Machine Translation for Twitter

Examination Number: 6898847

Master of Science

Speech and Language Processing

School of Philosophy, Psychology

and Language Studies

University of Edinburgh

2010

Abstract

We carried out a study in which we explored the feasibility of machine translation for Twitter for the language pair English and German. As a first step we created a small bilingual corpus of 1,000 tweets. Using this corpus we carried out an analysis of the linguistic features of tweets. We tested different strategies of domain adaptation and found that they improved translation performance. In our experiments we found large differences in performance due to the handling of unknown words. By using xml-markup we were able to reduce this difference. We also replaced special Twitter expressions with placeholders, which enabled us to learn more robust n-gram statistics from Twitter data. We carried out a small-scale human evaluation to balance our automatic scores. Finally, we tested strategies to enforce translation output of legal length. Generating n-best-lists of translation candidates and searching for legal tweets was found to be helpful, but ultimately too unreliable because there was no systematic way to determine the required value of n. We suggested a feature function based on character count as a potential solution.

Acknowledgements

I would like to thank my supervisor Miles Osborne for his enthusiasm in this project, his helpful suggestions and for always answering my questions; the members of the machine translation group for the chance to present my project and the helpful feedback; the University of Edinburgh for the scholarship which made it possible for me to come here and embark on this new challenge which has been worth every day of it; my friends and roommates here who helped to keep me sane.

Danke an meine Eltern, die mich unterstützt und meine Entscheidung zum Fachwechsel mitgetragen haben; an meine Freunde zu Hause, die mich nicht vergessen haben; und an den Andi, dafür, dass du mich noch durch eine zweite Abschlussarbeit ertragen hast. Und für so Vieles mehr.

Declaration

I have read and understood The University of Edinburgh guidelines on Plagiarism and declare that this written dissertation is all my own work except where I indicate otherwise by proper use of quotes and references.

Laura Elisabeth Jehl

Contents

1 Introduction

This study will explore the feasibility of applying a state-of-the-art phrase-based machine translation system to Twitter.

Twitter is a social networking platform which allows registered users to post messages of up to 140 characters at a time as well as to follow the messages of other users. Started in October 2006, the number of its users had reached over a 100 million by April 2010 (eco, 2010). According to the company's own blog, approximately 65 million tweets are published per day (twb, 2010). Being a social networking site, a large portion of Twitter messages, or tweets, consists of "pointless babble" and private conversations (Kelly, 2009). Nevertheless, Twitter has become relevant as an important source for jounalists, a "living, breathing tip sheet for facts, new sources and story ideas" (Farhi, 2009). Although Twitter provides the option for users to protect their account, most accounts are public, allowing everyone direct, unmediated access to tweets. This immediate access to information also triggered research on Twitter users and usage in general as in (Zhao and Rosson, 2009), (Krishnamurthy et al., 2008) and (Java et al., 2007), as well as on specific topics such as identifying the first tweet about a piece of news (Petrovic et al., 2010).

As for as we know there has been no previous work on machine translation for Twitter. Recent studies about language use on Twitter have shown that users tweet in many different languages, some of which account for a substantial portion of the entire volume of tweets. A study on a corpus of 2.8 million tweets found that only 50% of the tweets were in English. Besides English, the most frequently identified languages were Japanese (14%), Portuguese (9%), Malay (6%) and Spanish (4%). German, Dutch and Italian each accounted for 1 to 2% of the corpus (Guyot, 2010).[1] Thus, translating Twitter would make significantly larger amounts of information accessible for journalists, social media researchers and Twitter users in general.

Twitter lends itself to machine translation for a practical reason. Since tweets are informal and short-lived, translating them is not a task that could sensibly be carried out by human

[1] Another study, carried out on a corpus of 8.9 million messages presents slightly different statistics. This data set contained 61% English tweets, 11% Portuguese, 6% Japanese and 4% Spanish (Crawford, 2010).

translators. Introducing machine translation for Twitter would be an addition to, not a replacement for, human translation services. What is more, applying machine translation to Twitter seems promising, since tweets are limited to 140 characters. This constraint restricts users to short, simple sentences which are generally easier for statistical machine translation systems since the possibilities for reordering will be limited.

Machine translation for Twitter can be understood as a domain adaptation problem, since there are no large bilingual Twitter corpora. The field of domain adaptation has been considered important, because the performance of a statistical machine translation system decays when faced with tasks form a different genre. However, previous work has mostly looked into adaptation to domains that were fairly close to the genre of the training data.[2] Twitter not only presents a new challenge for domain adaptation research, but is also suitable for domain adaptation experiments, since large amounts of monolingual in-domain data are freely available through its streaming API.

This study was guided by three main research questions:

1. What are the challenges posed by linguistics features of tweets and how can they be addressed?
2. How can machine translation for Twitter be improved using in-domain data?
3. How can we ensure that the translations are legal tweets?

The first question aims at taking a close look at the data and isolating particular problems related to tweets. Besides a large amount of monolingual data, a small parallel English-German corpus of 1,000 tweets was created as part of this project. Adaptation experiments were carried out to determine the effect of using mono- and bilingual adaptation data and to give an answer to the second question. In addition to that, the standard pre-processing pipeline was modified to deal with some of the problems arising from specific linguistic features of tweets. The third question concerns the constraint on length which requires translations of tweets to be no longer than 140 characters. We attempted two strategies two enforce this constraint.

From our research, we concluded the following findings:

- While the language of Twitter combines characteristics of several other electronic media genres, it constitutes a unique variety.
- There are great differences between the domain of the training data (proceedings from the European parliament) and the Twitter domain.

[2] See for example (Koehn and Schroeder, 2007) and (Bertoldi and Federico, 2009).

- Using in-domain adaptation data improves machine translation for Twitter.
- When domain adaptation is applied, the results differ greatly, depending on how unknown words are handled.
- The problem caused by unknown words can be reduced by introducing extra pre-processing steps and xml-markup.
- There are certain features of translations which readers dislike, but which are not sufficiently punished by automatic evaluation metrics.
- N-best-search is a promising first step towards enforcing output which is a legal tweet.

The explanation of our findings will proceed as follows: **Chapter 2** gives an overview of statistical phrase-based machine translation. In **Chapter 3** we compare the domain of the training data to the Twitter domain from a linguistic point of view. **Chapter 4** describes the creation of a bilingual Twitter corpus and gives details of our basic experimental setup. In **Chapters 5 - 7** we will present and discuss our experimental results: **Chapter 5** is concerned with domain adaptation. **Chapter 6** shows how pre-processing can be applied to improve results. **Chapter 7** discusses solutions to the problem of enforcing output of the right length. Finally, **chapter 8** concludes our study and outlines some directions for future work on machine translation for Twitter, based on the observations made during our research.

2 Background: Statistical Machine Translation

The following section will present an overview of phrase-based machine translation. We will cover its principles underlying models as well as training, tuning, decoding and the question how to evaluate machine translation output.

2.1 The principles of phrase-based MT

2.1.1 Words or phrases?

Before phrase-based machine translation came to be the state of the art, word-based models were standard. These models used individual words as the basic units of translation. In phrase-based machine translation an input sentence is segmented into phrases which are then translated separately. Phrases are units of one or more words. Finally, the translated phrases are re-ordered.

Figure 2.1 illustrates this process:

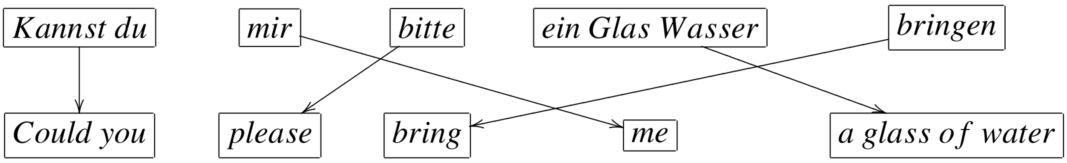

Figure 2.1: phrase-based machine translation

Using phrases instead of words offers an easy way of tackling the problem that there is often no one-to-one mapping between words in the source and target language. In figure 2.1 , for example, there is no German word which corresponds to the English preposition “of”. Word-based models are forced to adopt several complicated strategies such as introducing null-words or fertility models to fix this problem. But when we consider phrases instead of words, the model can learn that “a glass of water” is usually translated as “ein Glass Wasser”. Phrase-based machine translation provides two further advantages (Koehn, 2010, sect. 5.1.1). First, it helps to resolve translation ambiguities. Consider again the word “of” in the illustration. There are several translation options for “of”. But if the system uses phrases instead of words, it has a way to learn that in the context “a glass of water” “of” is omitted entirely when translating into German without altering “of” in other contexts. This contextual knowledge can therefore be incorporated into the model. Second, as training data increase, longer phrases can be learned. Consequently, phrase-based machine translation makes better use of training data.

2.1.2 The model

Phrase-based machine translation can be formulated as a noisy-channel model. In this model, which also applies to word-based translation, the probability of a target sentence **e** given a source sentence **f**, denoted $p(\mathbf{e}|\mathbf{f})$, is factored into two components: the reverse conditional probability $p(\mathbf{f}|\mathbf{e})$, which corresponds to the translation model, and the target sentence probability $p(\mathbf{e})$, corresponding to the language model.This factorization is derived using Bayes' rule:

$$p(\mathbf{e}|\mathbf{f}) = \frac{p(\mathbf{f}|\mathbf{e})p(\mathbf{e})}{p(\mathbf{f})}$$

The best translation is therefore given by

$$\text{argmax}_{\mathbf{e}} p(\mathbf{e}|\mathbf{f}) \;=\; \text{argmax}_{\mathbf{e}} \frac{p(\mathbf{f}|\mathbf{e})p(\mathbf{e})}{p(\mathbf{f})} \;=\; \text{argmax}_{\mathbf{e}} p(\mathbf{f}|\mathbf{e})p(\mathbf{e})$$

The equation holds because the denominator of the right-hand-side does not affect the argmax-function and can thus be dropped. In phrase-based machine translation the reverse translation probability of the whole sentence is further decomposed into the product of the translation probabilities of the individual phrases $\phi(\bar{f}_i|\bar{e}_i)$ and the reordering model. Reordering can either be handled by applying a cost function or using a lexicalized model. One simple cost function is the distance between a phrase and its translation, which will penalize larger reordering distances more heavily (Koehn, 2010, sect. 5.1.2). While conceptually simple, a distance-based reordering cost function does not take into account that some phrases are reordered more frequently than others. A lexicalized reordering model stores for each phrase the probability of it being swapped, monotone or discontinuous in a lexicalized reordering table (Koehn, 2010, sect. 5.4.2).

Applying the noisy-channel model to phrased-based machine translation can be problematic, since the distributions used in this model are estimated from limited training data and thus just a poor approximation of the true distribution. In addition to that, with the noisy-channel approach there is no straightforward way to include additional sources of knowledge, which might be helpful in determining the best translation (Och and Ney, 2002). A more flexible translation model has been formulated by (Och and Ney, 2002) using the maximum entropy framework. In this framework, the posterior probability $p(\mathbf{e}|\mathbf{f})$ is modeled directly as a combination of M feature functions with weights λ. The decision rule in this model is

$$\hat{e}_1^I = \text{argmax}_{e_1^I} \left\{ \sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J) \right\}$$

where I is the number of phrases in $\mathbf{e}$ and J the number of phrases in $\mathbf{f}$. This model allows for an arbitrary numbers of features to be included. For instance, instead of just using the reverse translation probability and the language model score, probabilities for both translation directions and several different language models can be used as features. Some common additional features include lexical translation probabilities, word and phrase counts or phrase pair frequency (Osborne and Koehn, 2010, lect. 11).

2.2 Training

The training process involves learning the parameters of the bidirectional translation model, the language model(s) and the lexicalized reordering model.

Learning phrase translation probabilities requires a sentence-aligned parallel corpus. Since language follows a Zipfian distribution – a small number of very frequent events occur alongside a large number of infrequent events – extracting robust statistics for phrase translation probabilities requires a very large corpus.

