-of-the-art techniques of data selection, we developed a novel translation quality-based data selection method. Developing a novel phrase-table merging technique to simulate the effect of full retraining, we selected batches of sentence-pairs from the supplementary dataset and added them to an existing baseline model in an iterative framework. A batch was finally selected only if it improved the translation quality over the baseline in terms of BLEU scores. In our experiments we compared this technique to existing methods of data selection in domain adaptation research to find that our technique provided better translation scores in the forum translation task. Additionally we also used the phrase-table merging technique as an alternative to existing techniques of data or model combination. The experimental results suggested that this technique performed on a par with most state-of-the-art techniques in literature (Koehn and

Schroeder, 2007; Foster and Kuhn, 2007).

At this point we revisit the research questions proposed in Chapter 1, clarifying how our experiments and findings addresses them in the individual chapters of this thesis.

> **(RQ1)** *Given a mixed domain and a set of mixed-domain training data, does a combination of translations from different domain-specific models, each trained on a subset of the data, provide better translation quality when compared to those from generic models, trained on the full dataset?*

In order to address **RQ1**, we developed an approach that combined translations from two domain-specific SMT systems using an automatic classifier. The classifier was used to identify the domain of the input sentence, which was then routed to the appropriate domain-specific SMT system for translation. To emulate the effect of a combined-data system on mixed-domain data translation, we combined domain-specific datasets from both domains and trained a single system on it. Furthermore, we used a sophisticated model combination technique using the multiple decoding path functionality of Moses (Koehn et al., 2007) to translate mixed-domain data. Since our approach was based on the combination of translations from multiple systems, we further compared it with a pure system combination approach based on confusion-network decoding (Mangu et al., 2000). Comparing the performance of our system to that of the other methods revealed the superiority of our approach in terms of automatic translation quality metrics. We further conducted a manual evaluation task comparing the translations produced by our method to those produced by the combined-data model which reaffirmed the superiority of the translations provided by the classifier-based combination approach in terms of fluency and adequacy. Our conclusions in Chapter 3 were thus able to comprehensively answer RQ1 in the affirmative.

> **(RQ2)** *In a scenario where the target domain is different from the training domain, how effective are normalisation and data selection methods in improving*

translation quality?

We used OOV-rate as a metric to quantify the difference between the training and target domains in response to **RQ2**. Classifying the OOVs observed on the target data with respect to the training data, into four broad categories, different normalisation and data selection methods were developed to tackle each category. Our experiments showed that both normalisation and data selection improved translation quality over the baseline in terms of automatic evaluation metric scores. We also presented a comparison of the effect of normalisation and data selection individually on test data with different noise densities. Our experiments demonstrated that for forum data with generic noise density, data selection was often as effective as normalisation and data selection put together. Additionally, we also conducted a manual evaluation comparing the effect of normalisation and normalisation with data selection on translation quality of forum content. The findings from manual evaluation further confirmed our observations based on automatic evaluation metrics about the effectiveness of normalisation and data selection in forum content translation task. The translation quality improvements provided by the normalisation and data selection methods in our experiments thus addressed the research question **RQ2**.

> **(RQ3)***How can multiple models be adapted at different component levels of an SMT system, and what is the effect of component-level adaptation on translation quality?*

In our efforts to answer **RQ3**, we compared the effect of corpus-level combination to that of model-level combination in a data selection setting in Chapter 5. Using mixture model adaptation to combine SMT component-models trained on in-domain and supplementary data, our experiments showed that linear mixture adaptation was the most effective technique in model combination. While this finding partially answered RQ3, we further conducted experiments to compare the effect of translation model adaptation with that of language model adaptation on the translation quality of forum data. The findings from these sets of experiments demonstrated the effect of language modelling adaptation to be more profound compared to that of translation model adaptation, thus providing a complete answer to both parts of the research question **RQ3**.

> **(RQ4)***How can translation quality be directly used to select relevant data from out-of-domain corpora and effectively combine it with in-domain data to drive domain adaptation?*

Finally we tackled **RQ4** by developing a novel supplementary data selection method which utilised translation quality directly to measure the 'fitness' of a set of sentence-pairs from a supplementary dataset. Compared with existing data selection methods in the literature, our approach provided better translation quality in terms of automatic evaluation scores. Furthermore, in order to facilitate the data selection in an iterative framework, we developed a phrase-table merging technique which simulated the effect of incremental retraining for the translation models. This method was effectively used in our experiments as an alternative to standard model combination techniques in the literature. The experimental results showed its performance to be on par with state-of-the-art techniques. The improvement in translation quality derived from the techniques presented in Chapter 6 provide a conclusive answer to RQ4.

7.1 Contribution

In summary, we have experimented with different domain adaptation techniques customised to our specific scenarios and research objectives and have presented a novel domain adaptation framework based on supplementary data selection approaches. As a part of this dissertation, we have made the following overall contributions:

- We have developed a classifier-based combination of domain-specific SMT systems to effectively translate mixed-domain data. We have compared the effect of this method with standard methods of corpus-level concatenation as well as state-of-the-art methods of model or system combination, and have found our approach to work better than all of the previous methods.

- We have successfully used different normalisation and data selection methods guided by OOV reduction to improve the quality of web-forum data translation. In addition we have presented a novel study on the effect of adapted automatic spell-checking on translation quality. Comparing the effect of normalisation and data selection we have shown that normalisation is really effective for noisy forum data translation and supplementary data selection alone works nearly as well as normalisation for the generic forum data.

- We have explored different combination techniques both at the corpus-level as well as at the model-level and have conclusively shown model-level combination to be more effective in terms of translation quality improvement. Furthermore, comparing the effect of translation model adaptation to that of language model adaptation, we have shown language model adaptation to be more effective in improving translation quality of forum content.

- We have proposed a new method of supplementary data selection based on actual translation quality measured by automatic evaluation metrics. We have shown through our experiments that this technique outperforms most of the existing techniques in data selection. Moreover, we have presented a phrase-table merging mechanism which can be used to simulate incremental training of translation models. Using this technique as an alternative method to model combination, we compared it with state-of-the-art methods in the literature and showed that it compares on a par with existing methods.

7.2 Future Work

In this thesis we have presented a number of domain adaptation approaches to SMT, in two distinct industrially relevant scenarios. Throughout the chapters we have provided a conclusive and comprehensive account of the experiments aimed at addressing specific research questions. However, we have identified a number of future research directions which we believe would require further exploration.

