Training Machine Translation for Human Acceptability

Xingyi Song

Department of Computer Science

University of Sheffield

PhD Thesis

Feb 2015

Acknowledgements

Finally I'm in the position to submit final version of my thesis. I would like to express my special thanks of gratitude to my supervisor Lucia Specia as well as my co-supervisor Trevor Cohn. Without my supervisors I would never finish my thesis. Secondly, I would like thanks my examiners gives me the helpful suggestions, and corrections on my poor English grammar. Also, I would like to thanks all my family and friends, especially my wife, give me a lot of patient during my thesis writing.

God bless you all :)

Abstract

Discriminative training, a.k.a. tuning, is an important part of Statistical Machine Translation. This step optimises weights for the several statistical models and heuristics used in a machine translation system, in order to balance their relative effect on the translation output. Different weights lead to significant changes in the quality of translation outputs, and thus selecting appropriate weights is of key importance.

This thesis addresses three major problems with current discriminative training methods in order to improve translation quality. First, we design more accurate automatic machine translation evaluation metrics that have better correlation with human judgements. An automatic evaluation metric is used in the loss function in most discriminative training methods, however what the best metric is for this purpose is still an open question. In this thesis we propose two novel evaluation metrics that achieve better correlation with human judgements than the current *de facto* standard, the BLEU metric. We show that these metrics can improve translation quality when used in discriminative training.

Second, we design an algorithm to select sentence pairs for training the discriminative learner from large pools of freely available parallel sentences. These resources tend to be noisy and include translations of varying degrees of quality and suitability for the translation task at hand, especially if obtained using crowdsourcing methods. Nevertheless, they are crucial when professionally created training data is scarce or unavailable. There is very little previous research on the data selection for discriminative training. Our novel data selection algorithm does not require knowledge of the test set nor uses decoding outputs, and is thus more generally useful and

efficient. Our experiments show that with this data selection algorithm, translation quality consistently improves over strong baselines.

Finally, the third component of the thesis is a novel weighted ranking-based optimisation algorithm for discriminative training. In contrast to previous approaches, this technique assigns a different weight to each training instance according to its reachability and its relationship to test sentence being decoded, a form of transductive learning. Our experimental results show improvements over a modern state-of-the-art method across different language pairs.

Overall, the proposed approaches lead to better translation quality when compared strong baselines in our experiments, both in isolation and when combined, and can be easily applied to most existing statistical machine translation approaches.

Contents

List of Figures

LIST OF FIGURES

List of Tables

LIST OF TABLES

1

Introduction

Machine Translation is the process of translating one human language to another language automatically. This idea was first introduced by Warren Weaver in his memorandum called 'TRANSLATION' in 1949. In the memorandum, Warren Weaver suggested that the translation process can be treated as a decoding process. He wrote: "I look at an article in Russian, I say 'This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode.'"

Machine translation research began in the 1950s. Early machine translation systems were **rule-based**, translating input sentences based on handcrafted rules which can include bilingual dictionaries and grammars. The rule building process is normally expensive: in order to achieve a certain translation quality level, the system needs thousands of bilingual rules. Therefore, rule-based machine translation is difficult to build and extend for other language pairs.

To overcome the limitation of rule-based machine translation, in the 1990s, Brown *et al.* (1993) introduced 'statistics-based' machine translation. Statistical Machine Translation (SMT) reduces human effort to a minimum: SMT automatically builds statistical models from the analysis of bilingual (and monolingual) corpora. Equation 1.1 illustrates (Brown *et al.*, 1993)'s SMT model. Instead of following translation rules, SMT considers every possible English translation (e) of a foreign sentence (f), and assigns each possible translation a probability according to the statistical model $Pr(e|f)$. The probability indicates how likely it is that the English translation is a correct and fluent translation of the foreign sentence. The ideal translation $\hat{e}$ will be the translation

with the highest probability, which is found by solving the arg max problem using a decoding algorithm.

$$\hat{e} = \arg\max_{e} Pr(e|f) \tag{1.1}$$

Brown *et al.* (1993) decompose Equation 1.1 by using **Bayes theorem** to obtain Equation 1.2. The translation probability is calculated by the product of $Pr(f|e)$ and $Pr(e)$, where $Pr(f|e)$ is the translation model, used to assign the likelihood of the foreign words being translated as the English words, and $Pr(e)$ is the language model, used to measure the likelihood of the translation in the target language.

$$Pr(e|f) \propto Pr(f|e)Pr(e) \tag{1.2}$$

Equation 1.2 has two limitations: First, it combines the translation and language model uniformly. This setting assumes we can obtain the real probability distributions of $Pr(f|e)$ and $Pr(e)$. In real situations it may desirable to weight them differently. Since the training corpora we use to built models only represent a small sample of real world data, which can be very different from the probability distribution for the entire population, assigning different weights to different components may yield better translations. Second, the translation probability is only based on the translation and language models. Modelling translation probabilities is very difficult, so there are invariably errors and biases. Adding other components, such as a model of reordering between different languages, may help obtain better translation. However, it is not straightforward to integrate other statistical models into Equation 1.2.

In order to address these two limitations, Och & Ney (2002) proposed a Direct Maximum Entropy Translation Model (referred to here as Och's Model) We illustrate Och's Model in Equation 1.3:

$$Pr(e|f) \propto p_{\lambda_1^M}(e|f) \tag{1.3}$$

where $Pr(e|f)$ is modelled as the probability of M features $p_{\lambda_1^M}(e|f)$ in a linear combination. In this model we can obtain the translation with highest probability using the following decision rule:

$$\hat{e} = \arg\max_{e}\{\sum_{m=1}^{M} \lambda_m h_m(e, f)\} \tag{1.4}$$

where $h_m(e, f)$ are feature functions, such as language model and translation model, and λ_m is their respective weights to indicate their relative importance. Under this framework, the model weights can be learnt based on data, and additional features (such as reordering or syntactic features) can be easily integrated into the model. This is now the dominant approach in SMT research. More recent work has attempted to add more features to it to improve translation quality. Chiang *et al.* (2009), for example, includes thousands of features. Different feature weights λ_m can heavily affect the final translation quality.

In order to find the optimum weights, approaches based on Och's Model require an additional training step, the so-called **discriminative training** or simply **tuning**. The principle of SMT discriminative training is to discriminate correct translations from incorrect translations generated by the SMT system, updating the feature function weights iteratively to give most probable translations higher overall model score. The correctness of translations is measured using an automatic evaluation metric against human translations. It is expected that translations generated for unseen segments with these weights will have the best possible quality, given the model.

Algorithm 1.1 shows a general discriminative training procedure. The training corpus used is often called 'development' or 'tuning' corpus, and it contains a set of foreign sentences (the source sentences) and their respective human translations (the reference sentences). The development corpus is normally distinct from the corpus used to extract rule tables and compute feature functions, in order to simulate a real world setting and avoid over-fitting. SMT discriminative training can be described as a three-step procedure: **1. generation**, **2. evaluation** and **3. optimisation**.

The generation step creates a set of candidate translations which are then used for evaluation and optimisation. The candidate translations are a subset of all possible translations and their feature values, given the source sentence. One cannot consider all possible translations for discriminative training since the number of possible translations grows exponentially with the source segment length. To limit the number of candidate translations, a common strategy is to select the top N most likely translation

Algorithm 1.1 A general discriminative training algorithm

Require: Development corpora $D = (f^t, r^t)_{t=1}^T$, initial weights Λ_0

1: $i = 0$
2: **while** Not meet training criterion **do**
3: Generate N-best list of translation candidates $Nb = (f_n^t, r_n^t)_{t=1}^T n = 1^N$ according to $\Lambda_i H(e, f)$
4: Evaluate translation candidates by automatic evaluation $metric(e, r)$
5: Optimise Λ_{i+1}
6: $i = i + 1$
7: **end while**
8: **return** Λ_i

candidates for later steps; this list is normally referred to as the 'N-best list'. The probability (or 'model score') is calculated as in Equation 1.4 with initial feature weights manually or randomly assigned.

The evaluation step aims to measure the correctness of translations. Since the model data distribution is different from the real world distribution, the most likely translation judged by linearly combining feature scores is unlikely to be reliable. A different metric is needed to evaluate the correctness of candidate translations. This is done using automatic evaluation metrics, as thousands of translations need to be scored. Current automatic evaluation metrics compare the similarity of each candidate translation to a reference translation. The candidate that is most similar to the reference is considered the best. The set of best translations for each source segment is referred to as the 'oracle translations'.

Once all candidate translations are scored against reference translations, the third step is to optimise the feature weights. New weights will replace the initial or current ones by giving the model score the ability to discriminate oracle from non-oracle translations in order to produce the best possible translation for unseen segments. This can be achieved in multiple ways. Och (2003) minimise the errors between model score and automatic evaluation metrics, Watanabe *et al.* (2007) and Chiang (2012) maximise the margin between oracle and non-oracle translations, and Hopkins & May (2011) redefine this problem as a ranking problem, to rank correct translations better than incorrect ones.

Although various SMT discriminative training algorithms have been proposed in the last decade, many problems remain to be addressed. We discuss some of these problems in what follows:

1. Existing automatic evaluation metrics cannot always reliably compute translation quality; the evaluation judgements are often very different from human judgements, particularly at segment-level. Most automatic evaluation metrics compute, at the surface level (word or sequence of words), the similarity between the candidate and reference translations. However, most foreign sentences can be translated in multiple ways and it is impractical to list every translation in the reference set (Dreyer & Marcu, 2012). Some metrics apply deep linguistic analysis to evaluate machine translation quality, but they are language-dependent and often have relatively low correlation to human judgements.

2. Over-fitting is a common problem in machine learning and SMT discriminative training also suffers from it, especially in unregularised training algorithms such as MERT (Och, 2003), the most popular SMT discriminative training algorithm. Recent research shows that applying a regularised objective function to these training algorithms helps reduce the effect of over-fitting (Galley *et al.*, 2013). Over-fitting can also be related to the selection of development corpora: clean and diverse training corpora can help reduce over-fitting.