Even though phrase-based MT has replaced word-based models as the common paradigm in machine translation, these models are still important as an integral part of the training process of a phrase-based machine translation system. Phrase extraction presupposes a word-by-word alignment of each sentence pair in the parallel corpus. Given the word alignment, all phrase-pairs which are consistent with the alignment are extracted. A phrase-pair $(\bar{e}, \bar{f})$ is consistent with a word alignment if and only if there is no word $f \in \bar{f}$ which is aligned to $e \in \bar{e}$ but e is not aligned to f, and vice versa (Koehn, 2010, sect. 5.2.2). Note that this notion of consistency does not allow discontinuous phrases to be extracted. The maximum possible phrase length is the length of the sentence. However, since the counts for long phrases will be very low and in order to keep the size of the phrase translation table manageable, a maximum length of the extracted phrases can be fixed. Phrase translation probabilities are learned using relative frequency estimation on the parallel corpus. After phrase extraction is completed, the lexicalized reordering model is trained by counting how many times a phrase pair contains swapping, discontinuities or monotone orientation of the word-alignment points.

While the translation probabilities are related to the notion of translation adequacy, the language model probabilities are used to model fluency. Fluency can be related to both the question of appropriate word choice and appropriate word order. In order to approximate the true distribution of a language the language model has to be trained from a very large quantity of data. The language model contains statistics for text chunks of up to a fixed length n (n-grams) based on relative frequency estimation. Due to the Zipfian distribution of language, we expect some unseen n-grams, even when training language models from very large corpora. Since the language model approximates the probability of a string of words as the product of the probabilities of all n-grams contained in the string, an unseen n-gram with a probability of 0 would assign a score of 0 to the whole string. In order to avoid this problem the language model scores need to be smoothed, which means taking some of the probability mass from the seen events and assigning it to unseen events.

2.3 Tuning

After training of the translation, language and reordering model, the weights λ_i of the feature functions still need to be set. We would like to find weights which generate the smallest error on a held-out development corpus, according to some error metric. Since we ultimately strive to improve performance, it makes sense to optimize the weights against the same metric is then used as evaluation criterion. In this study, the BLEU-score was used as automatic evaluation criterion. We will discuss the BLEU-score in the following section on evaluation.

Since Moses uses only about 15 features, reliable estimation of weights can be carried out on a relatively small tuning set of a few hundred or thousand sentences (Koehn, 2010, sect. 9.3.1).[1] In the present case, two tuning sets were used:

1. a section of 1000 sentences from Europarl (part 09-01)
2. a parallel corpus of 500 tweets

Parameter tuning was carried out using *Minimum error rate training* (MERT) as first suggested by (Och, 2003). MERT sets the feature weights by generating an n-best list of possible

[1] (Bertoldi and Federico, 2009) show that for 5 features weight optimization results are almost as good for a tuning set of 200 sentences than they are for a larger set of 1,000 sentences. Of course, such a small tuning set is only sufficient for a small number of features. There is ongoing research into systems, which use millions of features. In this case, discriminative training has to be carried out on the whole corpus. See, e.g. (Blunsom et al., 2008).

translations for each sentence in the tuning set, calculating the BLEU-score for each output sentence against the reference translation and then adjusting the weights λ_i in such a way that the translation with the highest BLEU-score (or the lowest error against the reference translation) is re-ranked at the top of the list. Setting the weights requires an efficient search heuristic, since for M features we would theoretically need to explore an M-dimensional space over real numbers. A line optimization algorithm can be used for this purpose (Och, 2003). However, it is possible that optimal weights for a given n-best list may fit the translations in the n-best list, but still produce bad results on a test set. This problem can be solved by re-using the optimized weights in the decoder to generate a new n-best list. The new list is then combined with the previous one and the process is iterated until the n-best list does not change any more. In our case, the algorithm normally converged after 9-12 iterations.

2.4 Domain Adaptation

A common problem of statistical machine translation is a drop in performance if the domain of the test data does not match the domain of the training data. This is due to the fact, that the parameters of the translation and language model reflect the empirical distribution of the training data domain, which can be very different from that of another domain. As we have seen in the previous sections, the training process requires three different sets of data: a large bilingual training corpus, a small bilingual development set and a large monolingual dataset in the target language to train the language model. Due to very limited availability of large parallel corpora, there is often no in-domain training corpus. However, there are various ways of supplementing the out-of-domain data with in-domain material in order to adapt the system to perform better on a particular domain. For example, for some domains such as Twitter large amounts of monolingual data are available which can be used for language model training. If a small amount of bilingual data is available, these can either be added to the translation model or used as a development set.

2.5 Decoding

In order to find the best possible translation, we would have to explore (theoretically)

- all the ways of breaking an input sentence down into phrases,
- all the different translation options for each phrase

- all possible ways of reordering the phrase translations.

This search problem has been shown to be NP-complete (Knight, 1999). Consequently, heuristics need to be applied in order to limit the search space. During phrase-based decoding, the search space is explored dynamically by *hypothesis expansion*. The decoder begins with an empty hypothesis, then picks a translation option and records the partial score of this hypothesis as well as the translated input words. The hypothesis is then expanded by adding another translation option, which covers other input words and so on until all source words have been translated. A safe option to limit the search space is by recombining hypotheses: If two hypotheses cover the same input words, the lower-scoring one can be dropped (Koehn, 2010, sect. 6.2.4).[2] However, hypothesis recombination is not sufficient to reduce the search space to a manageable level.

While hypothesis recombination guarantees that the highest scoring translation will survive, other methods are more risky: the best translation may be pruned early on. These methods include limiting the number of phrase translation options which are loaded from the phrase table as well as introducing a stack size limit which controls the number of hypotheses which are explored at a stage in the translation process. A third method to speed up decoding which was applied here is cube-pruning, in which only the most promising entries are scored (Huang and Chiang, 2007).

2.6 Evaluation

Evaluating the output of machine translation systems has been subject of constant debate. Preferably, evaluation should be carried out manually by human judges who rate the output according to adequacy and fluency (Koehn, 2010, sect. 8.1). Variations in human judgment can be balanced out by supplementing the subjective evaluation with a task-based evaluation. Performance could be measured on e.g. a post-editing task or an understanding task (Koehn, 2010, sect.8.4.1 and 8.4.2). However, since human evaluation is expensive and time-consuming, it is not suitable for monitoring everyday progress during system development. Several automatic evaluation method have been proposed for machine translation, the most common of which is the Bilingual Evaluation Understudy BLEU (Papineni et al., 2002).

BLEU is a similarity metric which measures n-gram precision of a translation on a set of reference translations. Unigram precision accounts for correct word choice, while higher n-gram precision accounts for correct word order. According to Papinieni et al., the best correlation

[2] Note, however, that recombination is subject to constraints from the language and reordering models.

with human judgment was obtained by using a maximum order of 4 for n. However, two modifications had to be introduced to prevent "cheating". First, the count of an n-gram in the output cannot be higher than the maximum count of that n-gram in any of the reference translations. Otherwise, high n-gram precision could be achieved by generating sentences which contain a lot of stop-words. Second, a brevity penalty was introduced to penalize output which is shorter than the reference translation. The brevity penalty forestalls achieving high precision scores with translations which are short and therefore less error-prone. It acts as a substitute for recall, which cannot be sensibly used when there are multiple reference translations. After modifications, BLEU is calculated as follows:

$$\textsc{Bleu} = \mathrm{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

where N is the maximum order of n-grams, p_n designates the geometric mean of the modified precision for order n and BP indicates the brevity penalty. The weights w_n for the different order of n-gram precision are typically set to 1. While n-gram precision is calculated sentence by sentence, the brevity penalty is computed over the entire corpus in order not to punish short sentences more harshly. BP is computed as follows:

$$\mathrm{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

where r is the length of the reference translation[3] and c is the length of the candidate translation (Papineni et al., 2002). Therefore, as c gets smaller with respect to r, BP approaches 0.

As mentioned above, BLEU has become widely adopted because it rewards correct word choice and word order and, most importantly, has been shown to correlate with human judgment.[4] However, there are several deficits of BLEU which should be kept in mind when interpreting experiment results (Callison-Burch et al., 2006): First, BLEU only operates on an n-gram level. On the one hand, it provides no check whether the output is grammatically coherent. On the other hand, simply measuring n-gram-precision allows for a high degree of variation: there are many possible permutations of matching n-grams which can vary greatly in perceived quality. Second, BLEU treats all words identically. But in reality the omission certain words, such as *not*, would change the meaning dramatically, while dropping an adjective would not have less impact on the output adequacy. Third, BLEU only rewards n-grams

[3] If there is more than one reference translation, the best matching length is used for each sentence.

[4] See for example (Coughlin, 2003)

which exactly match one of the reference translations. Paraphrases or synonyms which do not appear in any of the reference translations cannot be accounted for. Along with the first two problems this can cause severe underestimation of some translation systems or even human translations. Last, the absolute BLEU-score a system received is meaningless and cannot be compared across languages.

Due to its shortcomings it has been suggested that in evaluation BLEU should only be used to track changes in a single system or to compare similar systems, but not to compare systems with very different translation strategies (Callison-Burch et al., 2006). We believe that this point also applies to comparing results across different text genres as we do in this study. Later on we will see that the twitter test set achieved generally higher scores than the Europarl test set. This can easily be explained by the shortness of the tweets, which makes it easier to achieve higher n-gram precision. However, only a subjective evaluation could really inform us whether the difference in BLEU indicates that the translation quality of tweets is indeed higher. We will therefore not draw any conclusions about the difficulty of the two translation tasks based on the comparative BLEU-scores. Nevertheless, BLEU is a good approximation for measuring small relative improvements for which a subjective evaluation is impractical.

2.7 Summary

In this chapter we gave a brief introduction to phrase-based machine translation, covering the translation model, the advantages of using phrases over words, the training and decoding processes and the question of evaluation. We also talked about the problem of domain mismatch and mentioned ways to use small amounts of bilingual in-domain data or large amounts of cheap monolingual in-domain data to overcome this problem. The next chapter will present an analysis of the linguistic features of tweets and a discussion of which particular challenges they pose for statistical phrase-based machine translation.

3 Linguistic characteristics of Twitter and Europarl

Since in this study we are concerned with domain adaptation, we will now take a closer look at the Twitter domain and compare it to the domain of the training data which were taken from the Europarl corpus, a collection of proceedings of the European Parliament (Koehn, 2005). After establishing our analytical framework, we will contrast the linguistic features of both domains and try to point out potential challenges for statistical machine translation.

3.1 Methodology

In order to set the framework for our analysis we will consider the language of Twitter and the language of the European Parliament proceedings as language *varieties*. Our use of this concept follows David Crystal's, who in his *Language and the Internet* defines a variety as "a system of linguistic expression whose use is governed by situational factors" (Crystal, 2006, p. 6). Consequently, we will first try to describe the "situational factors" which influence language use in both domains, before we will analyze the linguistic characteristics of each variety. According to Crystal, the linguistic features of a written variety are of five kinds: Graphic, orthographic, grammatical, lexical and discourse features (Crystal, 2006, p. 8f). We will limit our anaysis to only three types, the grammatical, lexical and orthographic features. Graphic and discourse features, which mainly include aspects of typography, layout and textual organization, will not be discussed because these aspects are irrelevant to machine translation which works on a sentence level and expects input in plain text.

Even though we are applying the descriptive framework for written varieties, both Twitter and the European Parliament proceedings are in different ways connected to spoken language, due to their situational factors. In his work, Crystal discusses the claim of internet language to be "written speech" (Crystal, 2006, p. 27) and examines which criteria of spoken or written language apply to which of the internet varieties he discusses. We will examine in what ways tweets and the European Parliament proceedings can be said to resemble speech and try to show that each is the very opposite of the other in this respect.

Crystal discusses several internet varieties, among which are email, chat, webpages, instant messaging and blogging. However, since it was published in 2006, before the advent of Twitter, “Language and the Internet” does not yet have a section on microblogging. We will try to show why it should be treated as a separate internet variety and hope that our analysis can serve as a preparation for a more in-depth work on Twitter with a linguistic focus.

3.2 The language of the European Parliament

The language of Europarl is originally spoken language. The corpus consists of transcriptions of face-to-face interaction between speakers which were in the same room. However, the corpus does not display many of the features typical of spoken language.[1] Some of them, such as prosodic richness and possible speaker overlap are not rendered in the transcription. However, due to the formal setting of the European Parliament, other typical characteristics of spoken language such as spontaneousness and loose structure (Crystal, 2006, p. Table 2.3) did not occur at all. Parliament sessions follow a strict protocol. Contributions of members are therefore less spontaneous and their content is strictly regulated. The linguistic features are indeed closer to the ones of written language, as we shall examine now.