In Chapter 3, we address the scenario of mixed-domain data translation using two predefined domains based on Symantec product lines (cf. Chapter 2, Section 2.5.2). Instead of relying on predefined or manually defined domains, unsupervised clustering methods could be used to automatically identify latent domains in mixed-domain training data. While some work on clustering the training data using perplexity (Sennrich, 2012a) or language model entropy (Yamamoto and Sumita, 2008) have already been reported, this could be extended to use bilingual features (like the ones extracted in the Moses phrase table) to effectively split heterogeneous training data into useful sub-parts. Domain-specific models trained on each such sub-part could then be combined using our classifier-based technique for better translation quality.

In Chapter 4, we present a classification of the list of OOVs based on their generic characteristics in the target domain (cf. Chapter 4, page 97). While in our experiments, this classification has been done in a semi-manual fashion, this data could be used as annotated training data to train a classifier which would automatically identify the category of an OOV and apply the optimal normalisation method. Secondly, while using a spell-checker in our experiments we have only considered the first suggestion from the spell-checker (cf. Chapter 4, page 101). While, this can handle most of the spelling errors (depending upon the accuracy of spell-checkers) fairly well, further experiments considering the other options provided by the spell-checker could provide further improvements in translation quality.

Finally in Chapter 6, we used a somewhat *adhoc* batching scheme and batch size to handle the scaling issue of our approach (cf. Chapter 6, page 167). A deeper investigation into the effect of different batch sizes and batching schemes is necessary to further optimise the solution. Further work is required on deeper analysis of how different datasets contribute to the translation quality improvement in a data selection-based adaptation setting. Furthermore, considering the impact of language model adaptation in translation of forum content, extending the translation quality-based data selection method for language modelling will also be of considerable interest.

Appendix A

Manual Evaluation Guidelines for English to Simplified Chinese Translations

Objective A human evaluation on the output of different machine translation systems is being conducted to compare the performance of different Machine Translation techniques on translation quality. We are thankful to all the evaluators for their expertise, time and effort towards the successful completion of the task.

Task

- Language Direction: English to Chinese
- Type of Content: Enterprise Content from Symantec
- Number of Sentences to Evaluate: 100 sentences
- Reference Translations: Provided

Guidelines You will be provided with an excel sheet containing a table with 4 sentences in each cell. These include :

1. The source sentence in English
2. The reference translation in simplified Chinese
3. Translation output from System-1 (Sys-1)
4. Translation output from System-2 (Sys-2)

Based on your judgement you would need to rate (details of the ratings below) the translations according to their Fluency and Adequacy. For each of the two translations (Sys-1 and Sys-2) you would need to put in a value from the list of ratings mentioned below in the appropriate column. We also request you to put in a reason to substantiate your scoring for each translation.

Ratings **Fluency**: A 5 point scale indicates how fluent the translation is. When translating English to Chinese the values correspond to:

- 4 = Flawless Chinese
- 3 = Good Chinese
- 2 = Non-native Chinese
- 1 = Disfluent Chinese
- 0 = Incomprehensible

Please score a translation with the rating which you believe reflects the fluency of the translation.

Adequacy: A 5 point scale to indicate how much the meaning of the source sentence is also expressed in the translation.

- 4 = All
- 3 = Most
- 2 = Much
- 1 = Little
- 0 = None

Please score a translation with the rating which you believe reflects the adequacy of the meaning expressed in each translation.

Reason: For each translation rating you will also need to substantiate your scoring with a reason for the high/low scores. If translations from a system is better than the other then use any reasons from the following lists to substantiate your ratings.

- 1. Better word Order
- 2. Better Phrase/Word Selection
- 3. Less OOV words[1]

[1] OOV stands for Out-of-Vocabulary word. It is often the case that machine translation systems cannot translate individual words/phrases in the source sentences as these words/phrases are not present in the database of the system. Such words are referred to as OOV words.

- 4. Others (Please specify)

IMP: If you choose Others, please elaborate the specific reason you think a particular translation is better as per your ratings. In case of similar translations (two translations are exactly same) just mention same as the reason while rating the 2 translations similarly.

Worked out Example Below is an example of the table which is presented for evaluation. You would be required to fill each box under Fluency (Fl) and Adequacy (Ad) and provide reason for your ratings. Src refers to the source sentence while Trg refers to the reference translation.

The example here shows how the table should look like after the evaluation is complete. *Note that the evaluation ratings for fluency and adequacy and the reasons are arbitrary and just used to show usage*:

Type	**Sentence**	**Fl**	**Ad**	**Reason**
Src	the backup job will then become functional in & productnamefull ; and continue to apply password protection to recovery points .			
Trg	然后 ， 备份 作业 将 在 & productnamefull ; 中 正常 运行 ， 并 继续 对 恢复 点 应用 密码 保护 。			
Sys-1	然后 ， 备份 作业 将 在 & productnamefull ; 中 正常 运行 ， 并 继续 应用 到 恢复 点 密码 保护 。	2	3	
Sys-2	然后 ， 备份 作业 将 在 & productnamefull ; 中 正常 运行 ， 并 继续 应用 密码 保护 到 恢复 点 。	4	4	1,2

Things to Note:

- The English input and the Chinese translation outputs from all systems are tokenized and lowercased. This is a normal behaviour of the systems and should NOT be penalized in terms of translation quality rating.

- If a translation has a untranslated word and the translation of the same word is NOT present in the reference translation, it should NOT be treated as a OOV word in the translation. E.g. *&productnamefull* is not an OOV since the reference translation also has the same.

- There is a possibility of minor errors in the reference translations. In such cases please ignore the errors and do NOT penalize if the translation correct the error.