3. The reachability of references is also a problem. This means that the SMT system is unable to generate candidate translations for a source segment that is the same as its reference translation. This problem may be caused by several reasons. One possibility is that the words in the reference translation do not appear in the SMT training corpus, or the translation of certain words has not been extracted. It could also be because the reference is inherently wrong, which happens in crowd-sourced corpora (Smith *et al.*, 2013a). Both issues cause unreachable translations which cannot be correctly scored by automatic evaluation metrics. Therefore, we cannot learn useful information from unreachable translations to discriminate between good and bad translations, and often this harms training.

4. Another problem is related to the limitations of Och's Model. This model linearly combines a set of features in order to reduce the complexity in decoding. The

discriminative training process can be treated as a linear classification problem, as we expect the trained model to be able to classify translations as correct or incorrect. However, this problem is more complex than a simple binary classification, as translation quality is a complex, non-linear function. The problem is that non-linear SMT model would make decoding too complex computationally, and therefore linear models are still the dominant approach. Training the parameters of the linear model for each translation task can increase classification accuracy.

5. The use of N-best lists as an approximation is also a problem in discriminative training. The size of the N-best list is usually not greater than $1,000$. This is a very small number, compared to the total number of possible translations, which often totals millions, so the N-best list contains a very small subset of all possible translations (Dreyer & Marcu, 2012). Many correct translations or even the best translation may fall out of the N-best list. Therefore, we may often mislabel the oracle translations, affecting training quality.

6. The way of selecting oracle translations is another problem in SMT discriminative training. Early SMT discriminative training algorithms select the candidates with the best automatic evaluation metric score as the oracle translations. SMT decoders build the translation from smaller components (such as words or phrases), therefore one translation can be achieved by multiple ways. As a simple example, the translation 'good morning sir' could be built by using the phrases containing 'good morning' and 'sir', or 'good' and 'morning sir'. Therefore, as stated in Blunsom *et al.* (2008), the oracle should not only the best translation but also the best combination of smaller components. Chiang (2012) points out that the oracle translation selection should not consider an automatic evaluation metric score only; the model score should also be taken in consideration. In other words, the oracle should be a translation with high metric score and also high model score, i.e., likely to be generated by the model.

7. Scalability is also a problem for reliably optimising weights with large feature spaces. This is particularly an issue with the MERT algorithm, which is still used as the default discriminative training algorithm in most SMT toolkits. Scalability was not a major issue when MERT was introduced, as it was designed having as

goal to optimise only 10-15 dense feature functions. However, with the increasing number of features used in SMT models nowadays, including thousands of sparse features, scalability becomes a critical issue. Research shows that the MERT algorithm becomes unreliable if more than 15 features are used, and suggests alternative training algorithm (Watanabe *et al.*, 2007; Liang *et al.*, 2006; Hopkins & May, 2011) in those cases.

The aim of this thesis is to improve SMT discriminative training by solving or minimising some these problems. Issues 6 and 7 cannot be solved solely by improving discriminative training methods; they also require changing the decoding algorithm. Problem 5 is a particular issue for SMT approaches with a large number of sparse features, whereas our experiments are based on phrase-based SMT with only a handful of dense features. Therefore, this thesis focuses on addressing problems 1, 2, 3, and 4.

1.1 Objectives and Scope

The aim of this thesis is to improve SMT discriminative training and, as a consequence, improve SMT translation quality by addressing some of the problems discussed above. The objectives involved in achieving this aim are:

- To address Problem 1, we develop new evaluation metrics focusing on specific constraints related to their use for discriminative training, while at the same time, making sure these metrics correlate well with human judgements. Other requirements include factors influencing adoption by practitioners, such as ease of use and portability. The underlying assumption is that the components of existing evaluation metrics such as BLEU (Papineni *et al.*, 2002), METEOR (Banerjee & Lavie, 2005), TER (Snover *et al.*, 2006), and others are sound, but the way in which these metrics are formulated makes it difficult to adapt them for specific training purposes.

- To address Problems 2 and 3, we will quantify the effects of the development data selection strategy on tuning, particularly with respect to minimising over-fitting (Problem 2) and avoiding unreachable translations (Problem 3). We consider the scenario where adequate translation data for tuning may not be readily available, and seek to improve over random corpus sub-sampling by intelligent selection

methods. The assumption here is that adequate training data exists, but it needs to be appropriately selected from larger and potentially noisy collections.

- We address the weaknesses of the Och's Model (Problem 4) by designing a novel online discriminative training algorithm where the feature weights can be updated dynamically according to each source test sentence.

1.2 Research Contributions

This thesis contributes to SMT discriminative training in the following ways:

- A novel trained evaluation metric, Regression- and Ranking-based Optimisation for Sentence-level MT (ROSE) (Song & Cohn, 2011). ROSE is a trained metric that assigns weights to its features based on human judgements. The features used in ROSE include n-gram, word count and part-of-speech (POS) tags. ROSE has four variants: a regression-based approach – ROSE-reg, a ranking-based approach – ROSE-rank, and their extended versions using POTS tags – ROSE-regpos and ROSE-rankpos. ROSE-rank shows better correlation with humans than BLEU, the the most commonly used metric for ranking translations. Both ROSE-reg and ROSE-rank can be trained on languages other than the actual evaluation language.

- A novel heuristic evaluation metric, SIMPBLEU (Song *et al.*, 2013). SIMPBLEU is designed based on the limitations of the BLEU metric. It is a more flexible metric which not only correlates well with human judges, but also leads to more accurate discriminative training. In the WMT12 evaluation shared task (Callison-Burch *et al.*, 2012), SIMPBLEU showed better correlation with human judgements than any other metric for out-of English document-level evaluation, and in the WMT13 shared task (Macháček & Bojar, 2013), SIMPBLEU was also the best evaluation metric among all submitted metrics for into English and out-of English sentence-level evaluation, in addition to out-of English document-level evaluation.

- An investigation of the relationship between development corpora and SMT discriminative training quality. This includes the relationship of corpus size and

corpus diversity. Our findings include: 1) The length of the training sentence affects the training quality: overly long/short should be avoided for training. 2) Diverse training corpora reduces over-fitting. 3) Increasing training corpus size leads to very limited improvements; a corpus with 30,000–70,000 words is sufficient to train a standard phrase-based system.

- A novel data selection algorithm for SMT discriminative training based on the findings above: the LA selection algorithm (Song *et al.*, 2014). It focuses on the selection of development corpora to achieve better translation quality on unseen test data. Models trained on LA selected corpora achieved improvements of over 2.5 BLEU points in translation quality over those trained on randomly selected corpora.

- A novel discriminative training algorithm that adjusts the sampling strategy for the ranking-based optimisation algorithm PRO (Hopkins & May, 2011): Weighted Ranking Optimisation (WRO). WRO shows significant improvements over the standard PRO sampling strategy.

- A transductive learning technique for the WRO algorithm where each training sentence is weighted according to its reachability and similarity to the test sentence. This algorithm is able to optimise parameter values for each input sentence individually and leads to better translation quality.

1.3 Thesis Outline

This thesis includes six chapters. In Chapter 2 we review the existing developments in SMT discriminative training, including research in automatic evaluation metrics and strategies for development corpora selection.

In Chapter 3 we propose two novel evaluation metrics, ROSE and SIMPBLEU, which are then used as scoring function in discriminative training. ROSE is a sentence-level data-driven metric, combining word count and simple linguistic features. Additional features can be easily incorporated in the ROSE framework. SIMPBLEU is a language-independent metric which does not require training data and can work at both document- and sentence-level. In the WMT evaluation campaigns, ROSE and

SIMPBLEU achieved better correlation with human judgements than the BLEU metric, with SIMPBLEU having the best human correlation among all metrics in WMT12 and WMT13.

In Chapter 4 we analyse various aspects in development corpus selection and propose a novel development corpus selection algorithm, the LA selection algorithm. Previous development corpus selection algorithms either require knowledge of the test set (Li *et al.*, 2010; Lu *et al.*, 2008; Zheng *et al.*, 2010; Tamchyna *et al.*, 2012) or information from the decoder (Cao & Khudanpur, 2012). The LA selection algorithm does not require either of them; it relies on word-alignment and shallow linguistic information. LA's low run time requirements makes it especially suitable for large scale data selection from crowdsourced, potentially noisy translations. Our experiments show that models trained on LA-selected corpora perform significantly better than those trained on data randomly selected, and comparably to those trained on professionally created development corpora.

The new SMT discriminative training algorithm – Weighted Ranking Optimisation (WRO) – is introduced in Chapter 5. WRO is a ranking-based optimisation algorithm based on (Hopkins & May, 2011)'s PRO algorithm. Different from the standard off-line global training algorithm PRO, where a single set of weights is learnt for all test sentences, WRO is a online local training algorithm whose parameters are trained for each source test sentence. WRO is only slightly slower than the global training algorithm and its parallel computation design makes it feasible for real time translation. This chapter also puts together the data selection method, discriminative training algorithm and metrics proposed in this thesis.

Chapter 6 summarises the thesis by reviewing its key findings and results and discusses possible future work. All proposed approaches lead to better translation results individually, and we obtain further improvements by combining them.

2

Review of SMT Discriminative Training

Our goal is to improve SMT discriminative training focusing on three aspects: 1) designing better evaluation metrics, 2) designing algorithms to relevant select training data from potentially noisy parallel sources, and 3) designing discriminative training algorithms to weight each training instance according to its contribution to training. In this chapter we review the relevant background that will provide a basis for the following chapters.

The experiments in this thesis are conducted using the phrase-based SMT architecture. Therefore, our SMT literature review focuses on phrase-based approaches to translation. The discriminative training review includes four parts: 1) Training algorithms and training criteria, 2) Oracle selection strategies and related training algorithms, 3) Automatic evaluation metrics, and 4) Development data selection strategies.