At the sentence level, we notice fairly elaborate sentences with complicated structure. To draw a comparison, in a set of 1,000 sentences from the English portion of the Europarl corpus, the average sentence length was 154.20, while the average length of a tweet in a corpus of 1,000 English tweets was 79.98. The following example from the Europarl corpus serves as a – perhaps somewhat extreme – illustration of the long, convoluted sentences used in the language of the European Parliament:[2]

> Yet, while our own institution has done so much to ensure the successful completion of the work of the Convention, particularly by, on many occasions, making its premises available to the Convention, I am sorry that today, as our plenary part-session in Strasbourg is opening, it was not possible to organise the formal hand over of this Charter here in Strasbourg, in conditions worthy of it, now that the work of drafting it has been completed.

Complex sentence structures like this one can create difficulties for phrase-based machine translation, since (1), longer sentences give rise to more translation options, which increases

[1] For an overview of the characteristics of spoken and written language, see (Crystal, 2006, Table 2.3). For a criticism of viewing speech and writing as two distinct categories, see (Crystal, 2006, p. 25, fn. 5).

[2] Note, though, that with its length of almost 447 characters this sentence by far exceeds the average length.

the probability of errors and (2), phrased-based machine translation does not take syntactic information into account, consequently it is left to chance whether dependencies across clauses are adequately captured.

The vocabulary of the European Parliament sessions is strongly influenced by the formal setting. A very simple example is the choice of the German form of address. In German, the second person singular pronoun “Sie” (always with capital “S”) is always used to address someone in a formal context, “du” is used in an informal context. Due to the formal context, the German version of Europarl almost exclusively uses “Sie”. In the German version of the Europarl corpus 60,910 sentences containing one or more occurrences of “Sie” are paired with 381 sentences containing one or more occurences of “du”.

The Europarl proceedings follows the orthographical norms.

Having taken a very brief look at Europarl, we will now undertake a more detailed analysis of the language of Twitter.

3.3 The language of Twitter

There are big differences between the situational factors which shape the language of Twitter and Europarl. Unlike the debates of the European Parliament, tweets are not subject to the temporal and spatial constraints of spoken dialogue. Theoretically, they can be carefully examined and revised before publication. But tweets are not only temporally unconfined, their purpose is also not necessarily to take part in dialogue. Although there is conversational exchange on Twitter, as was mentioned in the introduction, many tweets are not directed at anyone in particular and there is often no expectation of a reply. Consequently the external situational factors of Twitter are very different from Europarl and more closely resemble written language. However, as we will see shortly, tweets share some of the spontaneity and expressivity of spoken language, due to the perceived informality of the setting.

In these respects Twitter is similar to other internet varieties, such as chatgroups, instant messaging or blogging (Crystal, 2006). However, what sets Twitter apart from these varieties is that each tweet is limited to 140 characters. This situational constraint lets Twitter border on text messaging. But while text messages are private and usually used in the context of a conversation, tweets are public and often not conversational. Twitter can thus be considered a hybrid of several different electronic media varieties which in its unique combination of features from these different genres constitutes its own variety.

Before we try to identify some linguistic features of tweets, it is important to mention that, due to the very low degree of regulation, they contain a high degree of variation. On the one hand, this variation is reflected in different user intentions, which are present on Twitter and lead to differences in language use. In (Java et al., 2007), the authors identify four main user intentions: Daily chatter, conversations, sharing information/URLs and reporting news. The following four examples from our Twitter corpus illustrate these different user intentions:

1. Awoke to loads of white stuff on the ground. There better have a been an explosion at the Colgate factory instead of what I think it is. #fb
2. @IssyK have fun! say hi to Joy :)) send her our love xox <3
3. RT @L9izHio: the @MovementofTruth free album!! http://www.dasouth.com/news/24-news/2490-download-qhollywood-letterq-by-movement-of-truth ...
4. R-Fed won 1st set! #AustralianOpen He'll surely win against Murray! :)

On the other hand, linguistic variation results from the many different social groups who use Twitter. Unlike members of a particular chatgroup, Twitter users are not associated with one big network, but are aggregated in countless clusters which could potentially establish their own language use.[3] Nevertheless, we will try to identify some of the salient phenomena which occurred repeatedly in our Twitter corpus, keeping in mind that they may not be applicable to all language use on Twitter.

Due to the constraint on length, sentences on Twitter are much shorter than in the Europarl corpus, leading to a much simpler syntactic structure, as is illustrated by the following example:

> @rediscover_me It's a 18 hour flight. Stay warm. See you soon.

As we will see below, the BLEU-scores achieved by the automatic Twitter translations are quite high compared to a control test set from Europarl. It is very likely, that this is, at least partly, caused by the shortness of sentences.

Another feature related to sentence structure, which could also be attributed to the length constraint, is the large number of elliptical constructions. We often see the first person singular pronoun and the copula ("I am") omitted at the beginning of a tweet, as in the following example:

> Enjoying my last morning in Brusse'ls/ sunny qfter the snow which wasn't a bas thing at all; Good conference, met interesting people;

[3] An exemplary analysis of communities on Twitter is carried out in (Java et al., 2007).

In the small Twitter dataset, there are 27 tweets out of 1000, which, like the example, begin with a verb in progressive form. Since German has no progressive form, most of these forms were translated using the first person present tense form of the verb. However, since Europarl does not contain such elliptical constructions, it is more likely, that the German present tense form will be aligned to the English "am –ing". Consequently, the translations tend to start with the auxiliary verb, as in the following example, in which, of course, the incorrect translation "looking" is selected for the German ambiguous verb "schaue" in addition to the spurious copula:

original:	watching jonas on disney channel =))
manual translation:	Schaue Jonas auf disney channel =))
automatic translation Ger → Eng:	am looking jonas on disney channel =))

However, the greatest challenges for a statistical translation system are not raised by Twitter's sentence structure. While this feature may even make the translation task easier, difficulties are bound to arise from the lexical and orthographic features.

The **vocabulary** of Twitter is as diverse as its users. Nevertheless, in comparison to the Europarl variety we can state that it is generally much more informal.

Throughout the dataset of 1,000 tweets, we noticed relatively frequent use of contracted forms of auxiliaries, such as "'I'll" (14 times), "won't" (2 times), "isn't"/ "ain't" (2 times), "gotta" (6 times) and "wanna" (8 times). In the case of "wanna", "gotta" and "I'll", these forms outnumbered their spelled out versions. In addition to contracted forms, we noticed ample use of strong language and slang expressions, many of which could only be understood with the help of websites like `urbandictionary.com`. While some of the slang expressions (like "oppa") probably mark affiliation with a particular social or ethnical group ("oppa" originates from Korean), others are common across informal internet language. Among these are commonly used abbreviations like "lol" (for "laughing out loud") or "smh" (for "shake my head"). Another common phenomenon in tweets is the frequent use of exclamations like "oh", "yay" and "aww", which are a means of representing a user's emotional state. The use of contracted forms, strong language, slang and exclamations all lend support to the assumption that the Twitter variety shares characteristics with spoken language.[4] These lexical phenomena also present a problem for a machine translation system trained on Europarl, since the phrase table will contain very few, if any, examples of them.

While these lexical phenomena can also be observed in other genres of the Internet, there are some terms which have been coined explicitly by Twitter. Among these are the verbs "to

[4] Crystal explicitely mentions contracted form, slang and obscenity as typical for spoken language (Crystal, 2006, table 2.1).

tweet", "to retweet", "to follow/unfollow", "to trend", "to favourite", "to list", but also nouns like "tweet", "twitter" or "follower". Since Twitter only came into existence in 2006, these terms are still in the process of being incorporated into language. In 2009 the verb "to twitter" was incorporated into the AP stylebook, a resource for journalists (aps, 2009). On `http://twtpoll.com/fapl15`, a German Twitter user created an opinion poll on whether people were calling the activity of posting tweets "twittern" or "tweeten" in German. Consequently, we are not only confronted with vocabulary which is likely to be unknown to the phrase table, but also with terms about whose use there is still insecurity among the users themselves.

Not only is the use of informal vocabulary common on Twitter, the orthography of tweets also tends to diverge from what is perceived as the norm.

First, there are some salient Twitter-specific orthographic features, namely the use of the characters "@" and "#". An @-symbol, immediately followed by a username either marks a reply to or a mention of another user. A #-tag, immediately followed by a word, is a topic marker. These special expressions present a challenge to an un-adapted machine translation system, since they have definitely not been seen in the training data. Also, using standard text-preprocessing will cause problems, since a standard tokenizer would separate the tag from the username or topic word. However, we would like to treat them as a unit. These issues will be addressed in later sections.

Since there are no spelling rules on Twitter, we observe much variation in spelling, punctuation and capitalization. We identified three kinds of variation: Careless variation, expressive variation and variation due to spatial constraints. Examples 1 to 6 below will be used as illustrations for all three kinds of variation.

1. Just Got up had some brecfast watched a bit of tennis and went on twitter! Cmon Andy Murray
2. party was awesome im buzzed but it was soo fun my girl daisy is freakin down!!!!!
3. I'm hooooome ppl http://myloc.me/3mJ6t
4. @Dionysius1 Dude, you ever get an iPhone, it's OOONNN lol.
5. Oi probably souldnt drunken tweet but oi am! Whats up?! :p
6. but when i go to her accnt, i can c my avatar on her following thingy but when i go 2 her LIST i dnt see myself.can she still see my tweets?

The first kind, variation out of carelessness, reflects the unimportance many users attribute to orthographic norms. Spelling errors, such as the word "brecfast" in example 1 below, are common. Punctuation is sometimes omitted as between the different sentences or clauses in example 2. Example 1 has sentence-initial capitalization, but the second word is also capitalized. In example 2 there is no capitalization either of the proper name ("daisy") or sentence-initial.

Nevertheless, while standard spelling, punctuation and capitalization are often neglected, many Twitter users make deliberate use of them for expressive purposes. We frequently saw repeated exclamation points or question marks to lend emphasis to a sentence, as can be seen in example 2. Emphasis is also marked by capitalizing words or repeating letters, as in examples 3 and 4. Crystal interprets such phenomena, which also occur in other internet varieties, as an imitation of the prosodic features of spoken language, but points out that their range of expressions is impoverished compared to the expressive power of prosody (Crystal, 2006, p.37f.). Some spelling variations are also used to mimick variation in a user's speech caused by their state, such as drunkenness or intoxication (example 5). Lastly, punctuation marks are also used to generate emoticons, which are a hint at a user's facial expressions as in example 5.

We believe that there is a third kind of orthographic variation on Twitter, which is caused by the constraint on length. In order to save space, users tend to omit spaces and abbreviate words as in example 6. These variations are intentional, but don't serve any expressive purpose. It is interesting, though, to see them appear even in tweets which could easily have been spelled out, such as

> its nly bcos of u i sttd twitter u

The use of abbreviations seem to have become a feature of Twitter language, regardless of its pragmatic use to avoid exceeding the length.

Non-standard spelling raises several issues for statistical machine translation. First, there are practical concerns: Non-standard spelling of a word would normally cause this word to be treated as unknown, even when a differently spelled version of it was seen during training. Also, non-standard use of punctuation can create problems for standard tokenization. However, there are also issues concerning translation quality and fidelity. If the output is supposed to match the input as closely as possible, the variations in spelling, punctuation and capitalization, which serve an expressive purpose should be replicated in the target language. But this is difficult for a statistical machine translation system, since it would either need to learn a translation model for every possible non-standard spelling and capitalization, which would require endless training data or apply standardization and subsequently recreate the original variations in the target language.