Appendix B

Regular Expressions used for handling MASK tokens

1. Regular expressions used to handle URLs (used in specific order in which they appear).

   (a) $s/(http|https|ftp):\backslash/\backslash//\backslash 1:\backslash/\backslash//ig$

   (b) $s/(http|https|ftp):\backslash/\backslash/(www\backslash.)?[A-Za-z0-9\backslash.\backslash\& =\sim \#\%\$\backslash * : \_\backslash?\backslash - \backslash/\backslash\backslash|!'\backslash+;\backslash(\backslash)]+/<url\_ph>/ig$

   (c) $s/()?www\backslash.[A-Za-z0-9\backslash.\backslash\& =\sim \#\%\$\backslash * : \_\backslash?\backslash-\backslash/\backslash\backslash|!'\backslash+;\backslash(\backslash)]+/\$1<url\_ph>/ig$

2. Regular Expression for handling email-ids.

   (a) $s/[A-Za-z0-9\_\backslash-\backslash.]+@([A-Za-z0-9\_\backslash-]+\backslash.)+\backslash w2,4/<email-id>/g$

3. Regular expressions used to handle IP Addresses or version numbers (used in same order as order of appearance).

   (a) $s/([\hat{}0-9\backslash.])(([0-9]|[0-9][0-9]|[01][0-9]2|2[0-4][0-9]|25[0-5])\backslash.)3([0-9]|[0-9][0-9]|[01][0-9]2|2[0-4][0-9]|25[0-5])([\hat{}0-9\backslash.])/\$1<ip\_ver>\$5/g$

   (b) $s/([\hat{}0-9\backslash.])(([0-9]|[0-9][0-9]|[01][0-9]2|2[0-4][0-9]|25[0-5])\backslash.)3([0-9]|[0-9][0-9]|[01][0-9]2|2[0-4][0-9]|25[0-5])([\hat{}0-9])(\backslash s+)?/\$1<ip\_ver>\$5/g$

   (c) $s/\hat{}(([0-9]|[0-9][0-9]|[01][0-9]2|2[0-4][0-9]|25[0-5])\backslash.)3([0-9]|[0-9][0-9]|[01][0-9]2|2[0-4][0-9]|25[0-5])([\hat{}0-9\backslash.])/<ip\_ver>\$4/g$

4. Regular expressions used for dates

   (a) $s/([\hat{}0-9])(19[0-9][0-9]|20[0-9][0-9]|[0-9][0-9])([\backslash-\backslash/\backslash\backslash\backslash.])([1-9]|0[1-9]|1[012])\backslash 3([1-9]|0[1-9]|[12][0-9]|3[01])([\hat{}0-9])/\$1<date>\$6/g$ (yyyy/mm/dd format)

(b) `s/([^0−9])([1−9]|0[1−9]|[12][0−9]|3[01])([\−\/\\\.])([1−9]|0[1−9]|1[012])\3(19[0 − 9][0 − 9]|20[0 − 9][0 − 9]|[0 − 9][0 − 9])([^0 − 9])/$1 < date > $6/g` (dd/mm/yyyy format)

(c) `s/([^0−9])([1−9]|0[1−9]|1[012])([\−\/\\\.])([1−9]|0[1−9]|[12][0−9]|3[01])\3(19[0 − 9][0 − 9]|20[0 − 9][0 − 9]|[0 − 9][0 − 9])([^0 − 9])/$1 < date > $6/g` (mm/dd/yyyy format)

5. Regular Expression for Windows Registry Entries

(a) `s/^(\s+)?(HKUS | HKCU | HKLM | HKCR | HKEY_LOCAL_MACHINE(S)? | HKEY_CLASSES_ROOT | HKEY_CURRENT_USER | HKEY_ALL_USERS | HKEY_USERS)[\\\/]([^, : ”\∗#@\?!\.\\\/]+[\\\/])∗[^\)”; :] + /$1 < winreg > /gi`

6. Regular Expression for Windows Path Entries

(a) `s/([^a−zA−Z0−9])[a−zA−Z0−9] : ([\\\/])([^, : ”\∗#@?!\\\/]+\2)+[^\)”; : , \?] + /$1 < winpath > /g`

(b) `s/^[a − zA − Z0 − 9] : ([\\\/])([^, : ”\ ∗ #@\?! <> \.\\\/] + \1) + [^\)”; : \?] + / < winpath > /g`

(c) `s/([′”])[a − zA − Z0 − 9] : \\[^′”] + [′”]/$1 < winpath > $1/g`

(d) `s/\w : \\[a − zA − Z0 − 9_\−] + \.\w3/ < winpath > /g`

(e) `s/\w : \/[A − Za − z0 − 9_\−] + \.\w3/ < winpath > /g`

Appendix C

Manual Evaluation Guidelines for English to German/French Translations

Objective A human evaluation on the output of different machine translation systems is being conducted to compare the performance of different Machine Translation techniques on translation quality. The task involves Statistical Machine Translation (SMT) of User-generated forum content from Symantec web-forums using different adapted SMT systems. We are thankful to all the evaluators for their expertise, time and effort towards the successful completion of the task.

Task

- Language Direction: English to German/French
- Type of Content: Forum content from Symantec Online web forums
- Number of Sentences to Evaluate: 100 sentences
- Reference Translations: Provided

Guidelines You will be provided with an excel sheet containing a table with 5 sentences in each block. These include :

1. The source sentence in English
2. The reference translation in German/French
3. Translation out from System-1 (Sys-1)
4. Translation output from System-2 (Sys-2)
5. Translation output from System-3 (Sys-3)

Based on your judgement you would need to rate (details of the ratings below) the translations according to their Fluency and Adequacy. For each of the three translations (Sys-1, Sys-2 and Sys-3) you would need to put in a value from the list of ratings mentioned below in the appropriate column. We also request you to put in a reason to substantiate your scoring for each translation.

Ratings **Fluency**: A 5 point scale indicates how fluent the translation is. When translating English to German/French the values correspond to:

- 4 = Flawless German/French
- 3 = Good German/French
- 2 = Non-native German/French
- 1 = Disfluent German/French
- 0 = Incomprehensible

Please score a translation with the rating which you believe reflects the fluency of the translation.

Adequacy: A 5 point scale to indicate how much the meaning of the source sentence is also expressed in the translation.

- 4 = All
- 3 = Most
- 2 = Much
- 1 = Little
- 0 = None

Please score a translation with the rating which you believe reflects the adequacy of the meaning expressed in each translation.

Reason: For each translation rating you will also need to substantiate your scoring with a reason for the high/low scores. The requirement of this task is to compare the three translations hence you would require to follow the following steps:

1. Compare Sys-2 translations with Sys-1 and update the reason (selected from the list below) into the reason box next to Sys-2.