2.1 SMT and Phrase-based Models

Before reviewing to SMT discriminative training algorithms, we first give a brief overview of SMT and the decoding process. Consider Brown *et al.* (1993)'s SMT model, as previously discussed in Chapter 1, which formulates the translation process as finding the string $\hat{e}$ in Equation 2.1.

$$\hat{e} = \arg\max_{e} Pr(e|f) = \arg\max_{e} Pr(f|e)Pr(e). \tag{2.1}$$

The basic components in this model are the translation model, $Pr(f|e)$, language model, $Pr(e)$, and an algorithm to solve the arg max problem. Until nowadays, these are the three most important components in SMT. In what follows we give an overview of these components and outline the candidate generation process.

2.1.1 Language Model

The language model (Equation 2.2) measures the probability of sequences of word in the target language. It is estimated using a corpus of the target language only. The probability of a sequence $e_1^I = e_1, e_2, \cdots e_I$ is calculated as the product of the probabilities of each word conditioned on all previous words:

$$Pr(e_1^I) = \prod_{i=1}^{I} Pr(e_i|e_1^{i-1}). \tag{2.2}$$

Calculating Equation 2.2 is not an easy task. Consider a 30-word long sentence. The probability of the last word has to be conditioned on the previous 29 words. Given language variability, it is unlikely that the previous sequence of 29 words will have appeared in the training corpus, which makes it difficult to model this probability distribution. Additionally, computing many such sequences is resource intensive. To simplify the problem, it is common to consider only n previous words as history. Equation 2.2 can thus be rewritten as:

$$Pr(e_1^I) = \prod_{i=1}^{I} Pr(e_i|e_{i-n}^{i-1}). \tag{2.3}$$

The model in Equation 2.3 is referred to as the 'n-gram language model', where an n-gram is a sequence of n words. This is the most common type of language model used in SMT and n typically ranges between 3 and 5. The size of n is dependent on the size of the training corpus available for training the language model.

2.1.2 Translation Model

The translation model measures the probability of a translation given a foreign text (source). This is estimated based on a corpus parallel sentences. However, the same problem occurs as for language modelling: it is unfeasible to model the probability

distribution of full sentences, given that sentences do not tend to be repeated in a corpus. Therefore, we also need break down the problem. The first generative translation model introduced by Brown *et al.* (1993) breaks this problem into words. Consider l_f is the length of the foreign input (in words), l_e is the length of target in English. The IBM models treat translation as a mapping process, where English words are mapped into foreign words. This mapping is called alignment, a. The translation probability of a foreign sentence f can be calculated as the product of the lexical probabilities, $t(f_j|e_{a_j})$, of each foreign translation in words f_j, given its aligned English words e_{a_j}. The product of lexical probabilities is then normalised by a term which reflects the number of possible alignments, $\frac{\epsilon}{(l_e+1)^l_f}$. In the IBM 1 model, each foreign word has to align to one word in the English side. In practice, languages differ and some foreign words may not have mappings to English words, for example, many function words in Chinese do not have a direct mapping to English. In this case, we align foreign words to a NULL token, which is treated as a special English word, such that the total number of alignable 'words' is $(l_e + 1)$. Therefore, the number possible alignments between f and e is $(l_e+1)^l_f$, and the sum of all possible probabilities has to be 1. The word-based IBM 1 model is formulated as:

$$Pr(f, a|e) = \frac{\epsilon}{(l_e + 1)^{l_f}} \prod_{j=1}^{l_e} t(f_j|e_{a_j}) \tag{2.4}$$

$$\epsilon \equiv Pr(l_f|f) \tag{2.5}$$

$$a_j \in [1 \cdots l_e] \cup Null. \tag{2.6}$$

The IBM 1 model is the first generative translation model. It does not take into account word order, nor the possibility of adding words to the translation. Brown *et al.* (1993) proposed several more advanced word-based models (IBM 2-5 models), which aim to handle these shortcomings. These models still provide the basis for most current SMT approaches. In phrase-based SMT, the alignments these and other word-based models produce are used as a starting point for phrase extraction. Marcu & Wong (2002); Koehn *et al.* (2003) have shown that phrase-based models are better able to handle divergences between languages, including short distance word-order, and differences in number of words. Phrase-based models break down $Pr(f|e)$ into phrases rather than words. The phrase, as proposed by Koehn *et al.* (2003), is formulated as

$$Pr(f|e) = \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i)d(.), \tag{2.7}$$

where the foreign sentence and its translation are divided into I phrases, with a phrase being a sequence of words observed in the corpus, as opposed to a linguistically motivated construction. The translation probability can be calculated as the product of each phrase translation probability $\phi(\bar{e}_i|\bar{f}_i)$ with an added reordering model $d(.)$. The phrase probability $\phi(\bar{e}_i|\bar{f}_i)$ is obtained by relative frequency counts in a bilingual parallel corpus that has been word-aligned. Phrase probabilities are stored in a 'phrase table'. Table 2.1 is an example of a phrase table, showing the English translation options for several Chinese phrases with their translation probability. This is a simplistic example. In practice, the phrase table contains other scores as well, such as the inverse translation probability and translation probabilities for words within phrases. Reordering and adding/dropping words is modelled by having multi-word phrases, where the length of phrases in the two languages can be different.

Several recent benchmarks show that the phrase-based models present state of the art or competitive performance for most language pairs (Bojar *et al.*, 2014). For this reason, in this thesis we build on phrase-based models. Other popular translation models include hierarchical phrase-based SMT (Chiang, 2007) and syntax-based SMT (Yamada & Knight, 2002), which we will not discuss in this thesis.

ZH	**EN**	**probability**
我	I	0.8
我	me	0.2
要	want	0.5
要	wish	0.5
我要吃	I want eat	0.8
要吃	want eat	0.6
吃	eat	0.9
鱼	fish	0.8
吃鱼	eat fish	0.7
...	...	...

Table 2.1: Example of phrase table

2.1.3 Decoding

In the last two sections we briefly described the language model and translation model. If we apply these two models using Och's framework to score translation candidates, the probability of an English translation is calculated as:

$$Pr(e) = \lambda_1 \times \sum_{i=1}^{I} Pr(e_i|e_{i-n}^{i-1}) + \lambda_2 \times \sum_{i=1}^{I} \phi(\bar{e}_i|\bar{f}_i)d(.). \tag{2.8}$$

The translation probabilities of all possible translations are obtained as outlined in Equation 2.8. In order to search for the highest scoring translation among them we need to solve the arg max problem in Equation 1.4. The process to solve this problem is called **decoding**. In what follows we give a brief review of (Koehn, 2004a)'s beam search decoder. This beam search decoder will be used to built the SMT systems for our experiments in subsequent chapters.

The decoding process builds the full translation sequentially from left to right by selecting possible sub-translation from the phrase table. We illustrate the example of generating the Chinese sentence '我要吃鱼' in Figure 2.1. It starts from an empty hypothesis, where a hypothesis means a partial translation. In our example, we use square box to illustrate the hypotheses, f is the foreign word covered (translated) so far, e is the translation of covered foreign words so far, P is the current partial translation probability and ID is the hypothesis identifier. The empty hypothesis is then expanded to a new hypothesis by picking a translation option in the phrase table to cover (translate) some of the untranslated foreign words. In order to allow reordering in the translation, the foreign words do not have to be covered sequentially. For example, in Figure 2.1, from the empty hypothesis we can either pick translation options for the first Chinese word '我' or for any other Chinese word. We continue expanding the resulting hypotheses in the same manner, until all the foreign words are covered (each word can only be translated once). We can then find the best translation by following links to the best hypothesis in the last stack, i.e., the hypothesis with the best score P. For example, the best hypothesis in our example is 'fish (id:14)', and tracing the link to the previous hypotheses we obtain 'I want eat (id:8)'. The best translation will thus be 'I want eat fish'. In discriminative training we are more interested in the top n best translations rather than the one best translation. We refer to this list of top translations as the 'N-best list'.

A problem with this process is that it often generates too many hypotheses: the number of possible hypotheses is $(l_f!)^m$, where m is number of possible translations in the phrase table. If every foreign phrase has 10 possible translation options, our example would have $(4!)^{10}$ hypotheses. Therefore it becomes too complex and inefficient to search for the best translation in such a large hypotheses space. There are many ways in which the number of hypotheses generated can be limited during the decoding process.

To reduce the search complexity, first we can set a reordering limitation for decoding, which can be done in several ways (Lopez, 2009). For instance, a fixed difference of up to 5 positions between the source and target word orders can be set as this limit. This constraint can in theory end up eliminating the best translation, although certain assumptions based on the language pair under consideration can be safely made. For example, one would not expect major differences in word order between languages like English and Spanish.

A second method to reduce the search complexity is called recombination: if two hypotheses have the same last n English words, the same last foreign words and the same number of foreign words covered, we can keep only the hypothesis with the highest probability and safely drop the other ones without the risk of pruning the best hypothesis. For example, in Figure 2.1, hypotheses ID 8 and ID 9 both cover three Chinese words '我要吃' and translate into the same English sequence 'I want eat'. In this case, we can drop the lower probability hypothesis, ID 8, and keep only hypothesis ID 9.

A third method for efficient search is called pruning. Pruning limits the number of hypotheses in each **stack**. Stacks are used as data structure to organise hypotheses according to the number of words covered (as illustrate at the bottom of each blue rectangle in Figure 2.1). If the number exceeds this limit, the lowest probability hypotheses will be dropped. In our example, if we limit the stack size to 3, the lowest probability hypothesis in stack 2, ID 4, will be dropped and thus no more hypotheses will be expanded from ID 4. Pruning is risky because hypotheses which contain the best translation may be dropped early based on the limited information available, leading the decoder to produce sub-optimal translations.