3.4 Summary

This chapter was devoted to an analysis of the situational factors which influence the language of Twitter and the resulting linguistic features of tweets. These were compared to the domain of the training data, which were taken from the Europarl corpus. We noticed great differences in the situational factors of both domains, which gave rise to very different linguistic characteristics of all three types (grammatical, lexical and orthographic). We would even suggest that the two domains are complimentary with respect to their position between written and spoken language. While Europarl originated from speech, its language has many of the characteristics of writing such as elaborate structure and careful wording. Tweets, on the other hand, are originally written, but their linguistic features resemble speech. We also pointed out some challenges for a machine translation system dealing with Twitter. Among those were the large number of unknown words created by twitter expressions (@usernames and hashtags) and slang, the variations in spelling, capitalization and punctuation, as well as elliptical constructions and abbreviations which arise from the constraint on length. This constraint, of course, poses a challenge in itself, since applying machine translation to Twitter would require output of legal length. Out of all these problems, this study will focus on the ones which are unique to Twitter, such as dealing with @usernames and hashtags as well as generating output of the correct length. In the next section, we will describe our experimental setup and mention some of the difficulties which had to be faced during the manual translation of 1000 tweets.

4 Experimental Setup

In this section we will describe our experimental setting for investigating the adaptation of a statistiscal machine translation translation system to the task of translating tweets from German into English. The first section describes how the datasets were created, focussing on the manual translation of 1,000 tweets. In sections two and three system parameters and evaluation methods will be discussed.

4.1 Data

Several datasets from the Twitter and Europarl domain were used in this study. We will give an overview of all datasets, before the manual creation of a small bilingual Twitter corpus will be described in more detail.

4.1.1 Overview of datasets

The Europarl corpus, on which our system was trained, is a parallel, sentence-aligned collection of proceedings of the European parliament from 1996 until 2009 in 11 languages (Koehn, 2005). The German and English sections are each comprised of about 40 million words. These sections were used to train the phrase translation model and the lexicalized reordering model. Furthermore, the English side of the training corpus was used to build a 5-gram language model. A test set (from section 10/2000) and tuning set (from section 04/2009) of 1,000 sentences each were set aside from the training data as a control for the domain adaptation experiments.

A large amount of monolingual Twitter data was gathered, using the Twitter streaming API. This interface allows to sample large amounts of tweets from public accounts.[1] The streamed data were filtered for English using a stop-word list.[2] Most of the crawled data (about 500 million tokens) was used to build a 5-gram language model. 1,000 tweets (13149 words)

[1] Twitter does not allow tweets from protected accounts to be sampled.

[2] On the whole, this method worked fairly well, but it was prone to occasional misclassification. We found that 64 out of 1,000 tweets were in other languages than English.

were set aside from the crawled data for the creation of a small bilingual corpus. The bilingual corpus was divided in a test set and a development set of 500 tweets. The next section provides more detail on the fabrication of the bilingual Twitter corpus.

4.1.2 Guidelines for the translation of Twitter

The foremost principle in manually translating tweets was to preserve the linguistic character of the input and the Twitter format. On the one hand, this meant observing the 140 character limit in the translations where possible. On the other hand, it was attempted to replicate the grammatical, lexical and orthographic features of the original in the translation.

Preserving the linguistic characteristics of tweets raised several questions. First, we had to decide how to handle special Twitter expressions (@usernames, hashtags and "RT"). Since replies and mentions (marked by "@") uniquely identify specific users, they had to be left as untranslated. The same is true for the acronym "RT" which identifies a tweet as a "retweet" from another user across languages. One could make a case for translating the topic markers, which are indicated by a #-sign. However, it seemed reasonable to leave them untranslated, since Twitter users might want to search for other tweets belonging to the original topic. If the hashtag were translated, the Twitter search would return tweets pertaining to the topic in the target language, not the source language.

Second, German translations had to be found for the new terms coined by Twitter. Translations were determined by first guessing possible candidates and then carrying out a Twitter and Google searches for them. Ultimately, Twitter terms were translated as follows:

to tweet	twittern
to retweet	retweeten
to unfollow	entfolgen
to list	listen
to trend	trenden
tweet (n)	tweet

Third, it was unclear how to deal with acronyms which frequently occur in internet varieties. These include expressions like "lol" (laughing out loud) and its variants "lmao" or "lmfao" and "smh" (shaking my head). http://de.wikipedia.org/wiki/Liste_von_Abkürzungen_(Netzjargon) listed these and many other originally English acronyms as being common practice in German web contexts. Being common usage, we used the English acronyms in our translations rather than attempted to invent German equivalents. There were two acronyms for which we found common German equivalents, which we used in our translations: According to the Wikipedia source, “ild” (ich liebe dich) replaces “ily” (I love you) in German web language and “kA” (keine Ahnung) is used instead of “idk” (I don’t know).

Fourth, it was attempted to preserve deliberate spelling variations where possible, as in the example below:

Original	@Lolo_B_Mackin aww thanxxxxx it shud be back on tomorrow!!
Translation	@Lolo_B_Mackin ooh dankeeee es sollte morgen zurück sein!!

However, the preservation of spelling variations should not outweigh the 140 character limit. We decided only to preserve clearly deliberate variation, while it was not attempted to imitate spelling or typing mistakes in the German translations. They were deemed less important, since they do not carry any specific pragmatic function.

Fifth, keeping translations within 140 characters was found to be very difficult in some cases, especially since English sentences tend to be slightly shorter than German ones. In order to save space, it was decided to mimick the English tweets’ tendency to abbreviate in the German versions. Nevertheless, English and German seem to lend themselves to somewhat different abbreviation strategies. In English we often see the omission of vowels or the replacement of the prepositions “to” and “for” by the numbers 2 and 4, which is not possible in German. Another strategy is the omission of apostrophes in contracted forms: “I’m” becomes “Im” etc. Since we had no German Twitter corpus available, finding abbreviations was guesswork. We tried to make educated guesses by searching on Twitter for what we thought likely to be an abbreviation. The example below serves as a sample of our approach:

Original	but when i go to her accnt, i can c my avatar on her following thingy but when i go 2 her LIST i dnt see myself.can she still see my tweets?
Translation	aber wennich auf ihr Konto geh, kannich mein Avatar in ihrm folgen-dings sehn aber auf ihrer LISTE seh ich mich net.Kann sie meine tweets noch sehn?

The English tweet is exactly 140 characters long. Note the dropping of vowels in “accnt” (account) and “dnt” (don’t), as well as the replacement of “see” with the phonetically equivalent “c”. In the German translation we eliminated the whitespaces, contracting “wenn ich” (when I) to “wennich” and equivalten with “kannich” (can I). Like the substitution of “c” for “see”, this strategy is inspired by speech, in which these words would also be contracted, eliminating the i-sound. In addition to that the third person singular pronoun “ihrem” (her) was shortened to “ihrm”, “nicht” (not) was replaced by the colloquial, shorter “net”. While these strategies intuitively make sense and were partially backed up by Twitter searches, we suggest that ideally a corpus analysis of the target language should be carried out to find common abbreviation trends.

Finally, some of the tweets were partially or entirely in a language other than English. These were left untranslated.

To summarise, our translations were guided by the following principles:

- The translation of a tweet should itself be in the format of a tweet.
- @usernames, hashtags and the retweet-marker should not be translated.
- Otherwise, as few words as possible should be left untranslated.
- Translations should be consistent. If the same tweet is repeated, the translation should be identical.
- Tweets or parts of tweets in other languages than the source language should not be translated.
- If possible, the linguistic characteristics of a tweet should be repeated in the translation.
- Target language abbreviations and expressions should be selected, which are already being used on Twitter.

After describing the datasets, the remaining part of the chapter will discuss system parameters and evaluation.

4.2 System

All experiments were carried out using the state-of-the-art open-source phrase-based statistical machine translation system Moses (Koehn et al., 2007).

The open source tool GIZA++ was used to obtain word-to-word-alignments during training (Och and Ney, 2003), using the symmetrized “grow-diag-final-and” method. This method first takes the intersection of the alignment points for both directions of a language pair and subsequently applies heuristics to “grow” missing alignment points. Since GIZA++ struggles with very long sentences, all sentences longer than 80 words were eliminated from the training corpus. Phrase extraction and training of the lexicalized reordering model were carried out using Moses’ training scripts. The default maximum phrase length of 7 was used. The orientation type for lexicalized reordering was msd-bidirectional-fe. The language models were trained using the SRILM language modeling toolkit (Stolcke, 2002) with Kneser-Ney-Smoothing.

ttable-length	Twitter test set
20	53.74
30	53.77
40	53.79
50	53.80
60	53.80

pop-limit/stack-size	Twitter test set
1000	53.77
2000	53.80
3000	53.77
4000	53.72
5000	53.70
10000	53.72

Table 4.1: Exploring different values for pop-limit and ttable-limit

We used the standard features implemented in Moses. This configuration consists of 5 features from the translation model (bidirectional phrase translation probabilities, bidirectional lexical weighting and a phrase penalty), 6 features from the lexicalized reodering model, one feature from the distance-based reordering model, 1 language model feature and a word penalty. For some experiments, we added an additional language model. Feature weights were estimated using Moses' built-in minimum error rate training procedure. The size of the n-best-list used in minimum error rate training was 100.

Since translation speed and quality are affected by the decoding parameters, we tried to choose settings which would maximize the quality of the Twitter translations, not worrying about speed too much. In order to determine good settings, we carried out some experiments on the Twitter test set. Two parameters needed to be selected: (1), the translation table length, i.e. the maximum number of translation option which are loaded for each source phrase and (2), the cube pruning pop limit, i.e. the number of hypotheses added to each stack. The reordering limit determining the maximum reordering distance between phrases, was left at the default value of 6 for almost all experiments, except an experiment on monotone translation for which the reordering limit was set to 0. Normally, the same decoding parameters are applied during testing as well as tuning. Since it would have been too time-consuming to re-run minimum error rate training for every parameter setting, which we wanted to test, we decided to evaluate different parameter settings only on the test set and to use the same weights for each experiment. The feature weights were obtained by using a translation table length (ttable-length) of 20 and a pop limit of 2000, which is a fairly small search beam.

For the test set, we first left the pop limit at 2000 and tried different settings for the ttable-length. Scores are shown in table 4.1.[3] Even though the difference between the scores is not significant, we do see a steady improvement which stagnates at ttable-length = 50. This result is unsurprising, since a too generous value of ttable-length will only cause spurious, low-

[3] In this setting, the Twitter development set and language model, as well as the Europarl development set were used. Unknown words were passed through the decoder.

probability translation options to be loaded. We also experimented with different numbers for the pop-limit, leaving the ttable-length at 50, since this value was found to be optimal. Results are shown in table 4.1. The BLEU-score was highest for pop-limit/stack-size = 2000. This ran counter to our expectations that a wider Beam would, while increasing decoding time, also increase the score. Possibly the slight drop in performance caused by larger beam width was caused by the shortness of tweets and the large number of unknown words. These two factors may have reduced the total number of meaningful hypotheses, since shortness leads to fewer reordering options and unknown words cause fewer translation options. Moses' experiment management system was used to automate the training, tuning and testing processes.[4]

4.3 Evaluation

We used BLEU as an automatic evaluation metric.[5] The BLEU score is always between 0 and 1, but for ease of reading we will report percent BLEU. In order to test whether an improvement was significant, we applied paired bootstrap resampling (Koehn, 2004). Since the BLEU score is not sentence-based, but calculated over the entire test set, we cannot compute confidence intervals from sentence-scores. In bootstrap resampling, this problem is solved by repeatedly sampling a test set of a fixed size *n* from all possible sentences. When the top and bottom 2.5% of the results are dropped, the remaining scores fall in an interval which approaches the 95% confidence interval as the number of trials gets large. Since sampling a test set from all possible sentences many times and translating it is computationally expensive, it is assumed that drawing a large number of test sets of size *n* from the same test set of size *n* with replacement is as good as drawing a large number of test sets of size *n* from an infinite set of test sentences (Koehn, 2004). This method requires only one run of the decoder. BLEU can then quickly be calculated from a stats-file which stores the required sentence-specific statistics. To evaluate whether the difference between a system A and a system B is significant with a p-value *p*, resampling and scoring is carried out *n* times for both systems, preferably with a large value of *n*.[6] If the score of system A was higher than system B's 95% of the time, we can conclude with 95% significance that A is better than B. It has been shown that the size of the test set affects the size of the difference in score required to achieve a significant result (Koehn, 2004). In the tests we carried out, a difference of 0.57% BLEU was enough for significance with a p-value of 0.05 for a test set of size 500. This corresponds roughly to the

[4] See `http://www.statmt.org/moses/?n=FactoredTraining.EMS`.