2. Now Compare Sys-3 translations with Sys-2 and update the reason from the list below into the box next to Sys-3

3. Use any of the reasons from the table below to mark the reasons.

Reasons for Better Translation	**Reasons for Poor Translation**
B1: Better translation of OOV words	P1: Poorer translation of OOV words
B2: Better word order	P2: Poor word order
B3: Better word/phrase selection	P3: Poor word/phrase selection
B4: Others	P4: Others

IMPORTANT: READ THE FOLLOWING FOR MARKING THE REASONS COLUMN

1. Better/Poorer translation of OOV words(B1/P1): An SMT system might not be able to translate all source words in a sentence. Such words are usually left as it is by the SMT system. These are termed as OOV words. Better translation of OOV words could be due to the following reasons:

   (a) Better handling or URLs, Computer File Path Entries, Windows Registry entries, date and time entries etc. *NOTE: Better handling might sometimes mean no translations. For Example, if the source sentence has the token "c: / / my documents / vacation pictures / london" and the reference sentence also has the same token, then the correct translation would mean the sub-parts of the tokens like "my documents" or "vacation pictures" should NOT be translated. If a system translation has translation in sub-parts of the path element, then it is not a correct translation.*

   (b) Better Translation due to spelling error corrections: The source might have spelling errors, which some translation might be able to handle while the others may not.

   (c) Better Translation due to Fused words: Sometimes two words in the source may be fused using a '.' symbol. One translation might handle this and provide separate translations for each word, while others may not.

   (d) Better Translation of any generic English word: If some translations have English words in them and others provide their German/French counterparts, then the latter better handles translation.

2. Better/Poorer word Order (B2/P2): Especially if the ordering happens within tokens like URLs, Path entries, dates, Windows Registry entries

3. Better Phrase/Word Selection (B3/P3)

4. Others (Please specify) (B4/P4)

General Instructions

- If you choose Others, please elaborate the specific reason you think a particular translation is better as per your ratings.

- Please try to use the numbers instead of the actual text in the reasons column. Like B1.1 for better URL/path/registry key handling or B3 for better phrase selection. Or use P1.2 for poorer translation due to spelling corrections etc. For Others(P4/B4) however mention the reason.

- In case of 2 translations being similar, just mention 0 as the reason while rating translations accordingly

Worked out Example Below is an example of the table which is presented for evaluation. You would be required to fill each box under Fluency (Fl) and Adequacy (Ad) and provide reason for your ratings. Src refers to the source sentence while Trg refers to the reference translation.

Here is an example of how the table should look like after the evaluation is complete. *Note that the evaluation ratings for fluency and adequacy and the reasons are arbitrary and just used to show usage:*

Things to Note:

- The English input and the German/French translation outputs from all systems are tokenized and lowercased. This is a normal behaviour of the systems and should NOT be penalized in terms of translation quality rating.

Type	Sentence	Fl	Ad	Reason
Src	5 . click on the folder button and navigate to c : \documents and settings \all users \application data \and select the carbonite folder			
Trg	5. klicken sie auf die ordnerschaltfläche und öffnen sie den ordner “ c : \documents and settings \all users \application data \carbonite ”			
Sys-1	5. klicken sie auf den ordner “ und navigieren sie zu c : \dokumente und einstellungen \alle benutzer \anwendungsdaten \und wählen sie die carbonite ordner	2	3	
Sys-2	5. klicken sie auf die schaltfläche “ und wechseln sie zum ordner c : \documents and settings \all users \application data \carbonite und wählen sie die carbonite ordners	3	4	B1.1
Sys-3	klicken sie auf die schaltfläche “ und wechseln sie zum ordner c : \documents and settings \all users \application data \carbonite und wählen sie die carbonite ordner (vista benutzer wählen sie c : \programdata \und wählen sie die carbonite ordner	2	4	P2
Src	re : nis09 did not detect 8 threats & 23 infected objects.and 16 suspicious objects ?			
Trg	re : nis09 n’ a pas détecté 8 menaces , 23 objets infectés et 16 objets suspects ?			
Sys-1	re : nis09 n’ a pas détecter 8 menaces et 23 infecté objects.and 16 les objets ?	2	3	
Sys-2	nis09 n’ a pas détecter 8 menaces et 23 infecté objets . et 16 les objets ?	3	4	B1.3
Sys-3	re : nis09 n’ a pas détecter 8 menaces et 23 infecté objets . et 16 des objets ?	4	4	B3

- If a translation contains an untranslated word (English), please refer to the reference translation. If the reference translation also has the same word, then that is proper translation and should not be penalised.

- Do not penalize German/French translation for using English conventions for decimal points in numbers or comma as thousand separators or English data conventions. The reference translation is to be considered as correct in this regard.

Bibliography

Axelrod, A., He, X., and Gao, J. (2011). Domain adaptation via pseudo in-domain data selection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 355–362, Edinburgh, United Kingdom.

Axelrod, A. E. (2006). Factored Language Models for Statistical Machine Translation. Master's thesis, University of Edinburgh.

Bach, N. (2012). *Dependency Structures for Statistical Machine Translation.* PhD thesis, LTI, Carnegie Mellon University.

Banerjee, P., AlMaghout, H., Naskar, S. K., Roturier, J., Jiang, J., Way, A., and van Genabith, J. (2011a). The DCU Machine Translation Systems for IWSLT 2011. In *Proceedings of International Workshop on Spoken Language Translation*, pages 254–260, San Francisco, CA.

Banerjee, P., Li, B., Naskar, S. K., Way, A., and Van Genabith, J. (2010). Combining Multi-domain Statistical Machine Translation Models using Automatic Classifiers. In *AMTA 2010: Proceedings of the Ninth Conference of the Association for Machine Translation in the Americas*, pages 141–150, Denver, CO.

Banerjee, P., Naskar, S. K., Roturier, J., Way, A., and van Genabith, J. (2011b). Domain Adaptation in Statistical Machine Translation of User-Forum Data using Component Level Mixture Modelling. In *Proceedings of the Thirteenth Machine Translation Summit*, pages 285–292, Xiamen, China.

Banerjee, P., Naskar, S. K., Roturier, J., Way, A., and van Genabith, J. (2012a).

Domain Adaptation in SMT of User-Generated Forum Content Guided by OOV Word Reduction: Normalization andor Supplementary Data? In *Proceedings of the 16th Annual Conference of the European Association for Machine Translation (EAMT-2012)*, pages 169–176, Trento, Italy.

Banerjee, P., Naskar, S. K., Roturier, J., Way, A., and van Genabith, J. (2012b). Translation Quality-Based Supplementary Data Selection by Incremental Update of Translation Models. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING-2012)*, page To Appear, Mumbai, India.