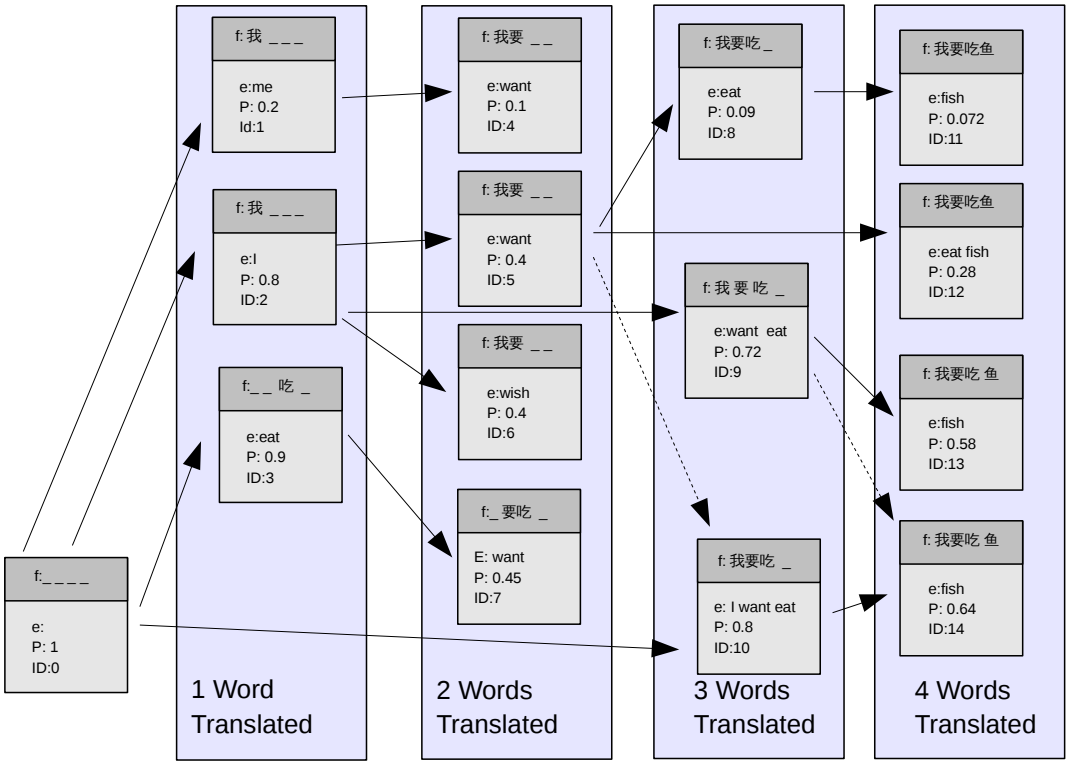


Figure 2.1: Example of the decoding process: square boxes illustrate hypotheses, f is the foreign word covered (translated) so far, e is the translation of covered foreign words, P is the current partial translation probability and ID is the hypothesis indicator.

2.2 SMT Discriminative Training

SMT discriminative training is commonly referred to as 'tuning'. As previously mentioned, it is a training step for SMT used to optimise the feature function weights. The first SMT discriminative training algorithm was a maximum likelihood approach proposed by Och & Ney (2002). Soon afterwards Och (2003) introduced the Minimum Error Rate Training (MERT) algorithm, which focuses directly on translation quality, having as target to minimise the number of errors produced by the decoder. Currently, SMT discriminative training can be categorised into maximum likelihood training, minimum error training, perceptron training, margin-based training and ranking-based training. In this Section we review the most popular discriminative training algorithms in each of these categories.

2.2.1 Maximum Likelihood Training

Maximum likelihood training (MLT) was proposed by Och & Ney (2002) as the first discriminative training algorithm for Och's Model. Similar to early SMT algorithms, MLT has been adapted from its application in speech recognition. The MLT training criterion for machine translation is shown in Equation 2.9.

$$\hat{\Lambda} = \arg\max_{\Lambda}\{\sum_{s=1}^{S} \log p_{\Lambda}(e_s|f_s)\} \tag{2.9}$$

This formulation assumes that we have a development corpus

$$\{E_1^S, F_1^S\} = \{(e_1, f_1), (e_2, f_2), \ldots (e_{S-1}, f_{S-1}), (e_S, f_S)\}$$

containing S sentence pairs, and the system has M features to be optimised, with corresponding feature weights denoted as $\Lambda = \{\lambda_1, \lambda_2, \ldots \lambda_{M-1}, \lambda_M\}$. MLT attempts to maximise the likelihood of having reference translation among candidate translations. However, this training criterion poses a problem for machine translation. Different from speech recognition, in MT there is normally not a unique good translation for a foreign sentence; the foreign sentence can be translated in multiple ways which will all count as correct translations. To address this issue, Och & Ney (2002) adapted Equation 2.9 into Equation 2.10.

$$\hat{\Lambda} = \arg\max_{\Lambda}\{\sum_{s=1}^{S} \frac{1}{R_s} \sum_{r=1}^{R_s} \log p_{\Lambda}(e_{s,r}|f_s)\} \tag{2.10}$$

The new training criterion assumes that each foreign sentence f_s contains an R_s number of references, with each reference denoted as $e_{s,r}$. In this case, the initial likelihood function in Equation 2.9 is averaged by the number of references. Equation 2.10 allows multiple reference for each training sentence, but with the current formulation of phrase-based SMT, including pruning in decoding and phrase extraction, and the size limitation of N-best lists, often none of the reference translations can be found in the N-best list, causing zero likelihood. In this case, maximising the probability of producing a reference translation is infeasible. A new training criterion was proposed to maximise the probability of the oracle translation:

$$\hat{\Lambda} = \arg\max_{\Lambda}\{\sum_{s=1}^{S} \frac{1}{R_s} \sum_{r=1}^{R_s} \log p_{\Lambda}(e_{s,oracle}|f_s)\}, \tag{2.11}$$

where the oracle translation in Och & Ney (2002) is the candidate with the minimum word error rate with respect to the reference(s). We will discuss other choices for oracle translation in Section 2.3.

Och (2003) state that maximising the oracle translation likelihood has little relation to translation quality of unseen data. (Och, 2003) proposes a new discriminative training algorithm, MERT, to replace the MLT algorithm.

2.2.2 Minimum Error Rate Training

MERT was proposed by Och (2003) to replace the maximum likelihood training algorithm. Instead of maximising the likelihood of the oracle candidate translation, MERT focuses on minimising the number errors (commonly by maximising a metric like BLEU) produced by the decoder. The latter is believed to be more closely related to translation quality. MERT is the most widely used discriminative training algorithm in SMT and therefore we adopt it as one of the baseline algorithms in subsequent chapters.

The training criterion of the MERT algorithm is given in Equation 2.12:

$$\hat{\Lambda} = \arg\min_{\Lambda}\{\sum_{s=1}^{S}\sum_{n=1}^{N} Error(r_s, e_{s,n})\}. \tag{2.12}$$

The objective in MERT is to minimise the total number of errors in candidate translations. The error function $Error(.)$ is measured by automatic evaluation metrics comparing the system output, $e_{s,n}$, against a reference translation (we will review different automatic evaluation metrics in Section 2.4). The complete process is outlined in Algorithm 2.1. To reduce computational costs, similar to MLT, MERT limits the scoring during training to an N-best list of translations. The errors in Equation 2.12 correspond to the total number of errors made by each individual translation candidate ($e_{s,n}$) in the N-best list, where n is the candidate's index in the list.

MERT applies Powell's algorithm to search for the minimum error value in one dimension (λ_c) at a time, optimising one parameter at a time while keeping the remaining parameters ($\lambda_m m \neq c$) fixed. Och (2003) adapted this search algorithm to reduce the size of the search space. The improved search algorithm only goes through the threshold points that affect the best candidate in the N-best list. Threshold points are those points where, if λ_c is changed, the top translation candidate in the N-best list

Algorithm 2.1 MERT algorithm

Require: Training data $D = (f^s, r^s)_{s=1}^S$, Initial weights Λ_0

$t = 0$, $Error_{previous} = +\infty$, $Error_{total} = +\infty$
while $Error_{previous} - Error_{total} > threshold$ **do**
  $Error_{previous} = Error_{total}$
  $Error_{total} = 0$
  Generate N-best list Nb according to Λ_t
  for Each candidate $e_{s,n}$ in Nb **do**
    Calculate error $Error(r_s, e_{s,n})$
    $Error_{total} = Error_{total} + Error(r_s, e_{s,n})$
  end for
  Update Λ_{t+1} by using Powell's search
  $t = t + 1$
end while
return Λ_t

also changes. Using the LLM formulation, the top probability candidate translation will be:

$$\hat{e} = \arg\max\{\sum_{m \neq c} \lambda_m h_m(e_x, f) + \lambda_c h_c(e_x, f)\}, \tag{2.13}$$

and because $\lambda_m m \neq c$ remain unchanged, we can denote $\sum_{m \neq c} \lambda_m h_m(e_x, f)$ as a constant $u(e_x, f)$, such that Equation 2.13 then can be written as

$$\hat{e} = \arg\max\{u(e_x, f) + \lambda_c h_c(e_x, f)\}. \tag{2.14}$$

The threshold point between candidates e_1 and e_2 is reached when the candidates have the same model score with same λ_c. This can be calculated by Equation 2.15:

$$u(e_1, f) + \lambda_c h_c(e_1, f) = u(e_2, f) + \lambda_c h_c(e_2, f) \tag{2.15}$$

$$\lambda_c = \frac{u(e_2, f) - u(e_1, f)}{h_c(e_1, f) - h_c(e_2, f)}. \tag{2.16}$$

The fact that MERT directly addresses translation quality and can be customised to use different evaluation metrics as scoring function led to a very successful adoption in SMT discriminative training. However, MERT has several limitations. These include:

Algorithm 2.2 Online Training Algorithm

Require: Training data $D = (f^i, r^i)_{i=1}^I$, Initial weights Λ_0

for t_{th} iteration K iterations **do**
  for i_{th} sentence pair in Training data D **do**
    Generate candidate Pool according to Λ_t
    Obtain oracle translations O from candidate pool
    Update Λ_{ti} towards to O
  end for
end for
return $\Lambda = \frac{\sum_{ti=1}^{KI} \Lambda_{ti}}{KI}$

1) MERT uses an N-best list as an approximation to the full space of translations, and therefore the reference or other correct translations may be missing from the N-best list. 2) The MERT objective function is unregularised and thus may overfit. 3) The loss function used in MERT (normally BLEU) is non-convex and non-smooth, which makes it unreliable for large feature spaces (more than 10-15 parameters).