[5] see section 2.6.

[6] In our case, *n*=1,000 was used.

findings reported in (Koehn, 2004) for a test set of 600 sentences.

As mentioned earlier, there are some severe problems with BLEU. In order to gain an impression whether our scores corresponded to human judgements, we carried out a small-scale human evaluation. Evaluators were presented with translations from German into English generated by four different settings and asked to rank them on a scale from 1 to 4 (1 = best, 4 = worst). Two or more translations could be assigned the same rank. The original English tweet was given as a reference. Ranking was chosen over a subjective rating of adequacy and fluency because it has been shown that human judgements were more consistent on a ranking task (Callison-Burch et al., 2007).

We will also try, wherever possible, to illustrate our conclusions with examples from our qualitative evaluation of the output.

4.4 Summary

In this chapter we described our experimental setup. We described the Twitter and Europarl datasets and discussed some problems of creating a bilingual Twitter corpus. As a conclusion, we arrived at some guidelines for translating tweets. We then described the configuration of the Moses' system, which we used for training, tuning and decoding and gave some experimental support for our choice of decoding parameters. Finally, we described bootstrap resampling, a method of determining statistical significance of differences in BLEU score and briefly mentioned qualitative evaluation. The subsequent chapters will give a detailed discussion of our experiments and results.

5 Domain adaptation

In this chapter we will present the results of several experiments whose aim was to adapt our system to the Twitter domain by using in-domain data.

5.1 Methodology

As mentioned previously, two sets of in-domain adaptation data were available:

- A small bilingual corpus (500 tweets)
- A large monolingual corpus (58 Mill. English tweets)

We ran experiments to investigate two adaptation strategies: using the small bilingual corpus as a development set and using an in-domain language model trained on the large monolingual corpus. Experiments were conducted for each strategy separately before combining both. A baseline was established by running an experiment with a Europarl development set and language model. Moses has two ways of handling unknown words: They can either be dropped or passed through the decoder. Consequently, each experiment was run twice, once dropping unknown words and once keeping them. Experiments were carried out in both translation direction, German-English and English-German. Since we had no monolingual Twitter data in German, we could only test the effect of tuning on in-domain data for English-German translation. All settings were tested on a test set of 500 tweets and a test set of 1,000 Europarl sentences. Table 5.1 presents on overview of all experiments and results.

We expected that all settings involving in-domain data would outperform the baseline. We also assumed that using both an in-domain tuning set with an in-domain language model would do better than using onely one of the two in-domain datasets. We were interested to see which strategy was more helpful on its own and how each strategy affected the translation output. We also wanted to test whether it was better to use the in-domain language model on its own or a combination of the language models built from Twitter and Europarl. Since Moses uses a log-linear model, a second language model could be added as an additional feature.

	Settings						Results	
	Language Model:		**Tuning Set:**		**Unknown Words:**		**Test Set:**	
ID	Europarl	Twitter	Europarl	Twitter	dropped (-)	passed (+)	Europarl	Twitter
German → English								
Baseline								
1	X	-	X	-	X	-	27.56	25.58
2	X	-	X	-	-	X	27.74	39.08
Single strategy								
3	X	-	-	X	X	-	22.96	29.06
4	-	X	X	-	X	-	25.25	**30.48**
5	X	X	X	-	X	-	**28.09**	27.70
Combined strategies								
6	-	X	-	X	X	-	18.77	**33.00**
7	X	X	-	X	X	-	19.25	**32.73**
Single strategy								
8	X	-	-	X	-	X	25.12	**49.43**
9	-	X	X	-	-	X	25.15	47.58
10	X	X	X	-	-	X	27.40	42.37
Combined strategies								
11	-	X	-	X	-	X	19.96	52.76
12	X	X	-	X	-	X	20.35	**53.45**
English → German								
Baseline								
13	X	-	X	-	X	-	19.60	24.15
14	X	-	X	-	-	X	**20.00**	33.99
In-domain tuning								
15	X	-	-	X	X	-	19.52	27.11
16	X	-	-	X	-	X	17.57	**42.62**

Table 5.1: Overview of adaptation experiments and results.

Three adaptation techniques which have been applied in previous work were not explored here: First, due to lack of data we could not use an in-domain translation model (Koehn and Schroeder, 2007). Second, in addition to adding an in-domain language model as an extra feature, Koehn et al. also used an interpolated language model (Koehn and Schroeder, 2007). However, the performance of the interpolated language model was very similar to using two language models as separate features. Third, Bertoldi and Federico used synthetically generated parallel data (Bertoldi and Federico, 2009). Since these were found to produce only very small improvements, this method was not applied.

5.2 Results

The assumption that any in-domain data would be helpful was supported by our results. The BLEU scores for all adapted settings were significantly higher than the baseline. We will now give a detailed description of the results for each adaptation strategy and discuss some examples of the output.

5.2.1 In-domain tuning

Using an in-domain development set with an out-of-domain language model (experiments 3, 8, 15 and 16) improved scores by 3.48 (German-English) and 2.96 (English-German) points when unknown words were dropped and by 10.35 (German-English) and 8.63 (English-German) points when unknown words were retained. The following example illustrates the effects of in-domain tuning, as well as the differences caused by the handling of unknown words:

Reference:	RT @NigelLeck: IPCC and peer review independence, oxymoron comes to mind #agw #tcot #alot #climate #climategate
Baseline + unknown:	peer-review-unabhängigkeit rt @nigelleck : ipcc and i find that a #agw #climate oxymoron #climategate #alot #tcot
Experiment 8:	rt @nigelleck : ipcc and peer-review-unabhängigkeit , that reminds me oxymoron a #agw #tcot #alot #climate #climategate
Baseline -unknown:	: ipcc and strikes me as a
Exp. 3:	: ipcc and , it reminds me of a

First, we notice a striking difference between the +unknown and -unknown setting because all of the special Twitter-expressions were dropped as unknown words. When comparing the +unknown baseline to the output of experiment 8 we also see that experiment 8 correctly reproduced the order of the hashtags and @usernames at the beginning and end of the tweet, while in the baseline setting they are completely jumbled. It is likely that tuning on Twitter assigned less weight to the reordering model. Since the output generated by the -unknown settings has been dramatically shortened, there are fewer differences between the baseline and the adapted system. This explains why tuning on in-domain data produced a bigger improvement when unknown words were retained.

5.2.2 In-domain language model

Using an in-domain language model with an out-of-domain tuning set (Experiments 4 and 9) caused an improvement of 4.9 percent points when unknown words were dropped and 8.5 points when they were retained. Consequently, the in-domain language model was more helpful in a -unknown setting, but the in-domain tuning set produced better scores in a +unknown setting.

While in-domain tuning caused an improvement by limiting the amount of reordering, we expected the in-domain language model to have an impact on word choice. Due to the differences in the distribution of the training and test domain, a phrase which frequently occurred in the test domain could have low probability in the translation model. An in-domain language model which reflects the distribution of the test data can balance this mismatch by assigning a higher score to translation options which are more likely to occur in the test domain. We will try to compare the output produced by in-domain tuning and using an in-domain language model with reference to an example:

Reference:	@axixe "I think you are wrong to want a heart. It makes most people unhappy. If you only knew it, you are in luck not to have a heart."
Baseline +unknown:	i think you irrst @axixe ’you want to a heart. it makes people wüsstest most unfortunate. if you, you are lucky to have a heart’.
Exp. 8:	@axixe "i think du irrst thee a heart too want. it makes people mostly unfortunate. if du wüsstest, you’re happiness no heart."
Exp. 9:	@axixe ’i think du irrst you want to a heart. it makes people are unhappy. if du wüsstest, you’re lucky to have no heart. "
Baseline -unknown:	’i think you want you to a heart. it makes people mostly unfortunate. if you, you are lucky to have a heart’.
Exp. 3:	"i think du thee a heart too it makes people mostly if du, du haste, fortunately, is not a heart too have."
Exp. 4:	"i think you want a heart to you. it makes people are unhappy. if you, you’re lucky to have no heart."

Our assumption about the language model’s influence on word choice is supported by the fact the experiments with an in-domain language model (4 and 9) correctly produced the adjective “unhappy” rather than “unfortunate”. We also notice that in the +unknown settings both in-domain tuning and in-domain language model correctly placed the @username at the beginning of the tweet. It is difficult to pin down differences in the output between using an in-domain tuning set and an in-domain language model, based on just one example. When comparing experiments 3 and 4, the output of 4 is much more fluent and captures most of the meaning of the reference. It also correctly translated the German second person singular pronoun “du” as “you”. This correct translation is also present in the baseline, but not in the version with in-domain tuning. This illustrates that while improving the overall score, domain adaptation could also be harmful.

In addition to experiments 4 and 9 with just the Twitter language model we also studied a setting in which both the Twitter language model and the Europarl language model were used as features (experiments 5 and 10). We found that this setting produced significantly lower scores on the Twitter test set than just the Twitter language model. The difference was 2.78 BLEU points for -unknown and 5.21 BLEU points for +unknown.

5.2.3 Combined adaptation

Combining both adaptation methods (experiments 6, 7, 11 and 12) beat both strategies when applied on their own, thus confirming our previous assumption. For -unknown, the improvement from the baseline was 7.42 points with only the Twitter language model and 7.15 points with both language models. For +unknown, the improvement was 13.68 points with only the Twitter language model and 14.37 points with both language models. We notice that in this setting using both language models did not harm the performance. For -unknown the difference between the scores for using just the Twitter language model or both language models was insignificant. For +unknown the two settings differed by 0.69 BLEU points with the higher score being received by the system with two language models. We performed paired bootstrap resampling and found this difference to be significant with 95% confidence. However, we were sceptical whether this would warrant any conclusions about the two methods or whether this variation was merely caused by the random starting conditions of minimum error rate training. We conducted both experiments a second time, leaving all settings identical. This time, the difference between the two systems was 0.02 percent BLEU which supported our hypothesis that the previous difference had not been caused by a true superiority of one setting over the other. Consequently, we can conclude that adding an out-of-domain language model made no significant difference when both adaptation strategies were employed. This confirms results by earlier experiments, in which it an extra out-of-domain language model did not make a significant difference (Koehn and Schroeder, 2007).

5.3 Discussion

The following conclusions could be drawn from the domain adaptation experiments:

- Any in-domain data was better than no in-domain data.
- Combining both adaptation strategies was more helpful than using each of them on their own.
- For +unknown, tuning on Twitter was slightly more helpful, for -unknown the in-domain language model was more helpful. In general, improvement rates were higher for +unknown.
- Adding an out-of-domain language model was harmful when out-of-domain adaptation data were used and made no difference in the combined setting.

The different effects of the out-of-domain language model can be explained by the fact that the language model is used during tuning. It is likely that the language model which matches the domain of the development set will be assigned a higher weight, since, as we have seen, using an in-domain language model produces higher BLEU scores. Consequently, including the Europarl language model when no in-domain tuning set was used caused the Twitter language model to have less weight and thus produced lower scores on the Twitter test set. This explanation is also supported by the scores of the Europarl test set. Using two language models with a Europarl development set even increased the BLEU score slightly compared to the baseline.

Besides the effects of domain adaptation, we observed that dropping or keeping unknown words caused an enormous difference in BLEU scores for the Twitter test set, while it only affected the Europarl test set very slightly. The next chapter will explore the causes of this observation and propose a remedy to one of them.

6 Pre-processing Tweets

In this chapter we will analyse the reasons for the differences in scores on the Twitter test set for settings which include or drop unknown words. We will then introduce a way to address one of the reasons.

6.1 Causes for score differences caused by unknown words

Dropping unknown words causes a drop in BLEU for two reasons. When unknown words are omitted, shorter output is produced which is penalized by BLEU. What is more, our Twitter corpus contained many words which were identical in the source and target language. Not only does the brevity penalty not apply when unknown words are preserved, but the fact that many of them are identical in the original and the reference translation also leads to higher n-gram precision. We would nevertheless like to avoid passing through unknown words, since this setting produces output like the following example:

> green energy in eigenbau schritt-für-schritt installation of solar-und windmodulen - http://bit.ly/kitiu

We assume that the human perception of quality of such translations is much more negative than is suggested by BLEU.