Banerjee, S. and Lavie, A. (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, MI.

Barrault, L. (2010). MANY : Open Source Machine Translation System Combination. *Prague Bulletin of Mathematical Linguistics, Special Issue on Open Source Tools for Machine Translation*, (93):145–155.

Bertoldi, N. and Federico, M. (2009). Domain adaptation for statistical machine translation with monolingual resources. In *EACL 2009: Fourth Workshop on Statistical Machine Translation, Proceedings of the Workshop*, pages 182–189, Athens, Greece.

Biber, D. (1988). *Variation across speech and writing.* Cambridge University Press, Cambridge.

Bisazza, A., Ruiz, N., and Federico, M. (2011). Fill-up versus Interpolation Methods for Phrase-based SMT Adaptation. In *Proceedings of International Workshop on Spoken Language Translation*, pages 136–143, San Francisco, CA.

Blitzer, J., McDonald, R., and Pereira, F. (2006). Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 120–128.

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)*, pages 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*, 19:263–311.

Bulyko, I., Matsoukas, S., Schwartz, R., and Makhoul, J. (2007). Language Model Adaptation in Machine Translation from Speech. In *Proceedings of the 32nd International Conference on Acoustics, Speech, and Signal Processing (ICASSP-2007)*, pages 117–120, Honolulu, HI.

Callison-Burch, C., Bannard, C., and Schroeder, J. (2005). A Compact Data Structure for Searchable Translation Memories. In *Proceedings of 10th Annual Conference of European Association for Machine Translation (EAMT-2005)*, pages 59–65, Budapest, Hungary.

Callison-Burch, C., Koehn, P., Monz, C., Peterson, K., Przybocki, M., and Zaidan, O. F. (2010). Findings of the 2010 Joint Workshop on Statistical Machine Translation and Metrics for Machine Translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, WMT '10, pages 17–53, Uppsala, Sweden.

Cappé, O. and Moulines, E. (2009). On-line expectation-maximization algorithm for latent data models. *Journal of the Royal Statistical Society Series B*, 71(3):593–613.

Carter, D. (1994). Improving language models by clustering training sentences. In *Proceedings of the fourth conference on Applied natural language processing*, ANLC '94, pages 59–64, Stuttgart, Germany.

Caruana, R. (1997). Multitask Learning. *Machine Learning*, 28(1):41–75.

Chang, M.-W., Connor, M., and Roth, D. (2010). The Necessity of Combining Adaptation Methods. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 767–777, Cambridge, MA. Association for Computational Linguistics.

Chapelle, O., Schölkopf, B., and Alexander, Z., editors (2006). *Semi-Supervised Learning*. MIT Press.

Chelba, C. and Acero, A. (2006). Adaptation of maximum entropy capitalizer: Little data can help a lot. *Computer Speech & Language*, 20(4):382–399.

Chen, S. F. and Goodman, J. (1996). An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, ACL '96, pages 310–318.

Chiang, D., Knight, K., and Wang, W. (2009). 11,001 new features for statistical machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 218–226, Boulder, CO.

Chickering, D. M., Heckerman, D., and Meek, C. (1997). A Bayesian approach to learning Bayesian networks with local structure. In *In Proceedings of Thirteenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann.

Civera, J. and Juan, A. (2007). Domain adaptation in statistical machine translation with mixture modelling. In *ACL 2007: Proceedings of the Second Workshop on Statistical Machine Translation*, pages 177–180, Prague, Czech Republic.

Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., and Singer, Y. (2006). Online Passive-Aggressive Algorithms. *Journal of Machine Learning Research*, 7:551–585.

Daumé, III, H., Kumar, A., and Saha, A. (2010). Frustratingly easy semi-supervised domain adaptation. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, DANLP 2010, pages 53–59.

Daumé, III, H. and Marcu, D. (2006). Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26(1):101–126.

Daume III, H. (2007). Frustratingly Easy Domain Adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic.

Daume III, H. and Jagarlamudi, J. (2011). Domain Adaptation for Machine Translation by Mining Unseen Words. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 407–412, Portland, OR.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, B*, 39:1–38.

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)*, pages 128–132, San Diego, CA.

Doherty, S. (2012). *Investigating the effects of controlled language on the reading and comprehension of machine translated texts: A mixed-methods approach.* PhD thesis, Dublin City University.

Du, J., He, Y., Penkale, S., and Way, A. (2009). MaTrEx: the DCU MT system for WMT 2009. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 95–99, Athens, Greece.

Du, J., Pecina, P., and Way, A. (2010). An augmented three-pass system combination framework: DCU combination system for WMT 2010. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, WMT '10, pages 290–295, Uppsala, Sweden.

Dumais, S., Platt, J., Heckerman, D., and Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. In *Proceedings of the seventh international conference on Information and knowledge management*, CIKM '98, pages 148–155, Bethesda, MD.

Eck, M., Vogel, S., and Waibel, A. (2004). Language model adaptation for statistical machine translation based on information retrieval. In *Proceedings of 4th International Conference on Language Resources and Evaluation, (LREC 2004)*, pages 327–330, Lisbon, Portugal.

Federico, M., Bertoldi, N., and Cettolo, M. (2008). IRSTLM: an open source toolkit for handling large scale language models. In *Interspeech 2008: 9th Annual Conference of the International Speech Communication Association*, pages 1618–1621, Brisbane, Australia.

Finch, A. and Sumita, E. (2008). Dynamic model interpolation for statistical machine translation. In *ACL-08: HLT: Third Workshop on Statistical Machine Translation, Proceedings of the Workshop*, pages 208–215, Columbus, OH.

Finkel, J. R. and Manning, C. D. (2009). Hierarchical Bayesian domain adaptation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 602–610, Boulder, CO.

Fleiss, J. L. (1971). Measuring Nominal Scale Agreement among Many Raters. *Psychological Bulletin*, **76**(5):378–382.

Flournoy, R. and Callison-Burch, C. (2000). Reconciling User Expectations and Translation Technology to Create a Useful Real-world Application. In *Proceedings of the 22nd International Conference on Translating and the Computer*, London, United Kingdom.

Flournoy, R. and Rueppel, J. (2010). One Technology : Many Solutions. In *AMTA 2010: Proceedings of the Ninth Conference of the Association for Machine Translation in the Americas*, pages 6–12, Denver, CO.