To address these problems, Kumar *et al.* (2009) use lattice decoding to encode more translation candidates than an N-best list. Tillmann & Zhang (2006), Liang *et al.* (2006) and Yu *et al.* (2013) use forced decoding to force the decoder to produce the reference translation. Zens *et al.* (2007), Smith & Eisner (2006), Li & Eisner (2009) and Arun *et al.* (2010) minimise the expected error instead of real error. Gimpel & Smith (2012) proposed a structured ramp loss and convert a non-convex loss function into a convex loss function to minimise over-fitting. We will cover some of these algorithms in subsequent sections.

2.2.3 Perceptron and Margin-based Approaches

Perceptron and margin-based optimisation algorithms perform SMT discriminative training as a binary classification problem, by discriminating oracle from non-oracle translations in order to update weights such that the model is more likely to produce oracle translations. Different from MLT and MERT, the algorithms we review in this section use online training: the weights are updated after each training sentence pair. The advantage of an online approach is that we can improve the training immediately when a training instance is available. These algorithms can also be applied in off-line

training, with the final weights defined as the average of all previous weights to reduce over-fitting.

Perceptrons have been successfully applied for learning in many natural language processing tasks. Liang *et al.* (2006) applied the perceptron algorithm in SMT discriminative training. The objective is to minimise the hidden variables (such as the probability of translation model and language model) between the oracle and predicted translations. The update rule used in Liang *et al.* (2006)'s perceptron algorithm is given in Equation 2.17.

$$\Lambda_{t+1} = \Lambda_t + H(f, e_t, h_t) - H(f, e_p, h_p), \tag{2.17}$$

where the $H(f, e_t, h_t)$ is the target and $H(f, e_p, e_h)$ is the prediction. The new weights Λ_{t+1} will be updated when the prediction is not equal to the target with current weights Λ_t. In SMT, the prediction is the arg max in Och's Model and the target is the oracle translation. (Liang *et al.*, 2006) proposed three target (oracle) selection strategies, which we will discuss in Section 2.3.

$$\hat{\Lambda}_{t+1} = \arg\min_{\Lambda_{t+1}} \|\Lambda_{t+1} - \Lambda_t\|^2 + C \sum_{e_t, e_p} \xi(e_t, e_p)$$

$$s.t.$$

$$\Lambda_{t+1} \cdot (h(f, e_t) - h(f, e_p)) + \xi(e_t, e_p) \geq Loss(e_t, e_p)$$

$$\xi(e_t, e_p) \geq 0$$

$$Loss(e_t, e_p) = metric(e_t) - metric(e_p),$$

where $\xi(e_t, e_{pj})$ is a slack variable, and C is a constant to control how much the slack variable influences the objective function. The automatic evaluation function is denoted as $metric(.)$ and measures the correctness of a candidate, and $Loss(e_t, e_p)$ is the loss function of MIRA. The training objective is to keep the margin between oracle and non-oracle translations no less than the loss difference, and at the same time, make updates as small as possible to avoid over-fitting. Watanabe *et al.* (2007)'s MIRA algorithm uses an N-best list in the training by setting the oracle translations to the top k candidates in the N-best list.

MIRA shows better performance than the perceptron algorithm and scales better than MERT. However, MIRA needs to solve a number of constraints and therefore is

Algorithm 2.3 PRO algorithm

Require: Training data $D = (r^t, f^t)_{s=1}^{S}$, Initial random weights Λ_0,Γ,Ξ

1: **for** i_{th} iteration K iterations **do**
2: Sampled rank $R = \{\}$
3: **for** s_{th} sentence pair Training data D **do**
4: $s = \{\}$
5: Generate N-best list Nb according to current weight λ_i
6: **while** length(s) $< \Gamma$ **do**
7: random sample candidate pair (e_s, e'_s)
8: **if** $|metric(e_s,) - metric(e'_s,)| >$ threshold **then**
9: add $[|score(e_s,) - score(e'_s,)|, (h(f_s, e_s) - h(f_s, e'_s)))]$ to s
10: **end if**
11: **end while**
12: sort s according to $[|metric(e_s,) - metric(e'_s,)|$
13: add Ξ samples in s with top BLEU difference to R
14: **end for**
15: Update weights Λ^{i+1}
16: **end for**
17: **return** Λ^{i+1}

more difficult to implement than the perceptron or MERT algorithms, and does not outperform MERT in a standard small feature space.

2.2.4 Ranking-based Optimisation

Ranking-based optimisation treats the SMT discriminative training as a ranking problem, in which we seek to rank the translation candidates in the correct order according to their quality. The first attempt is Hopkins & May (2011)'s pairwise ranking optimisation (PRO). PRO is illustrated in Algorithm 2.3. Different from the training algorithms already reviewed, which classify the candidate into oracle and non-oracle, PRO classifies candidate pairs into 'correctly ranked' and 'incorrectly ranked'. However, enumerating all possible pairs in the N-best is impractical, even with a small 100-best list the number of pairs is still be impractical. PRO proposes a sampling strategy to avoid that problem. This strategy is shown from Line 3 to Line 14 of Algorithm 2.3.

The sampling process first selects a random pair Γ from pairs of candidate in the N-best list and then generates samples according to Equation 2.18.

$$h(e,f) - h(e',f) = \begin{cases} 1 & \text{if } metric(e) - metric(e') > 0 \\ -1 & \text{if } metric(e) - metric(e') < 0 \end{cases} \tag{2.18}$$

which computes the feature vector difference between two candidates. In Equation 2.18, we assume candidate e has higher model score than candidate e'; the sample is labelled positive if the candidate e also has a higher automatic evaluation score than candidate e' (i.e., it is correctly ranked), and negative if e has a lower automatic evaluation score than e' (i.e., it is incorrectly ranked).

Two problems with PRO are that 1) The automatic evaluation metric is not really a 'gold-standard', i.e., the metric is not 100% reliable and often very different from human judgements; and 2) Often the candidates in the N-best list are very similar to each other. Therefore, the algorithm may mislabel some of the samples if the candidate pairs are similar (and thus score bad translations higher and than a good translation). To minimise this problem, PRO uses a threshold to filter out pairs with very little difference in metric scores, only producing Ξ pairs with the largest metric score difference from the initially selected Γ pairs. With the selected samples, the new weights can be optimised by any off-the-shelf classifier. Hopkins & May (2011) use the maximum entropy model optimisation. Bazrafshan *et al.* (2012) suggests changing sample labels to real numbers (Equation 2.19):

$$h(e,f) - h(e',f) = score(e) - score(e'), \tag{2.19}$$

where weights are optimised by linear regression to achieve faster convergence.

In Hopkins & May (2011)'s experiments, PRO-trained systems show better translation performance (in BLEU score) than MERT- and MIRA-trained systems with both high and low feature space dimensionality. In addition, PRO can be easily applied to several SMT approaches. Therefore, we consider the PRO algorithm as another baseline training algorithm in subsequent chapters. One issue with PRO is its sampling strategy: uniformly sampling is not the optimum way to select sample pairs, as we will discuss in Chapter 5.

2.3 Oracle Selection and Related Training Algorithms

In the last section we reviewed some of the most popular discriminative training approaches in SMT. Apart from MERT and PRO, all existing algorithms use oracle translation(s) as the target for classification. This section will review the state of the art oracle selection strategies and the discriminative training algorithms related to the selection strategy.

The first SMT discriminative training algorithm – MLT – maximises the probability of reference candidates (candidates with same translation as the reference translation), but is severely limited by the reachability issue, i.e. the reference may not be in the N-best list, or is not in the space of translation candidates, and thus cannot be found by the decoder. MLT compromises by targeting the lowest error candidate instead of the reference. This compromise is the first type of oracle translation: the candidate with the highest metric score. The most widely used automatic evaluation metric is the BLEU metric, so we call this type of oracle **maxBLEU oracle**.

The maxBLEU oracle strategy has been adopted by many training algorithms such as Watanabe *et al.* (2007)'s MIRA and Liang *et al.* (2006)'s perceptron. However, in cases where reference candidates can be found by decoder, one just needs to ensure the reference candidates are listed in the N-best list and then use directly the reference candidate as the oracle. For that, Liang *et al.* (2006) adjust the decoding algorithm to force the decoder to produce the reference translation (if the reference is reachable) and set the oracle translation as the reference candidate. We call this type of oracle the **reference oracle**. Although the reference oracle approach cannot be widely applied in all cases, it directly focuses on the reference translation and reduces the risks of errors due to suboptimal evaluation metrics.

Chiang *et al.* (2008b) proposed a different approach for oracle selection: the oracle should not only depend on the metric score or the reference translation, but also consider the model score. If the maximum BLEU candidate or reference candidate is very difficult to generate by the decoder (i.e., it can only be generated with a very low model score), the updated weights targeted on this candidate may be too extreme. To avoid this problem, the oracle should contain low error and it should be easy enough for the model to move it to the top of the N-best list. Chiang *et al.* (2008b)'s oracle selection strategy, here called **mixed oracle**, is defined as:

$$e^{oracle} = \arg\max_{e}(metric(e) - \mu(metric(e) - model(e))), \tag{2.20}$$

where $metric(e)$ is the metric score of candidate e, $model(e)$ is its model score, and μ is a constant to control the reliability on the model score. If $\mu = 0$, Equation 2.20 is the maxBLEU oracle, and if $\mu = 1$, it is the 1-best candidate according to the model score. Chiang *et al.* (2008b) suggest $\mu = 0.5$. Based on the mixed oracle selection strategy, Gimpel & Smith (2012) proposed a ramp loss function that can be used in both perceptron and margin-based training algorithms. It considers a loss function that incorporates both metric and model scores as components for oracle selection.