In order to reduce the gap between scores for dropping and keeping unknown words we first looked at the translation output and identified what kind of input words were dropped that should have been preserved. The examples below serve to illustrate our findings:

Example 1	
reference:	Wedeh,monas RT @raimasiv: Tiba di ibu kota tercinta... Monas I'm in love http://myloc.me/3mITO
manual translation:	Wedeh,monas RT @raimasiv: Tiba di ibu kota tercinta... Monas Ich bin verliebt http://myloc.me/3mITO
+ unknown words:	wedeh, monas rt @raimasiv: tiba di ibu kota tercinta... monas i am in love http://myloc.me/3mito
- unknown words:	, : di ... i am in love http://myloc.me/3mito
Example 2	
reference:	RT @BBVIPForum: WE NEED MORE VIPS AND KPOP FANS! BBISVIP
manual translation:	RT @BBVIPForum: WIR BRAUCHEN MEHR VIPS UND KPOP-FANS! BBISVIP
+ unknown words:	rt @bbvipforum: we need more vips and kpop-fans! bbisvip
- unknown words:	: we need more ! and

The following kinds of words/phrases were identical in the input and output.

- In example 1, the reference contains non-English (*code-switching*).
- Example 2 contains expressions from popular language ("VIPS", "KPOP-FANS"), which have been imported into German as *English loans*.
- Example 1 and 2 contain *proper names* ("Monas" and "BBISVIP") which are preserved in the translation.
- Both example 1 and 2 contain *special Twitter-expressions* ("RT", "@raimasiv", "@BB-VIPForum").

We believe that these four phenomena, code-switching, domain-specific English loans, proper names and Twitter-expressions, are mainly responsible for the dropping of scores caused by the dropping of unknown words, along with the brevity penalty.

In this study we will only address the question of how to preserve Twitter-expressions, since this question is specific to Twitter. We will suggest a way of forcing them to be rendered in a certain way in the output and to preserve their correct order. URLs were also treated as Twitter-expressions, since these appeared frequently in our data.

6.2 Methodology

In order to deal with Twitter-specific expressions, several additional pre- and post-processing steps were introduced.

First, we needed to ensure that these expressions were treated as a single word. When just the regular tokenizer was used, @- and #-symbols were separated from their initial symbols. This caused them to be dropped or the tag to be translated into the target language. We tackled this problem by introducing an extra tokenization step. Tokenization was followed by a repair-step which eliminated whitespace from tokenized @usernames, hashtags and URLs. Since in our data the @-symbol was also used in tweets instead of the preposition “at”, we also introduced a cleaning step, which – prior to tokenization – replaced all @’s between whitespaces with the word “atsign” to prevent them from being accidentally “repaired”. The @-symbols were restored after the repairing step was carried out.

Second, an additional step was added to the pre-processing pipeline which applied xml-markup to all hashtags, @usernames, URLs and “RT”s. This method exploits Moses’ feature to enforce a particular translations by using xml-markup which prevents The Twitter-specific expressions from being dropped or wrongly translated into the target language. A particularly significant example of such a mistranslation occurred when “rt” was rendered in German as “programmatischen Ohnmacht befallen”. Not only did this cause very strange output, it also increased output length. We expected xml-markup to cause an improvement in performance for the -unknown setting, since it ensured that Twitter-expressions were preseved

Third, it was decided that hashtags, @usernames, URLs should not be treated like regular words. An additional pre-processing step was added to the pipeline. In this step, all @usernames, hashtags and URLs were replaced with dummy placeholders, distinguishing only between the number of occurrences of these expressions within a tweet. For example, the first occurrence of an @username in a tweet would be replaced by @1, the second one by @2 etc. Our rationale behind this extra step was that a language model trained on pre-processed data would learn more robust n-gram-statistics. If usernames and topic markers were treated like regular words, separate statistics would be collected for each different username. The trigrams [RT ,@arshas, :] and [RT, @IccaAyu, :] would in this setting be stored separately. But, instead of these separate trigrams, we would prefer the language model to learn that the combination [RT, @username, :] occurred frequently. By replacing Twitter expressions with placeholders this aim could be achieved. Since in the final translation output the original @usernames etc. should be restored, a post-processing step was added which re-inserted the original expressions. We preferred the pre-processed language model as a more consistent way of modelling the distribution of language in tweets. But it was unclear what contribution it would make towards translation performance. We expected it to help recreate the correct order of Twitter expressions.

Figure 6.1 illustrates the new pre- and post-processing pipelines. In the following section we will present the results of applying pre- and post-processing to translation for twitter. Pre-processing was only tested for a setting which applied the combined domain adaptation strategies, since we were interested in whether it would create any further improvement. We will report both BLEU scores as well as the results of a small scale human evaluation.

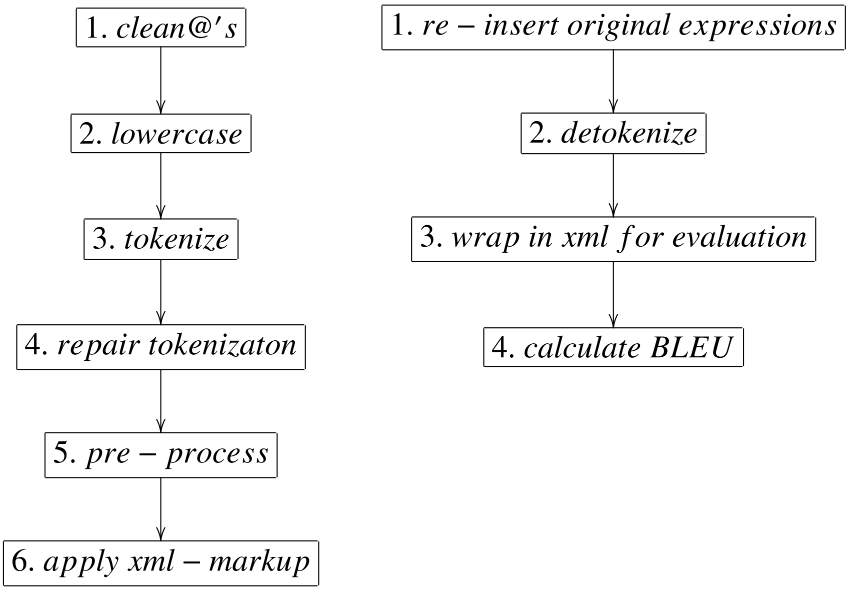

Figure 6.1: The pre-processing (left) and post-processing (right) pipelines

6.3 Results

The results of applying the new pre- and post-processing pipeline are shown in table 6.1.

There was no significant difference between domain adaptation and domain adaptation with pre-processing for the +unknown setting. This is probably due to the facts that were preserved in this setting anyway and that, as we have seen above, using adaptation data already caused correct reordering in most cases. However, there were exceptions, as can be seen in the following example:

Reference:	RT @audidess: RT @syally: RT @alyssaadzhani: RT @LoveHasQuotes: #tellmewhy you're too hard to forget.
Baseline +unknown:	@audidess: rt rt @alyssaadzhani: rt @syally: rt @lovehasquotes: #tellmewhy you are too difficult to forget.
+Adaptation:	@audidess @syally rt: rt: rt: rt: @alyssaadzhani @lovehasquotes #tellmewhy you are too hard to forget.
+Pre-processing:	rt @audidess: rt @syally: rt @alyssaadzhani: rt @lovehasquotes: #tellmewhy you are too hard to forget.

In this case neither the baseline nor the translation with domain adaptation preserved the correct order of the preamble. Only after pre-processing could the correct arrangement be restored. However, since quadruple retweets like the one in this example seem to be fairly

settings	- unknown words	+ unknown words
Baseline	25.58	39.08
+ domain adaptation	33.00	**53.45**
+ pre-processing	**43.17**	**53.31**

Table 6.1: Results for applying pre-processing

uncommon (this was the only one in 1,000 tweets), this improvement was not reflected in the BLEU score.

The assumption that pre-processing would improve results for the -unknown setting was correct. Pre-processing and xml-wrapping increased BLEU by 10.17 points, compared to just using domain adaptation and by 17.59 points compared to the baseline.

In order to determine whether the conclusions drawn based on BLEU corresponded to human judgements, a small-scale human evaluation was carried out. We obtained rankings of translation output from 22 human evaluators who contributed a total of 156 datapoints.[1] Figure 6.2 shows how many times each system received each rank. Table 6.2 gives an overview of the systems which were evaluated and their BLEU scores. Due to the small size of the sample we did not carry out statistical tests. But we will outline some tendencies which can be seen in figure 6.2. System 3 was ranked first most often, systems 1 and 2 were ranked third most often and baseline was most often ranked last. The rankings of system 1, 2 and 3 appear to correspond with the rankings by BLEU. However, the tendency of the baseline to be perceived as worse than system 1 deviates from the conclusion suggested by BLEU. The BLEU score of the baseline is 6.35 points higher than the score of system 1. We believe this deviation to be caused by two differences between system 1 and the baseline which are perceived as more harmful by human evaluators than BLEU can account for. First, the baseline translations tended to contain more reordering than all the adapted systems, as we showed in chapter 5. As we explained in section 2.6, since BLEU is based on n-gram precision, it allows for permutations at n-gram boundaries which do not lead to a change in score. This explains why incorrect order may not be punished as severely as desired.

Second, some of the baseline translations contained unwanted source words. It is likely that unwanted source language words would lead to a decrease in readability which could lead human readers to a negative judgement, an aspect which is not accounted for by BLEU. This tendency also emerges when we compare the rankings of system 2 and 3. While system 3 outperformed system 2 in being ranked first, it was also ranked last more often. System 2, on the other hand, received the fewest rankings at the bottom of the scale.

[1] Unfortunately, due to problems with the online survey, only one judgement was recorded for some participants.

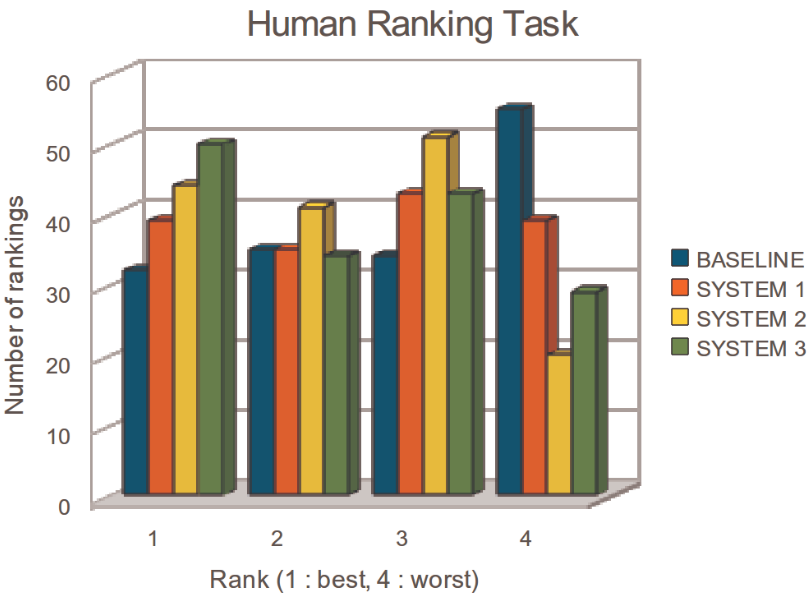

Figure 6.2: Results of Human Evaluation

SYSTEM	TUNING	LANGUAGE MODEL	UNKNOWN WORDS	PRE-PROCESSING	BLEU
Baseline	Europarl	Europarl	+	no	39.08
System 1	Twitter	Europarl+Twitter	-	no	32.73
System 2	Twitter	Europarl+Twitter	-	yes	53.47
System 3	Twitter	Twitter	+	yes	43.17

Table 6.2: Systems in human evaluation

6.4 Discussion

In this chapter we first analyzed the causes of the difference in BLEU scores which is caused by differences in handling unknown words. We then investigated the possibility of improving BLEU-scores by using a modified pre-processing-pipeline which was applied to the training data and the language model, as well. This pipeline contained a new tokenization step to repair tokenization of Twitter expressions, a pre-processing step which replaced Twitter expressions with uniform placeholders and a step in which xml-markup was applied to Twitter expressions.