Foster, G., Goutte, C., and Kuhn, R. (2010). Discriminative instance weighting for domain adaptation in statistical machine translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 451–459, Cambridge, MA.

Foster, G. and Kuhn, R. (2007). Mixture-model adaptation for SMT. In *ACL 2007: Proceedings of the Second Workshop on Statistical Machine Translation*, pages 128–135, Prague, Czech Republic.

Gahbiche-Braham, S., Bonneau-Maynard, H., and Yvon, F. (2011). Two Ways to Use a Noisy Parallel News Corpus for Improving Statistical Machine Translation. In *Proceedings of the 4th Workshop on Building and Using Comparable Corpora: Comparable Corpora and the Web*, pages 44–51, Portland, Oregon. Association for Computational Linguistics.

Gao, Q., Lewis, W., Quirk, C., and Hwang, M.-Y. (2011). Incremental Training and Intentional Over-fitting of Word Alignment. In *Proceedings of the Thirteenth Machine Translation Summit*, pages 106–113, Xiamen, China.

Germann, U., Jahr, M., Knight, K., Marcu, D., and Yamada, K. (2001). Fast decoding and optimal decoding for machine translation. In *Proceedings of the*

39th Annual Meeting on Association for Computational Linguistics, ACL '01, pages 228–235.

Gildea, D. (2001). Corpus Variation and Parser Performance. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pages 167–202, Pittsburgh, PA.

Good, I. J. (1965). *The Estimation of Probabilities: An Essay on Modern Bayesian Methods*, volume 30 of *Research Monograph.* M.I.T. Press, Cambridge, MA.

Habash, N. (2008). Four techniques for online handling of out-of-vocabulary words in Arabic-English statistical machine translation. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 57–60, Columbus, OH.

Haque, R., Naskar, S. K., Van Genabith, J., and Way, A. (2009). Experiments on Domain Adaptation for English–Hindi SMT. In *Proceedings of PACLIC 23: the 23rd Pacific Asia Conference on Language, Information and Computation*, pages 670–677, Hong Kong.

Hara, T., Miyao, Y., and Tsujii, J. (2005). Adapting a probabilistic disambiguation model of an HPSG parser to a new domain. In *Proceedings of the Second international joint conference on Natural Language Processing*, IJCNLP'05, pages 199–210.

Hardt, D. and Elming, J. (2010). Incremental Re-training for Post-editing SMT. In *Proceedings of the Ninth Conference of the Association for Machine Translation in the Americas (AMTA-2010)*, Denver, CO.

Hasan, S. and Ney, H. (2005). Clustered language models based on regular expressions for SMT. In *10^{th} EAMT Conference: Practical Applications of Machine Translation, Conference Proceedings*, pages 133–142, Budapest, Hungary.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY.

Heafield, K. (2011). KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, United Kingdom.

Heckman, J. J. (1979). Sample Selection Bias as a Specification Error. *Econometrica*, 47(1):153–61.

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 *10^{th} EAMT Conference: Practical Applications of Machine Translation, Conference Proceedings*, pages 119–125, Budapest, Hungary.

Iyer, R. and Ostendorf, M. (1999). Modeling long distance dependence in language: topic mixtures versus dynamic cache models. *IEEE Transactions on Speech and Audio Processing*, 7:30–39.

Iyer, R., Ostendorf, M., and Gish, H. (1997). Using out-of-domain data to improve in-domain language models. *Signal Processing Letters, IEEE*, 4(8):221 –223.

Japkowicz, N. and Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5):429–449.

Jelinek, F. and Mercer, R. L. (1980). Interpolated estimation of Markov source parameters from sparse data. In *In Proceedings of the Workshop on Pattern Recognition in Practice*, pages 381–397, Amsterdam, The Netherlands.

Jiang, J. and Zhai, C. (2007). Instance Weighting for Domain Adaptation in NLP. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 264–271, Prague, Czech Republic.

Joachims, T. (1999). Making Large-Scale SVM Learning Practical. In Schölkopf, B., Burges, C., and Smola, A., editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press, Cambridge, MA, USA.

Kneser, R. and Ney, H. (1995). Improved backing-off for m-gram language modeling. In *In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume I, pages 181–184, Detroit, MI.

Kneser, R. and Steinbiss, V. (1993). On the dynamic adaptation of stochastic language models. In *Proceedings of the 1993 IEEE international conference on Acoustics, speech, and signal processing: speech processing - Volume II*, ICASSP'93, pages 586–589, Minneapolis, MN.

Koehn, P. (2004). Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, (EMNLP 2004)*, pages 388–395, Barcelona, Spain.

Koehn, P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. In *MT Summit X: The 10th Machine Translation Summit*, pages 79–86, Phuket, Thailand.

Koehn, P. (2010). *Statistical Machine Translation.* Cambridge University Press, Cambridge, United Kingdom.

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 *ACL 2007, Proceedings of the Interactive Poster and Demonstration Sessions*, pages 177–180, Prague, Czech Republic.

Koehn, P. and Knight, K. (2001). Knowledge Sources for Word-Level Translation Models. In Lee, L. and Harman, D., editors, *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pages 27–35, Pittsburgh, PA.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *HLT-NAACL 2003: Conference Combining Human Language Technology Conference Series and the North American Chapter of the Association for Computational Linguistics conference series*, pages 48–54, Edmonton, Canada.

Koehn, P. and Schroeder, J. (2007). Experiments in domain adaptation for statistical machine translation. In *ACL 2007: Proceedings of the Second Workshop on Statistical Machine Translation*, pages 224–227, Prague, Czech Republic.

Kuhn, R. and De Mori, R. (1990). A Cache-Based Natural Language Model for Speech Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence - PAMI*, 12:570–583.

Kumar, S. and Byrne, W. (2004). Minimum Bayes-Risk Decoding for Statistical Machine Translation. In *Proceedings of the Human Language Technology Conference / North American Chapter of the Association of Computational Linguistics Annual Meeting (HLT-NAACL-04)*, pages 169–176, Boston, MA.

Kwok, J. T. (1998). Automated Text Categorization Using Support Vector Machine. In *In Proceedings of the International Conference on Neural Information Processing (ICONIP)*, pages 347–351, Kitakyushu, Japan.

Landis, J. R. and Koch, G. C. (1977). The Measurement of Observer Agreement for Categorical Data. *Biometrics*, **33**:159–174.

Langlais, P. (2002). Improving a general-purpose Statistical Translation Engine by terminological lexicons. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, pages 1–7, Taipei, Taiwan.