The three oracle selection strategies select certain translations based on the reference and/or model score. However, in Section 2.1.3 we showed that a translation can be achieved in multiple ways (the so called derivation d). Therefore, Blunsom *et al.* (2008) state that targeting only a certain translation is not enough, we should also target the right way to produce the translation, or the **oracle derivation**. The difficulty there is that we normally have a reference translation to help select oracle candidate translations, but we do not have reference derivations. To solve this problem, Blunsom *et al.* (2008) treat the derivation d as a latent variable, where the probability of a candidate translation e is the sum of all its derivations ($\Delta(e, f)$):

$$p_\Lambda(e|f) = \sum_{d \in \Delta(e,f)} p_\Lambda(d, e|f). \tag{2.21}$$

The conditional probability of d is calculated by Equation 2.22:

$$p_\Lambda(d, e|f) = \frac{\exp \sum_m \Lambda_m H_m(d, e, f)}{Z_{\lambda_1^M}(f)}, \tag{2.22}$$

where $H_m(d, e, f) = \sum_{r \in d} h_m(f, rule)$ is the sum of the feature scores of all translation rules ($rule$) used in the derivation d, and the probability is normalised by $Z_{\lambda_1^M(f)}$, which is the sum of all derivation feature scores ($\exp \sum_m \Lambda_m H_m(d, e, f)$). The $\Delta(e, f)$ in Equation 2.21 represents all derivations of the translation e for the foreign sentence f.

With the latent variable model, Blunsom *et al.* (2008) maximise the posterior probability of the reference candidate translation with all its derivations, subject to a Gaussian prior ($p_0(\Lambda) = exp(-\Lambda_k^2/2\sigma^2)^2$):

$$\hat{\Lambda} = \arg\max_{\Lambda} p_{\Lambda}(\{E_1^S, F_1^S\})p(\Lambda). \tag{2.23}$$

Up to now our review of the four existing oracle selection strategies, for the reference oracle and oracle derivations targeting the candidates that are exactly the same as the human reference, it may not be possible to produce the reference candidates for all training instances, which are thus discarded, resulting in a far from efficient use of the data available. maxBLEU oracle and mixed oracle do not require the oracle to match exactly the reference translation, instead choosing a candidate with the fewest errors. This improves the use of data, but risks reliance due to inaccurate evaluation metrics. Yu *et al.* (2013) proposed an oracle selection strategy targeting partially reachable reference candidates. This approach is based on the assumption that the decoder may be unable to produce the full reference translation for a sentence, but can normally produce partial reference translations. For example, in Table 2.2 Candidate 8: '**torrential rain disaster kills** file action count alignment each'. This translation does not exactly match references, but if only consider the boldface portion, that is an exact match. These partial reference translations are called 'y-good' translations, and the non-matching partial translations are called 'y-bad' translations.

(Yu *et al.*, 2013) use 'y-good' translation oracles with the perceptron algorithm in SMT discriminative training. In order to use these partial translations they proposed two novel updating strategies: **early update** and **max-violation update**. Different from (Liang *et al.*, 2006)'s perceptron, (Yu *et al.*, 2013) update the feature weights before the full candidate translation is generated.

The full candidate translation is generated by series hypotheses expansion steps, and pruning will result in only keeping a limited number of hypotheses with the highest model score in the beam. If the 'y-good' translation fall out of the beam, the early update strategy will stop decoding and update the weights to reward the 'y-good' translation and penalise the highest model score ('y-bad' translation). The max-violation update does not stop the decoding but uses a trace back step after decoding to find the step where 'y-good' and 'y-bad' have the largest difference and updates the weights based on these partial translations.

The early update and max-violation update strategies efficiently use all available data and do not rely on an evaluation metric to approximate oracles, avoiding risks of errors due to evaluation metrics. However, similar to the reference oracle approach,

Example 1
Source: 印度西部豪雨成灾四十三人丧生
Reference 1: torrential rain disaster kills 43 people in western india
Reference 2: heavy rains plague western india leaving 43 dead
Candidate 1: **torrential rain disaster**
Candidate 2: torrential torrential torrential **torrential rain disaster**
Candidate 3: **rain torrential disaster**
Candidate 4: **torrential rain disaster kills 43 people in western india**
Candidate 5: **43 people killed in disastrous torrential rain in western india**
Candidate 6: **heavy rains plague western india leaving 43 dead**
Candidate 7: **torrential rains hit western india , 43 people dead**
Candidate 8: **torrential rain disaster kills file action count alignment each**

Table 2.2: Example of two English reference translations and seven candidate translations for Chinese source

these y-good oracle also require forced decoding to produce y-good translations, and thus cannot be used with some decoders that cannot perform forced decoding. In addition, a sentence can be translated in multiple ways. An 'y-bad' translation may be not a bad translation at all, so applying 'y-good' oracle selection in early updates could penalise a correct translation.

In this section we reviewed five oracle selection strategies that are used in current discriminative training algorithms. Among these, maxBLEU has wider applicability than the other oracle selection strategies, as it can be applied with any kind of SMT decoder. The problem of maxBLEU oracle is that it is affected by the accuracy of the automatic evaluation metric used. In the next section, we will review current developments in translation evaluation metrics.

2.4 SMT Evaluation Metrics

Automatic evaluation metrics are used to assess the machine translation quality. Ideally, they should provide an assessment that is as close to what a human would do,

considering both the **Adequacy** and **Fluency** of the translation. Adequacy refers to how much of the source text meaning is conveyed by the translation. Fluency refers to how grammatical and readable the translation is.

Most of the popular discriminative training algorithms require an evaluation metric to compute their loss function. The WMT11 (Callison-Burch *et al.*, 2011) Tunable Metrics Task shows that using the same discriminative algorithm to optimise a different evaluation metric can lead to a training quality difference of up to 10%. That is, different metrics penalising different aspects of translation quality result in different training performances. In this section, we will review some of the most widely used machine translation evaluation metrics. In general, we can categorise them into: **word-based** and **linguistically motivated**. The **word-based** approaches measure word similarity between the candidate and reference segments without considering any deeper linguistic information.

Early word-based metrics for SMT evaluation are based on the **Word Error Rate** (WER) metric, which was first introduced for speech recognition. Unlike speech recognition, machine translation evaluation has to account for differences in word order between the translation candidates and the reference, a problem that does not exist in speech recognition and is poorly addressed by WER. In addition, multiple equally good outputs are possible in machine translation. Although one we can create multiple references to improve the evaluation quality, this is an expensive process and in practice it is virtually impossible to list all valid translations for each test sentence.

Papineni *et al.* (2002) introduced a metric especially designed for machine translation evaluation, BLEU. Instead of measuring single word level matching between reference and system output, BLEU takes into account longer n-grams, offering a more flexible and reliable strategy to measure the similarity between the candidate and reference translation. BLEU is a precision-oriented metric and uses a brevity penalty as an proxy for recall. METEOR (Banerjee & Lavie, 2005) considers both precision and recall and uses stemmers, WordNet and paraphrase dictionaries to account for inexact word matches such as synonyms. More details about these metrics will be given in Section 2.4.3.

Some evaluation metrics include **deeper linguistic information** to evaluate translation quality. For instance, TESLA (Liu *et al.*, 2010) includes part of speech (POS)

tags and lemmas, while Liu & Gildea (2005) propose a tree based metric which evaluates translation quality at syntactic level. Many of these metrics follow a machine learning approach to combine multiple components with various linguistic information (Corston-Oliver *et al.*, 2001; Quirk, 2004; Albercht & Hwa, 2008; Specia *et al.*, 2009; Blatz *et al.*, 2004) or decoder information (Specia & Gimenez, 2010). In next sections, we will review a few popular evaluation metrics in detail.

2.4.1 Word Error Rate Metrics

Word error rate (WER) is the earliest automatic evaluation metric used in machine translation. We review WER and two improved variants: position-independent error rate (PER) Tillmann *et al.* (1997) and translation edit rate (TER) Snover *et al.* (2006).

The original WER formulation measures the Levenshtein distance between words in the candidate and reference translations. WER is defined as the ratio between minimum number of errors to the number of words in the reference, i.e., the minimum distance between the two versions. Errors include substitutions, deletions and insertions:

$$WER = \frac{min(S + D + I)}{N} \tag{2.24}$$

where S is the number of substitutions, D it the number of deletions, I is the number of insertions and N is the number of words in the reference.

Table 2.2 shows a list of candidate translations with two reference translations for the Chinese sentence '印度西部豪雨成灾四十三人丧生' (Heavy rains plague western India leaving 43 dead). Figure 2.2 illustrates the counts of errors in WER. For translation candidate '43 people killed in disastrous torrential rain in western india', one can either count 4 deletions and 5 insertions, or 6 substitutions and 1 insertion, but the latter has fewer errors and therefore is chosen, resulting in WER = 7/9.

WER works well in speech recognition as an ideal recognition candidate should be identical to the reference. However, we can translate one foreign sentence into English in many different ways. These translations may use different words (synonyms), or be written with different word orders, but still be correct and grammatical. Considering the examples in Table 2.2, Candidates 4, 5 and 6 are perfect translations for the Chinese sentence '印度西部豪雨成灾四十三人丧生' while Candidates 1, 2 and 3 are poor translations which miss important information.

Reference :	torrential	rain	disaster	kills	43	people		in				western	india	
Candidate :					43	people	killed	in	disastrous	torrential	rain	in	western	india
Errors:		D	D	D	D			I		I		I	I	I
Reference :	torrential	rain	disaster	kills	43	people		in	western	india				
Candidate :	43	people	killed	in	disastrous	torrential	rain	in	western	india				
Errors:	S	S	S	S	S	S	I							

Figure 2.2: Example of WER, 'D' indicates a deletion, 'I' indicates an insertion and 'S' indicates a substitution. The example candidate has 4 deletions and 5 insertions (9 errors) or 7 substitutions and 1 deletion (8 errors). The latter has fewer errors and is therefore used as the error count, so the WER for this candidate will be 7/9

If we only provide Reference 1, the WER score for Candidate 4 is 0, meaning a perfect translation without any error. However, Candidate 5 has WER 7/9, and Candidate 6 has WER 9/9 (8 substitutions and 1 deletion), which means a completely incorrect translation. Candidates 1 and 2, both with WER 6/9, are incorrectly considered better translations than Candidates 5 and 6.