The following conclusions could be drawn from our results:

- The differences in score, depending on the handling of unknown words, were caused by code-switching, in-domain English loans, proper names and Twitter expressions.
- Since pre-processing successfully halved the gap in BLEU scores, dropping Twitter expressions was indeed one of the prominent causes of this problem.
- Pre-processing did not cause any significant changes in the BLEU score for the +unknown setting, but an example showed that it improved reordering.
- A human evaluation of translations generated by four different settings suggested that incorrect ordering and unwanted source words were perceived as more harmful than was accounted for by BLEU.

With xml-markup and replacing Twitter expressions by uniform placeholders we proposed a solution prevent Twitter expressions from being dropped as unknown words. We decided to work on this problem, since it is specific to Twitter. We believe that at least some of the other three problems could also be solved: Named entity recognition methods could be applied to identify proper names. A more sophisticated language filter which operates on a sub-sentence level could be used to identify code-switching. Only domain-specific loans seem to require in-domain training data.

When we conclude that the pre-processed language model “improved reordering”, we are actually saying that, unlike other settings, it preserved the order of the source. This may lead to the suggestion that instead of trying to reconstruct the original order, we could simply apply monotone translation.[2] In order to see how well monotone translation worked we ran one experiment with a reordering limit of 0, using the +unknown setting and combined adaptation strategies. The fact that the result, 51.49 percent BLEU, was just barely lower than our best

[2] This suggestion was made by Adam Lopez.

overall results could either mean that the reordering model barely had any influence in the adapted systems or that tweets do not require large amounts of reordering.

With respect to the last conlusion, one could argue that not only does BLEU not account for aspects of readability, but that it actually prefers unknown source words to be kept rather than dropped. If we imagine two translations of the same sentence, both of which contain the exact same phrase translation options, but one of them kept unknown words while the other dropped them. Since both translations are identical apart from the unknown words, they would receive identical precision scores. Yet, the translation which dropped the unknown words would receive a brevity penalty while the other translation would not.

In the next chapter we will depart from trying to improve scores and attack the problem of how to enforce correct output length.

7 Creating legal tweets

In this chapter we will discuss some solutions to the problem of forcing the decoder to produce translations which are legal tweets, i.e. translations whose length does not exceed 140 characters. We will present two approaches to dealing with this problem.

7.1 Methodology

Before exploring solutions we should answer the question whether overlong output is indeed a problem. For German → English the baseline setting produced 3.4% overlong translations (17 tweets). For the English → German baseline, there were 17.2% overlong translations (86 tweets). This difference probably results from characteristics of the target language.[1]

Our first strategy consisted in modifying the tuning process. Since minimum error rate training optimizes against the BLEU score, we were hoping that if the BLEU scorer was modified to penalize overlong tweets, the weights would be adjusted to prefer legal translations. One might argue that the regular BLEU score already controls translation length through its combination of n-gram precision and brevity penalty: If the translation contains too many words, the additional n-grams will cause precision to drop. But if the output has fewer words than the reference length, a brevity penalty will be applied. However, both n-gram precision and brevity penalty only operate on word count, not on character count. What is more, the brevity penalty is not calculated sentence by sentence, but on the entire translation, allowing some freedom for the length of individual sentences. For this reason, the only way to introduce a penalty which operates on sentence level was modifying the precision score which is calculated per sentence. We first tried setting precision to 0 if the character count of a sentence exceeded a length l. In the second attempt, precision of sentences with more than l characters was multiplied by 0.1. Since the development set was tokenized, the parameter l had to be set to a higher value than 140. We ran experiments with l=145 and l=152.

The second strategy took advantage of Moses' option to generate n-best lists of translations,

[1] Some of the overlong tweets in German could also have been caused by the aforementioned incorrect translation for “rt”, which was contained in the phrase table.

instead of just returning the highest scoring translation. An n-best search was implemented, which would search the n-best list for a legal candidate translation if the top translation was too long. The search was carried out top to bottom, returning the first suitable candidate. The advantages of this strategy are its simplicity and its guarantee to succeed or at least not to cause any harm.

Each method was evaluated by investigating whether it successfully reduced the number of overlong tweets and, if this was the case, by testing whether or not its BLEU score beat a baseline of simply truncating overlong lines. We used the +unknown setting for all experiments, since dropping unknown words shortened the output.

7.2 Results

7.2.1 Modifying tuning

The first strategy – modifying the BLEU-scorer during tuning – turned out not to be very helpful. However, there were some tendencies which we find worth reporting. All translations were produced using a translation table limit of 20 and a cube pruning pop limit of 2000, with unknown words passed through the decoder. The system was tuned on tweets and used both Twitter and Europarl language model. Looking at the length of the output in table 7.1, we observe that the modified BLEU-scorer caused a decrease in the average output length for all parameter settings. The degree the reduction depended on the length limit (*l*=145) and on the type of penalty. The harsher penalty of reducing precision to 0 for all overlong translations lead to greater reductions in average length. There was also a reduction in the number of overlong tweets. However, the reduction lowered the number of overlong tweets by no more than 37.5%. In addition to that, there was no clear correlation between *l*, the penalty and the amount of overlong tweets.[2]

We will compare the translations generated with weights learned from modified minimum error rate training to translations generated with weights learned through the regular MERT-process. Examples 1 and 2 show the different output for different settings.

[2] However, a correlation might still be revealed by exploring more parameter values.

MERT-settings	average length	number of overlong sentences
reference	79.91	0
no modifications	80.69	16
maximum length=152, precision=0	79.35	13
maximum length=145, precision=0	78.19	10
maximum length=152, lowered precision	80.39	12
maximum length=145, lowered precision	78.61	15

Table 7.1: Length reduction through modified minimum error rate training

Example 1

Reference:	@LHGP4 haha. Knowing your luck you will! So will I dw. Lmao. Yeah, but apparently he teaches yr 9 re!! D: I want mr koch again. :D (130)
No modifications:	@lhgp4 haha. when are you going down to your happiness! and i also dw. lmao. yes, but it seems that teaches it 9 kl. 're! d: i want mr cook again.: d (149)
adjusted length=152, lowered precision:	@lhgp4 haha. when are you going down to your happiness! and i also dw. lmao. yes, but apparently he teaches 9 kl. 're! d: i want back mr koch.: d (145)
adjusted length=145, lowered precision:	@lhgp4 haha. when are you going to your happiness! and i also dw. lmao. yes, but it seems that teaches it 9 kl. 're! d: i want back mr koch.: d (143)
adjusted length=152, 0-precision penalty:	@lhgp4 haha. if your lucky you bet! and i also dw. lmao. yes, but apparently he has 9 kl. 're! d: i want mr. cook again.: d (123)
adjusted length=145, 0-precision penalty:	@lhgp4 haha. your lucky you bet! and i also dw. lmao. yes, but apparently they're taught 9th kl.! d: i want mr cook again.: d (125)

Example 2

Reference:	Your fantasies can be quite active today, but you cannot affor... More for Libra http://twittascope.com/twittascope/?sign=7 (123)
No modifications:	your fantasies are very active today, but you may not be doing... more for non-automatic weighing instruments http://twittascope.com/twittascope/?sign=7 (152)
adjusted length=152, lowered precision:	your fantasies can be very active today, but they cannot do more for non-automatic weighing instruments http://twittascope.com/twittascope/?sign=7... (149)
adjusted length=145, lowered precision:	your fantasies today is very hard, but they cannot do more for non-automatic weighing instruments http://twittascope.com/twittascope/?sign=7... (143)
adjusted length=152, 0-precision penalty:	your fantasies are very active today, but you can not do... more on non-automatic weighing instruments http://twittascope.com/twittascope/?sign=7 (145)
adjusted length=145, 0-precision penalty:	your fantasies are very active today, but they cannot do... more on non-automatic weighing instruments http://twittascope.com/twittascope/?sign=7 (145)

We see that some of the differences between the translations actually come close to what human translators would do to create shorter output. In Example 1 all but one of the modified settings choose "apparently" instead of "it seems that", which brings them closer to the reference translation. Also, the versions with the harsher precision penalty prefer "lucky"

modification	BLEU (truncated)
+adaptation/pre-processing	**53.49**
l=152, precision=0	52.81
l=145, precision=0	52.59
l=152, lowered precision	**53.25**
l=145, lowered precision	52.34

Table 7.2: Scores produced by modified minimum error rate training

to "happiness". In example 2, shorter output is created by using the contracted form "cannot" instead of "may not" or "can not", as well as by dropping the three dots. We see that sometimes enforcing shorter translations actually helped pick better translation options for individual words. However, when looking at the BLEU score, we see that modifying the tuning process did not beat the baseline (see table 7.2). All modifications produced lower scores (for one setting, however, the difference to the unmodified score was not significant with $p = 0.05$ as indicated by bold print). The drop in score was more severe for $l = 145$. It is not clear from the results how precision penalty affected the score.

7.2.2 N-best-search

In order to test different values of n, we set aside a portion of 18 tweets from the development set which had generated overlong translations during previous tuning runs. Minimum error rate training was carried out on the remaining portion of the tuning set. A pop-limit of 2000 and a ttable-limit of 50 were used. After running several tests for

$$n \in \{10, 25, 50, 100, 500, 1000, 2000, 3000, 4000, 5000, 10,000, 20,000, 30,000, 50,000\}$$

on the development set, we used the best value of n to do a run on the English and German test set.

Figure 7.1 shows the BLEU scores for truncated and non-truncated output (right) as well as the number of overlong sentences after n-best search on the 18 development sentences (left). As expected, n-best-search decreased the number of overlong sentences as n got larger. However, the decrease was not linear. 5 overlong tweets were already eliminated when $n = 10$, while it took $n = 30,000$ to eliminate another 4 overlong tweets. The BLEU score was slightly higher for larger values of n, both for the truncated and untruncated versions of the output. This increase can be explained by the difference between the training and adaptation domain. The translation model contains an approximation of the empirical distribution of

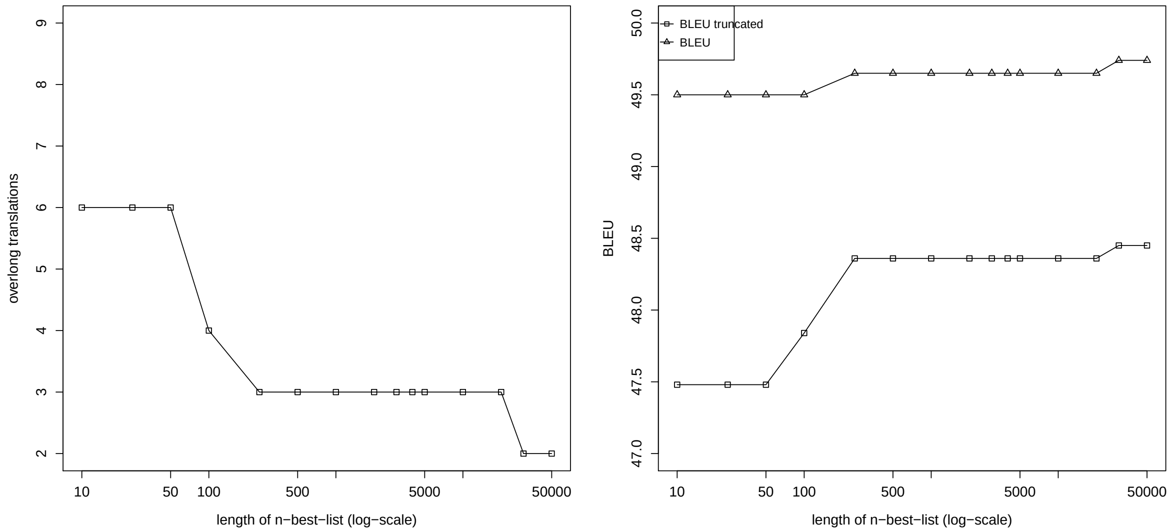

Figure 7.1: BLEU scores (right) and number of overlong lines (left) after n-best-search. All scores and numbers beat the baseline (11 overlong tweets, $\text{BLEU}_{\text{truncated}}$: 47.02, $\text{BLEU}_{\text{regular}}$: 49.40.)

training data. Twitter, however, may have a very different distribution. A translation option which has a low probability on Europarl may have a much higher probability in the Twitter domain. Consequently, hypotheses containing these "underrated" options could end up somewhere deep down in the n-best-list. Looking at the output, we see that it contained sensible corrections such as replacing "everyone" with "all" and "in the morning" with "a.m.".