Lavergne, T., Le, H.-S., Allauzen, A., and Yvon, F. (2011). LIMSI's experiments in domain adaptation for IWSLT11. In *Proceedings of the eigth International Workshop on Spoken Language Translation (IWSLT)*, pages 62–67, San Francisco, CA.

Lavie, A. and Agarwal, A. (2007). Meteor: an automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, StatMT '07, pages 228–231.

LDC (2002). Linguistic Data Annotation Specification: Assessment of Fluency and Adequacy in Chinese-English Translations. Technical Report 1.0, Linguistic Data Consortium.

Lease, M., Charniak, E., Johnson, M., and McClosky, D. (2006). A look at parsing and its applications. In *proceedings of the 21st national conference on Artificial intelligence - Volume 2*, AAAI'06, pages 1642–1645, Boston, MA.

Lee, D. Y. (2001). Genres, registers, text types, domains and styles: clarifying the concepts and navigating a path through the BNC jungl. *Language Learning and Technology*, 5(3):37–72.

Levenberg, A., Callison-Burch, C., and Osborne, M. (2010). Stream-based translation models for statistical machine translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 394–402, Los Angeles, CA.

Lidstone, G. J. (1920). Note on the general case of the bayes-laplace formula for inductive or a posteriori probabilities. *Transactions of the Faculty of Actuaries*, 8:182–192.

Lim, C. and Kirchhoff, K. (2008). Domain Adaptation Through Phrase Generalization for Improved Statistical Machine Translation Quality.

M. Crego, J., Yvon, F., and B. Mariño, J. (2011). N-code: an open-source Bilingual N-gram SMT Toolkit. *Prague Bulletin of Mathematical Linguistics*, 96:49–58.

Mahajan, M., Beeferman, D., and Huang, X. D. (1999). Improved topic-dependent language modeling using information retrieval techniques. In *Proceedings of the*

Acoustics, Speech, and Signal Processing, 1999. on 1999 IEEE International Conference - Volume 01, pages 541–544, Washington, DC. IEEE Computer Society.

Mangu, L., Brill, E., and Stolcke, A. (2000). Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Computer Speech & Language*, 14(4):373–400.

McClosky, D., Charniak, E., and Johnson, M. (2006). Effective self-training for parsing. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 152–159.

McClosky, D., Charniak, E., and Johnson, M. (2010). Automatic domain adaptation for parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 28–36, Los Angeles, CA.

Mitchell, L. and Roturier, J. (2012). Evaluation of Machine-Translated User Generated Content: A pilot study based on User Ratings. In *Proceedings of the 16th Annual Conference of the European Association for Machine Translation (EAMT-2012)*, pages 244–251, Trento, Italy.

Moore, R. C. and Lewis, W. (2010). Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort '10, pages 220–224.

Mosquera, A. and Moreda, P. (2011). Enhancing the discovery of informality levels in Web 2.0 texts. In *Proceedings of the 5th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics*, pages 243–247, Poznan, Poland.

Munteanu, D. S. and Marcu, D. (2005). Improving Machine Translation Performance by Exploiting Non-Parallel Corpora. *Computational Linguistics*, 31:477–504.

Muraki, K. (1987). PIVOT: Two-Phase Machine Translation System. In *Proceedings of the 1st Machine Translation Summit (MT Summit I)*, pages 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*, pages 173–180. North-Holland, Amsterdam, The Netherlands.

Nakov, P. (2008). Improving English-Spanish statistical machine translation: experiments in domain adaptation, sentence paraphrasing, tokenization, and recasing. In *ACL-08: HLT: Third Workshop on Statistical Machine Translation, Proceedings of the Workshop*, pages 147–150, Columbus, OH.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, pages 160–167, Sapporo, Japan.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29:19–51.

Ozdowska, S. and Way, A. (2009). Optimal Bilingual Data for French-English PB-SMT. In *Proceedings of the 13th Annual Conference of the European Association for Machine Translation (EAMT-2009)*, pages 96–103, Barcelona, Spain.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. In *40th Annual Meeting of the Association for Computational Linguistics, (ACL 2002)*, pages 311–318, Philadelphia, PA.

Pecina, P., Toral, A., Papavassiliou, V., Prokopidis, P., and van Genabith, J. (2012). Domain Adaptation of Statistical Machine Translation using Web-Crawled Resources: a Case Study. In *EAMT 2012: Proceedings of the 16th Annual Confer-*

ence of the European Association for Machine Translation, pages 145–152, Trento, Italy.

Penkale, S., Haque, R., Dandapat, S., Banerjee, P., Srivastava, A. K., Du, J., Pecina, P., Naskar, S. K., Forcada, M. L., and Way, A. (2010). MaTrEx: the DCU MT system for WMT 2010. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR*, WMT '10, pages 143–148, Uppsala, Sweden.

Plank, B. (2011). *Domain Adaptation for Parsing.* Ph.d. thesis, University of Groningen.

Ratnaparkhi, A. (1999). Learning to Parse Natural Language with Maximum Entropy Models. *Mach. Learn.*, 34(1-3):151–175.

Reichart, R. and Rappoport, A. (2007). Self-Training for Enhancement and Domain Adaptation of Statistical Parsers Trained on Small Datasets. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 616–623, Prague, Czech Republic.

Roark, B. and Bacchiani, M. (2003). Supervised and unsupervised PCFG adaptation to novel domains. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 126–133, Edmonton, Canada.

Rosti, A.-v. I., Ayan, N. F., Xiang, B., Matsoukas, S., Schwartz, R., and Dorr, B. J. (2007). Combining outputs from multiple machine translation systems. In *Proceedings of the North American Chapter of the Association for Computational Linguistics Human Language Technologies*, pages 228–235, Rochester, NY.

Roturier, J. and Bensadoun, A. (2011). Evaluation of MT Systems to Translate User Generated Content. In *Proceedings of the Thirteenth Machine Translation Summit*, pages 244–251, Xiamen,China.

Sagae, K. (2010). Self-training without reranking for parser domain adaptation and its impact on semantic role labeling. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, DANLP 2010, pages 37–44, Uppsala, Sweden.

Salton, G. and Buckley, C. (1997). *Term-weighting approaches in automatic text retrieval*, pages 323–328. Morgan Kaufmann Publishers Inc., San Francisco, CA.