From the example above, we can see the WER is very unreliable as a measure of translation quality. To address this shortcomings, Tillmann *et al.* (1997) extended WER into a position-independent word error rate (PER) metric for machine translation evaluation. PER addresses the problem of differences in word order in translation by **ignoring the position in which words are translated.** PER's computation includes the number of words in the candidate which are different in the reference (substitutions), the number of words in the reference that do not belong to the candidate (deletions), and the number of words in the candidate that do not appear in the reference (insertions):

$$PER = -\left(\frac{Error - max(0, (T - N))}{N}\right), \tag{2.25}$$

where $Error$ is the total number of errors (substitutions + deletions + insertions) of a candidate. PER limits the score to range between 0 and 1 by capping it to 1 if the candidate length (T) is longer than the reference length (N). As an example, let us apply PER to score candidates in Table 2.2. Based on Reference 1, the PER scores for Candidates 1-6 are: 6/9, 6/9, 6/9, 0/9, 4/9, 6/9 respectively. Candidate 4 is scored the best. In this example, PER does better than WER but still cannot adequately

Reference 1:	torrential	rain	disaster	kills	43	people		in western india
Candidate 5:	torrential	rain	people	killed	43	in	disastrous	in western india
	(shift)	(shift)	sub	sub	(shift)	sub	ins	

Figure 2.3: Example of TER, 'I' indicate a insertion and 'S' indicate a substitution. Compared to WER (Figure 2.2), by shifting 'torrential rain' to the beginning of the sentence and '43' after 'people killed', the error is reduced to 3 substitution and 1 insertion, plus 2 shifts, resulting in a TER score of 6/9

discriminate a good quality translation such as Candidate 6 from poorer candidates (Candidates 2 and 3). This issue is addressed by Snover *et al.* (2006), with another variant of WER: the Translation Edit Rate (TER) metric. TER measures translation quality with multiple references, and is defined as shown in Equation 2.26.

$$TER = \frac{\min_{shift}(S + D + I)}{\text{average(N)}}. \tag{2.26}$$

The TER score is obtained by the number of edits against average number of reference words. Similar to WER, the edits include insertions, deletions and substitutions, and **additionally the 'shift' operation** to model permutation of blocks of words: a sequence of words of any length which is found in a different position to that of the reference is considered a single edit to avoid multiple errors being counted because of phrase reordering in machine translation.

Considering References 1 and 2, according to the TER metric, Candidates 4 and 6 in Table 2.2 can be distinguished from poor translations. Candidates 4 and 6 are identical to References 1 and 2, respectively. By allowing the shift of words (Figure 2.3), Candidate 5 has a TER score of 6/9.

PER and WER are more suitable metrics for machine translation evaluation than WER, but the treatment given by PER and TER to word order is not adequate to measure the fluency of the candidates, an important criterion when measuring the quality of translations. For example, in Table 2.2, both Candidates 1 and 3 are poor translations, but Candidate 1 is clearly better than Candidate 3 in terms of fluency. In the next Section, we will review n-gram-based metrics that can offer a better treatment for both fluency and adequacy.

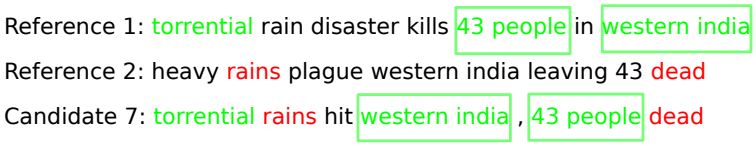


Figure 2.4: Example of n-gram precision: green words indicate a unigram match to Reference 1 and red words indicate a unigram match to Reference 2. A green box indicates a bigram match to Reference 1

2.4.2 N-gram-based Metrics

Instead of measuring only word matches, n-gram-based metrics consider matches of sequence of n words. The most famous n-gram-based metric is BLEU (Papineni *et al.*, 2002). BLEU measures the clipped n-gram precision (with n normally equal to 1-4) between a candidate translation and one or more human authored reference translations. BLEU is composed by three components: **n-gram precision**, **clipping** and **brevity penalty**:

$$BLEU = BP \times \exp\big(\sum_{n=1}^{N} w_n log(p_n)\big), \tag{2.27}$$

where p_n is the n-gram precision, w_n is the relative weight of each n-gram precision and BP is the brevity penalty. **N-gram precision** is the ratio between the number of matched n-grams and the total n-grams in candidate translation, measured for all candidates in the document:

$$p_n = \frac{\sum_{C \in \{Candidates\}} Count_{clip}(\text{n-gram}_{matched} \in C)}{\sum_{C' \in \{Candidates\}} Count(\text{n-gram}_{total} \in C')}, \tag{2.28}$$

where $Count_{clip}(\text{n-gram}_{matched})$ is the number of clipped n-grams in the candidate that also appear in the reference (see 'clipping' below), and $Count(\text{n-gram}_{total})$ is the number of n-grams in the candidate.

One advantage of measuring precision as opposed of recall is that it makes it easier to evaluate translations based on multiple reference translations. Figure 2.4 shows an example of using multiple references in n-gram matching. The coloured words indicate the unigram (word) matches; green indicates words in Candidate 7 matching words in Reference 1, and red indicates words matching Reference 2. The boxes indicate bigram

matches between the candidate and the references. In this example, the candidate matches unigrams 'torrential', 'western', 'india', '43' and 'people' with Reference 1, and in addition matches 'rains' and 'dead' with Reference 2. The unigram precision will be 7/9. There only two bigrams, '43 people' and 'western india', which match Reference 1 (and Reference 2), so the bigram precision will be 2/8.

Clipping aims at penalising over generated words in candidates, such as in Candidate 2 in Table 2.2. Candidate 2 is a poor translation that misses large amounts of information and repeatedly generates the word 'torrential'. However, it still has a very high unigram precision (6/6). To avoid this type of sentence being rewarded, BLEU limits the maximum count of an n-gram match to the maximum number of times the n-gram occurs in any of the reference translation. In our example, the unigram 'torrential' only appears once in Reference 1. Therefore unigram matches of 'torrential' can only be counted once. In this case the clipped unigram precision for Candidate 2 will be 3/6.

The **brevity penalty (BP)** component penalises translations which are too short. BLEU is a precision-based metric since the denominator is the number of n-grams in the candidate translation. Therefore, without a brevity penalty, BLEU would be biased towards shorter candidate translations. Consider Candidate 2 in Table 2.2, which only translates a small part of the Chinese source sentence '豪雨成灾', and misses most of the source information. Assume we only use up to trigram precision, without the brevity penalty, the uni-, bi- and trigram precision are all one, resulting in the sentence scoring as a perfect translation. However, this is a poor translation and the brevity penalty downgrades its score based on its length relative to the reference's length. The brevity penalty is defined as

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-\frac{r}{c})} & \text{if } c \leq r, \end{cases} \tag{2.29}$$

where c is the candidate sentence length and r is the reference sentence length. If multiple references are used we choose between average reference length or the reference with closest length to that of the test candidate translation. The penalty increases exponentially as c become shorter than r.

Until nowadays, BLEU is the most widely used evaluation metric and is commonly applied in the SMT discriminative training algorithms as a loss function. However,

BLEU is designed to measure translation quality at the document level, and has been shown unreliable for sentence-level evaluation. Consider again the example in Figure 2.4. In this example, the candidate does not have 3- and 4-gram matches with either of the references. This will cause the overall BLEU score to be zero, which is uninformative. Evaluation at the sentence level is required in many discriminative training algorithms, such as PRO and MIRA. A simple solution to overcome this issue and make BLEU more applicable to sentence level evaluation is the add-α smoothing strategy. This strategy adds a small value (e.g. $\alpha = 1$) to both the numerator and denominator in the n-gram precision computation to avoid obtaining zero precision scores for longer n-grams. Note that different α values will affect the accuracy of BLEU, as we will discuss in Chapter 3.

A major advantage of the BLEU metric is its ability to consider n-grams rather than individual words to measure fluency as opposed to adequacy only. However, it has three major well known limitations. The first is the use of the brevity penalty as an approximation of recall. Consider the candidates in Table 2.2. Candidates 1 and 2 deliver the same amount of information, but the over generated word in Candidate 2 breaks the grammar rules of English. However, with the brevity penalty the 1-3-gram BLEU score for Candidate 1 is 0.135, while Candidate 2 has a BLEU score of 0.698, indicating that Candidate 2 is a much better translation than Candidate 1.

The second problem is that using higher order n-grams tends to over-bias towards fluency. For example, in Table 2.2 unigram to 4-gram precision for Candidates 7 and 8 are: 0.78, 0.25, 0, 0, and 0.44, 0.375, 0.285, 0.16 respectively, making the overall geometric mean n-gram precision of 0 for Candidate 7 and 0.29 for Candidate 8. Even if we add a smoothing value $\alpha = 1$[1], the overall n-gram precision for Candidates 7 and 8 are 0.26 and 0.39. Thus Candidate 8 would be judged a better translation than Candidate 7. In fact, Candidate 8 is a very poor translation missing almost half of the information in the original Chinese sentence; Candidate 7 is much better. Lin & Och (2004) and Zhang *et al.* (2004) analyse the contribution of each n-gram order to the overall BLEU measure. They found that unigrams and bigrams account for 95% of the overall precision, and thus adding higher order n-grams is unnecessary.

[1]The smoothed unigram to 4-gram precision for Candidates 7 and 8 are: 0.8, 0.33, 0.125, 0.14, and 0.5, 0.44, 0.375, 0.286 respectively.

The third problem of BLEU is that fails to account for synonyms and verbal inflections, such as words 'rains' and 'plague' in Candidate 6, which express the same or similar meaning as 'rain' and 'disaster' in Reference 1, but are considered as mismatches. BLEU handles this problem by adding multiple references, such as the use of both References 1 and 2 in the example. However, multiple reference require more human effort and are difficult and costly to obtain in many cases.