The results of n-best-search on the German and English test sets with *n*=30,000 are shown in table 7.3. As a comparison, we also looked at the number of overlong lines and the BLEU scores for the baseline and the output after domain adaptation and pre-processing had been applied. We expected domain adaptation to be of some help in producing shorter output, since the in-domain tuning set contained much shorter sentences than the out-of-domain tuning set.

For German → English The length statistics support this assumption: While the baseline translation contained 17 overlong lines with an average line length of 81.19, the adapted, preprocessed output contained 14 overlong lines with an average line length of 80.25. Running n-best-search on the test set with *n*=30,000 was very successful: all but one overlong translation could be replaced with a legal tweet. There was no positive or negative effect of n-best-search on the BLEU-score. This is unsurprising, since the overlong translations only constituted a very small subset of the test set. However, we see a slight improvement in BLEU when considering the truncated versions.

For English → German, domain adaptation also caused a small reduction from 86 to 75

	overlong lines	$\text{BLEU}_{\text{regular}}$	$\text{BLEU}_{\text{truncated}}$
German → English			
baseline (no adaptation)	17	39.08	38.67
+adaptation/pre-processing	14	53.47	53.20
+n-best-search (*n*=30000)	1	53.47	53.45
English → German			
baseline (no adaptation)	86	33.99	32.57
+adaptation (tuning)	75	42.62	39.41
+n-best-search (*n*=30000)	33	42.66	40.57

Table 7.3: n-best-search on the test set

overlong tweets. N-best-search with *n*=30,000 further reduced the number of overlong tweets to 33, which reduced the overall percentage of overlong translations in the output from 17.2% to 6.6%. In both directions BLEU scores were identical or nearly identical for the detokenized output. But while for German → English the improvement in scores on the truncated output was insignificant, there was a significant improvement on the German output.

7.3 Discussion

In this chapter we presented two strategies to generate legal tweets. The first strategy – modifying the BLEU-scorer used in tuning to penalize overlong sentences by reducing n-gram precision – was shown to reduce the average character length. However, this reduction seemed to apply globally to all translated tweets and not simply to the translations which were too long. We also showed that modified tuning produced BLEU-scores which were worse than a baseline of simply truncating the output. This makes sense, since the feature weights were optimized against a criterion which differed from the evaluation criterion. We conclude that the first strategy should not be used to enforce output of a specific length.

Unlike the first strategy, the second strategy – generating n-best-lists and searching them for legal tweets – is applied only locally, i.e. it only affects translations which are too long. Experiments showed some promising results: First, we were able to reduce the number of overlong tweets from 11 (out of 18) to 2 on the development set and from 14 to 1 (75 to 33 for English-German) on the test set. Second, n-best-search was able to beat the baseline of simply truncating the output. This was explained by the different distributions of the training data and adaptation data.

However, n-best-search also has some shortcomings. First of all, there is no correlation between the length of a translation and its position in the n-best-list. This means that there is no certainty about the value of n which would be required to replace all overlong translation. Second, our experiments have shown that repairing all overlong output could require a very large value of n. But as n gets larger, translation speed becomes an issue. Since tweets are very short-lived, we would expect the need for speedy translation to outrun other concerns. Third, n-best-search does not perform equally well across languages. While n-best-search with $n = 30,000$ could repair almost all overlong tweets on the English → German task, there were still 33 overlong tweets left in the reverse direction.

A more principled approach which could be tackled in the future would be to implement an additional feature based on character count. During decoding, this feature would keep track of the character count of hypotheses. If the character count of a hypothesis exceeded a threshold value, its cost would be increased. This way, overlong translations would either be pruned or at least receive lower scores. The results of adding this feature would be similar to performing n-best-search, since the best translation of legal length would be returned. However, we would be able to save some costly computation time. This method is also more likely to show stable performance across languages.

8 Discussion

8.1 Conclusions

This aim of this study was to explore the feasibility of machine translation for Twitter.

In the first part of this project, a small bilingual corpus of 1,000 tweets was created and an analysis of its linguistic characteristics was carried out in order to identify challenges for machine translation. The comparison between Twitter and Europarl data showed that the language of the two domains was influenced by very different situational factors. While the Europarl data contain transcriptions of spoken dialogue, their linguistics features were found to be closer to written language. Twitter, on the other hand, contains originally written language which, but due to its informal setting many tweets exhibited the spontaneity and loose structure of spoken language. What is more, tweets were shown to use orthographic variation to imitate the expressive means of speech. The orthographic, lexical and grammatical characteristics were tweets were further found to be influenced by the length limit of 140 characters. The variation in spelling and word choice as well as the length limit were identified as potential challenges for statistical machine translation.

In our experimental research we found that any in-domain adaptation data we used improved translation performance. Both tuning on an in-domain development set and using an in-domain language model improved the BLEU score significantly. In-domain tuning was found to have more influence on limiting the reordering while the language model improved word choice. Adding an out-of-domain language model either did not improve or even harmed the score. The highest scores were achieved by combining both adaptation strategies.

On the Twitter test set the BLEU score was greatly affected by the way unknown words were handled. We related this problem to four factors: domain-specific English loans, code-switching, proper names and special Twitter expressions (“RT”, @usernames and hashtags). We focused on tackling the last factor, since it is specific to Twitter. By using xml-markup we could force Twitter expressions to be preserved. By replacing Twitter-specific expressions with placeholders and retraining the in-domain language model we managed to collect more robust n-gram-statistics. Combining both strategies reduced the difference in BLEU score between the +unknown and -unknown settings from 20 to 10 points.

In order to evaluate our conclusions based on BLEU scores against human judgement, we conducted a small scale human evaluation. The results outlined a tendency that for human readers excessive reordering and unwanted source language words had a greater negative effect on perceived output quality than is accounted for by BLEU. We also argued that, since BLEU's brevity penalty is simply based on word counts, BLEU is somewhat biased towards keeping unknown words.

Finally, we tackled the problem of creating translations of legal length. We found that the severity of this problem varied depending on the language pair. For translations from German into English, only 3.2% of the translations were too long, while in the reverse direction 17.2% of the translations were too long. We found a small improvement to be caused by domain adaptation. We then tested two strategies to enforce legal output length. The first strategy - modifying the BLEU-scorer which is used during minimum error rate training - could not beat a baseline of simply truncating overlong output. The second strategy successfully reduced the number of overlong tweets and produced better BLEU scores than the baseline.

We can conclude that the format of Twitter facilitates statistical machine translation, while the characteristics of the linguistic variety of tweets raise problems. However, as we tried to point out in this work, many of these problems can be solved through other means than by using in-domain training data. We showed that domain adaptation, pre-processing and n-best-search provided solutions to some of the problems. Overall, these results let us take a more optimistic stance on the perspective of machine translation for Twitter.

8.2 Future Work

Based on our work, we can outline four principal directions for further research on machine translation for Twitter.

First, out-of-vocabulary words should be reduced further. Ideally, more bilingual data should be created from Twitter or a related domain to improve the out-of-vocabulary rate. However, even without more costly training data, the out of vocabulary words could be reduced by applying spelling correction. As we pointed out, spelling errors and variations are a feature of Twitter language. We believe that identifying the standard spellings of words could improve coverage. However, the spelling correction software would need to be adapted to deal with phenomena like letter repetition ("amazingggg"), which differ from common spelling errors.

Second, we identified proper names and code switching as two causes for the difference in score between dropping and keeping unknown words. Further research could inquire into applying named entity recognition and language detection on a sub-sentence level to Twitter.

Third, more work could be done on the problem of producing legal tweets. As we mentioned in chapter 7, there are problems with our strategy of searching n-best-lists, namely its computational cost and the need to guess the value of *n*. These problems could probably be avoided by implementing a character penalty feature which causes overlong translations to be discarded during decoding.

Finally, in our guidelines on manual translation of tweets we postulated that the translation should preserve the features of the original tweet. It would be interesting to explore to what extent spelling variation, abbreviations and capitalization could be modeled and automatically generated by a translation system, in order to preserve the original character of tweets. Possibly a "translation model" between words in standard spelling and spelling variations could be estimated from the output of running a spelling corrector on Twitter data. The Twitter language model could then be used to determine likely variations or abbreviations. Being able to generate abbreviations would be another way to avoid overlong tweets.

Bibliography

(2009). New edition of ap stylebook adds entries and helpful features. AP Press Release. Retrieved from http://www.ap.org/pages/about/pressreleases/pr_061109a.html.

(2010). Big Goals, Big Game, Big Records. Twitter blog. Retrieved from http://blog.twitter.com/2010/06/big-goals-big-game-big-records.html.

(2010). Twitter snags over 100 million users, eyes money-making. Retrieved from http://economictimes.indiatimes.com/infotech/internet/Twitter-snags-over-100-million-users-eyes-money-making/articleshow/5808927.cms.

Bertoldi, N. and Federico, M. (2009). Domain adaptation for statistical machine translation with monolingual resources. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 182–189. Association for Computational Linguistics.

Blunsom, P., Cohn, T., and Osborne, M. (2008). A discriminative latent variable model for statistical machine translation. *Proc. ACL-08: HLT*.

Callison-Burch, C., Fordyce, C., Koehn, P., Monz, C., and Schroeder, J. (2007). (Meta-) evaluation of machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 136–158. Association for Computational Linguistics.

Callison-Burch, C., Osborne, M., and Koehn, P. (2006). Re-evaluating the role of BLEU in machine translation research. In *Proceedings of EACL*, volume 2006, pages 249–256. Citeseer.

Coughlin, D. (2003). Correlating automated and human assessments of machine translation quality. In *Proceedings of MT Summit IX*, pages 63–70. Citeseer.

Crawford, C. (2010). How informative is twitter? Retrieved from http://blog.textwise.com/2010/01/08/how-informative-is-twitter/.

Crystal, D. (2006). *Language and the Internet*. Cambridge University Press, 2 edition.

Farhi, P. (2009). The Twitter Explosion. *American Journalism Review*, 31(3):26–31.

Guyot, P. (2010). Half of messages on Twitter are not in English. Semiocast Press Release. Retrieved from `http://semiocast.com/publications/`.

Huang, L. and Chiang, D. (2007). Forest rescoring: Faster decoding with integrated language models. In *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, volume 45, page 144.

Java, A., Song, X., Finin, T., and Tseng, B. (2007). Why we twitter: understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 56–65. ACM.

Kelly, R. (2009). Twitter Study Reveals Interesting Results About Usage – 40% is "Pointless Babble". Retrieved from `http://www.pearanalytics.com/blog/2009/twitter-study-reveals-interesting-results-40-percent-pointless-babble/`.

Knight, K. (1999). Decoding complexity in word-replacement translation models. *Computational Linguistics*, 25(4):607–615.

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*, volume 4, pages 388–395.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5. Citeseer.

Koehn, P. (2010). *Statistical Machine Translation*. Cambridge University Press.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 177–180. Association for Computational Linguistics.

Koehn, P. and Schroeder, J. (2007). Experiments in domain adaptation for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 224–227. Association for Computational Linguistics.

Krishnamurthy, B., Gill, P., and Arlitt, M. (2008). A few chirps about twitter. In *Proceedings of the first workshop on Online social networks*, pages 19–24. ACM.

Och, F. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, page 167. Association for Computational Linguistics.

Och, F. and Ney, H. (2002). Discriminative training and maximum entropy models for statistical machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 295–302. Association for Computational Linguistics.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Osborne, M. and Koehn, P. (2010). Machine translation. Lecture slides.

Papineni, K., Roukos, S., Ward, T., and Zhu, W. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Petrovic, S., Osborne, M., and Lavrenko, V. (2010). Streaming First Story Detection with application to Twitter. In *Proceedings of NAACL*.

Stolcke, A. (2002). SRILM-an extensible language modeling toolkit. In *Seventh International Conference on Spoken Language Processing*, volume 3, pages 901–904. Citeseer.

Zhao, D. and Rosson, M. (2009). How and why people Twitter: the role that micro-blogging plays in informal communication at work. In *Proceedings of the ACM 2009 international conference on Supporting group work*, pages 243–252. ACM.