Schwarm, S., Bulyko, I., and Ostendorf, M. (2004). Adaptive Language Modeling with Varied Sources to Cover New Vocabulary Item. *IEEE Speech and Audio Process*, 12(3):334–342.

Sekine, S. (1997). The domain dependence of parsing. In *Proceedings of the fifth conference on Applied natural language processing*, ANLC '97, pages 96–102, Washington, DC.

Sennrich, R. (2011). Combining Multi-Engine Machine Translation and Online Learning through Dynamic Phrase Tables. In *Proceedings of the 15th Annual Conference of the European Association for Machine Translation (EAMT-2011)*, pages 89–96, Leuven, Belgium.

Sennrich, R. (2012a). Mixture-Modeling with Unsupervised Clusters for Domain Adaptation in Statistical Machine Translation. In *Proceedings of the 16th Annual Conference of the European Association of Machine Translation (EAMT-2012)*, pages 185–192, Trento, Italy.

Sennrich, R. (2012b). Perplexity minimization for translation model domain adaptation in statistical machine translation. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2012)*, pages 539–549, Avignon, France.

Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227–244.

Shimohata, M., Sumita, E., and Matsumoto, Y. (2004). Building a Paraphrase Corpus for Speech Translation. In *Fourth International Conference on Language Resources and Evaluation, LREC 2004*, pages 1407–1410, Lisbon, Portugal.

Snover, M., Dorr, B., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231, Cambridge, MA.

Sriram, B. (2010). Short Text Classification in Twitter to Improve Information Filtering. Master's thesis, Ohio State University.

Steedman, M., Osborne, M., Sarkar, A., Clark, S., Hwa, R., Hockenmaier, J., Ruhlen, P., Baker, S., and Crim, J. (2003). Bootstrapping statistical parsers from small datasets. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics - Volume 1*, EACL '03, pages 331–338, Budapest, Hungary.

Stolcke, A. (2002). SRILM–An extensible language modeling toolkit. In *ICSLP 2002, Interspeech 2002: 7th International Conference on Spoken Language Processing*, pages 901–904, Denver, CO.

Storkey, A. J. and Sugiyama, M. (2007). Mixture Regression for Covariate Shift. In *Advances in Neural Information Processing Systems 19*, pages 1337–1344.

Tiedemann, J. (2009). News from OPUS - A collection of multilingual parallel corpora with tools and interfaces. In Nicolov, N., Bontcheva, K., Angelova, G., and Mitkov, R., editors, *Recent Advances in Natural Language Processing*, volume V, pages 237–248.

Tseng, H., Chang, P., Andrew, G., Jurafsky, D., and Manning, C. (2005). A conditional random field word segmenter. In *In Fourth SIGHAN Workshop on Chinese Language Processing*, Jeju Island, Republic of Korea.

Ueffing, N., Haffari, G., and Sarkar, A. (2007a). Transductive learning for statistical machine translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, pages 25–32, Prague, Czech Republic.

Ueffing, N., Simard, M., Larkin, S., and Johnson, H. (2007b). NRC's PORTAGE system for WMT 2007. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 185–188, Prague, Czech Republic.

Vashee, K. and Gibbs, R. (2010). Scenarios for Customizing an SMT Engine Based on Availability of Data. In *AMTA 2010: the Ninth conference of the Association for Machine Translation in the Americas*, pages 100–104.

Vauquois, B. and Christian, B. (1985). Automated Translation at Grenoble University. *Computational Linguistics*, **11**:28–36.

Weaver, W. (1949). Recent Contributions to the Mathematical Theory of Communication. In Shannon, C. E. and Weaver, W., editors, *The Mathematical Theory of Communication*, pages 94–117. The University of Illinois Press, Urbana, IL.

Wu, H. and Wang, H. (2004). Improving domain-specific word alignment with a general bilingual corpus. In *Machine Translation: From Real Users to Research, 6th Conference of the Association for Machine Translation in the Americas, AMTA 2004, Proceedings*, pages 262–271, Washington, DC.

Wu, H., Wang, H., and Liu, Z. (2005). Alignment model adaptation for domain-specific word alignment. In *ACL-05: 43rd Annual Meeting of the Association for Computational Linguistics*, pages 467–474, Ann Arbor, MI.

Wu, H., Wang, H., and Zong, C. (2008). Domain adaptation for statistical machine translation with domain dictionary and monolingual corpora. In *Proceedings of Coling 2008, 22nd International Conference on Computational Linguistics*, pages 993–1000, Manchester, United Kingdom.

Xu, J., Deng, Y., Gao, Y., and Ney, H. (2007). Domain Dependent Statistical Machine Translation. In *Proceedings of the 11th Machine Translation Summit, (MT SUMMIT XI)*, pages 515–520, Copenhagen, Denmark.

Yamamoto, H. and Sumita, E. (2008). Bilingual Cluster Based Models for Statistical Machine Translation. *IEICE - Trans. Inf. Syst.*, E91-D:588–597.

Yang, Y. and Liu, X. (1999). A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '99, pages 42–49, Berkeley, CA.

Yasuda, K., Zhang, R., Yamamoto, H., and Sumita, E. (2008). Method of Selecting Training Data to Build a Compact and Efficient Translation Model. In *Proceedings of International Joint Conference on Natural Language Processing*, pages 655–660, Hyderabad, India.

Yvon, F. (2010). Rewriting the orthography of SMS messages. *Natural Language Engineering*, 16(2):133–159.

Zens, R., Bender, O., Hasan, S., Khadivi, S., Matusov, E., Xu, J., Zhang, Y., and Ney, H. (2005). The RWTH Phrase-Based Statistical Machine Translation System. In *In Proceedings of the International Workshop on Spoken Language Translation (IWSLT-05)*, pages 155–162, Pittsburgh, PA.

Zens, R. and Ney, H. (2004). Improvements in Phrase-Based Statistical Machine Translation. In *In Proceedings of the Human Language Technology Conference / North American Chapter of the Association of Computational Linguistics Annual Meeting (HLT-NAACL-04)*, pages 257–264, Boston, MA.

Zhao, B., Eck, M., and Vogel, S. (2004). Language model adaptation for statistical machine translation with structured query models. In *Proceedings of 20th International Conference on Computational Linguistics*, pages 1–7, Geneva, Switzerland.

Zhu, X. (2005). Semi-Supervised Learning Literature Survey. Technical report, Computer Sciences, University of Wisconsin-Madison.