2.4.3 Metrics with Shallow Linguistic Information

This section will review some of the metrics that consider not only the matching of words or n-grams, but also the matching of synonyms, part-of-speech (POS) and inflections of words. We categorise these variants of linguistic information as shallow as they are still mostly limited to word-level processing and do not require significant computation time to be extracted. Metrics of this type include wpBLEU (Popović & Ney, 2009) which measures BLEU over both words and POS tags; TESLA (Liu *et al.*, 2010) and MaxSim (Chan & Ng, 2008), both weighed n-gram-based approaches, with weights given by analysis of POS tags, synonym and phrase-level semantics; and METEOR (Banerjee & Lavie, 2005) – which uses stemming, WordNet and paraphrase dictionaries to consider word inflections, synonyms and paraphrases. METEOR is the most popular of these metrics.

METEOR was designed to address the shortcomings of the BLEU metric. METEOR only considers unigram overlap between the candidate and the reference. It uses F-score to combine precision and recall to measure translation adequacy, while fluency is measured by a fragmentation penalty. The unigram matchings can be exact, consider stems, synonyms or paraphrases. METEOR is defined as:

$$Score = \frac{10PR}{R + 9P} * (1 - Penalty), \tag{2.30}$$

and

$$Penalty = 0.5 \times \frac{\#chunks}{\#\text{unigrams matched}} \tag{2.31}$$

where P is unigram precision, R is the unigram recall and $Penalty$ is a component to penalise matching of short n-grams.

METEOR calculates its final score in three steps. The first step is alignment, where the words in the candidate and the reference sentences are aligned. Each word

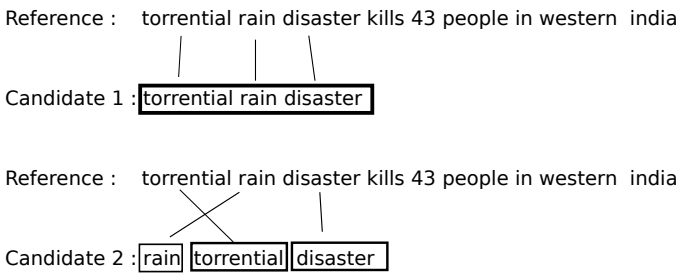


Figure 2.5: Example of METEOR alignment. A line between the candidate and the reference indicates an alignment and box indicates a chunk. Both candidate have three alignments but Candidate 1 has one chunk, and therefore would be considered a better translation than Candidate 2 according to METEOR

can only be aligned once and the aligned words have to meet one of the following three conditions: 1) the word in candidate is identical to the word in reference, 2) the word stems in the candidate and reference match, or 3) the words in the candidate and reference are synonyms or paraphrases. The last two matches require WordNet to search for stems and synonyms.

The second step groups all aligned words into *chunks*, where a *chunk* is the largest n-gram of non-grouped words in the candidate that aligns to the reference. For example, Figure 2.5 shows the alignment and grouping step of METEOR, where a box of words represents a *chunk*. Both candidates have three aligned words to the reference, but Candidate 1 has them in a single *chunk* as the n-gram 'torrential rain disaster' also appears in the reference. Candidate 2 has three *chunks* as the largest n-gram matching against the reference is for unigrams only: 'rain', 'torrential' and 'disaster'. Each of these constitutes a *chunk*.

The final step of METEOR is to calculate the penalty in Equation 2.31. The overall score is then given according to Equation 2.30.

2.4.4 Trained Metrics

The metrics previously described measure machine translation quality based on the overlap between words in the candidate and reference translations. Although metrics like METEOR and TESLA also include linguistic information, this is still done at the word or phrase level, and therefore they do not capture syntactic information and other contextual relationships. These metrics are efficient to compute in large sets

of translations, which is crucial for discriminative training. Additionally, they can be easily applied or adapted to different languages. However, in order to obtain more accurate evaluation, we may need analyse deep linguistic information such as syntactic or semantic trees. Various metrics have been designed that incorporate deep linguistic informations. Liu & Gildea (2005) propose tree-based metrics which count matchings at the sub-syntactic tree level. Corston-Oliver *et al.* (2001) train a decision tree to determine whether the candidate is machine translated (poor translation) or human translated (good translation). 46 features were used, some of which (such as tree branching features) require a POS tagger and parser. Although Corston-Oliver *et al.* (2001)'s decision tree metric is not very applicable to distinguish translation quality among several machine translated candidates, the approach started a new direction in machine translation evaluation research under which many features that represent different aspects of quality can be combined.

Most of the subsequent work on trained metrics tends to use regression-based models to obtain continuous scores rather than only binary scores. These are trained to predict various types of scores and therefore avoid the need for reference translations at evaluation time. For example, Blatz *et al.* (2004) combine a number of features reflecting the confidence of the translation system, the complexity of the source segment and the fluency of the translated segment, among others, to predict BLEU/NIST/WER scores, while Specia *et al.* (2009); Specia & Gimenez (2010) predict human scores for post-editing effort. Trained approaches outperform reference-based metrics in terms of correlation with human judgements.

2.5 Development Data Selection

In SMT, as in many learning tasks, the accuracy of the models is heavily dependent on the training data used to build them. Two factors are particularly important. The first is the domain of the training data and its similarity to the domain of the test data. If the training data is drawn from the same source as the test data, we expect to obtain better translation accuracy. The second is the level of noise in the training data. Sources of noise in discriminative training include misaligned training sentence pairs and unreachable target sentences. Training quality normally increases if clean data is used by filtering out these sources of noise. In this section we will review research on 1)

selecting development corpora based on a given test set and 2) selecting development corpora without knowing what the test set will be.

2.5.1 Development Data Selection with Test Set

Over-fitting is a common problem in discriminative training whereby the learned parameters are able to discriminate training data well, but fail to discriminate unseen (test) data. The problem can be reduced by improving the training algorithm, or by selecting training data that is more similar to the test set. The assumption of this approach is that we have large data pool of potentially relevant training instances D_F, and we know the data that needs to be translated, i.e., the test data T. In addition, we assume that discriminative (re-)training is possible before translating *each* test set. During retraining, we can select the subset of D_F which is most similar to T as the training data, i.e.:

$$D^* = \arg\max_{D \subseteq D_F} Sim(D, T). \tag{2.32}$$

The problem of this setting is how to define the similarity function $Sim(.)$. A straightforward method is to apply one of the evaluation metrics introduced before to measure the similarity between sentences in the test set and development data pool. However, word based metrics such as BLEU and WER only measure the overlap between words without weighting the importance of the words. Function words and punctuation overlap between sentences do not reliably indicate similarity. Therefore, previous work has applied information retrieval techniques, which measure the similarity between sentences by the cosine distance of Term Frequency and Inverse Document Frequency (TF-IDF) (Lu *et al.*, 2008; Hildebrand *et al.*, 2005). TF-IDF represents each sentence as a vector containing m vocabulary terms $W = \{w_1, w_2, \cdots, w_m\}$. The TF-IDF of w_i in a sentence will be calculated as

$$w_i = \text{tf}_i \times log(\text{idf}_i), \tag{2.33}$$

where tf_i is number of occurrences of w_i in the sentence, and idf_i is calculated by

$$\text{idf}_i = \frac{\text{number of sentences}}{\text{number of sentence containing } w_i}. \tag{2.34}$$

TF-IDF provides a fast and weighted similarity metric, but it only considers the similarity at word-level. Eck *et al.* (2005) and Liu *et al.* (2012) combine TF-IDF with evaluation metrics that are also able to measure weighted similarity of sequences of words.

Similarity scores given by both TD-IDF and evaluation metrics are based on the assumption that two sentences are similar if they have the same words or sequences of words. (Li *et al.*, 2010) however suggests that for SMT training the similarity should be based on what the SMT model believes is similar. Li *et al.* (2010) define a new similarity function where the similarity of two sentence f_1 and f_2 is measured by the similarity of their feature score vectors, where $H(\hat{e}, f)$ is the feature score vector of the most probable translation candidate $\hat{e}$ for f. The drawback of this approach is that is requires translating the entire pool of sentences that could be candidate for the development dataset.

$$Sim(f_1, f_2) = Sim(H(\hat{e}, f_1), H(\hat{e}, f_2)), \tag{2.35}$$

2.5.2 Development Data Selection without Test Set

Similarity to the test set is not the only important criterion to define a good quality development dataset. Additionally, in a more realistic setting we do not know the test sets which will be used at system training time. In principle the system could be used to translate any new text. Therefore, selecting good quality development data without access to test sets is important for SMT discriminative training.

Cao & Khudanpur (2012) use a separability measure to select SMT discriminative training samples. The idea is that candidates in a good development corpus should be easily separable by both BLEU scores ($B(D)$) and feature vector scores ($J(D)$). The approach in Cao & Khudanpur (2012) is defined as follows:

$$Q(D) = B(D) \times J(D) \tag{2.36}$$

$B(D)$ is a score separability metric that can be calculated as follows: we consider R_1 is the set of oracle candidates in the N-best-list, R_i and $i \neq 1$ is the set of other candidates in the N-best-list, N is the size of the N-best-list, and $M(.)$ is the document

level metric score of the set (for example, $M(R_1)$ is the document level metric score for the set of best candidates). The $B(D)$ can be defined as:

$$B(D) = \frac{(N-1)M(R_1)^2}{\sum_{i \neq 1} M(R_i)} \tag{2.37}$$

$J(D)$ is a class separability score that measures the separability between oracle candidates (class 1) and non-oracles (class 2). We consider m_i is the mean feature vector of class i, $m_{1,2}$ is the mean feature vector of all candidates and C_i is the set of candidates in the class i. The $J(D)$ is defined as follows:

$$J(D) = \frac{t_r(\sum_{i=1}^{2}(m_1 - m_{1,2})(m_i - m_{1,2})^T)}{t_r(\sum_{i=1}^{2} \frac{1}{|C_i|} \sum_{x \in C_i}(x - m_i)(x - m_i)^T)} \tag{2.38}$$

Cao & Khudanpur (2012)'s method does not require knowledge of the test sentences to be translated and therefore is suitable for use in off-line global feature weight training. However, this method requires decoding a large set of N-best lists to calculate the suitability of a training sentence and therefore it may not be applicable to large scale development data selection.

2.6 Summary

In this chapter we reviewed background and previous work on discriminative training, including algorithms, evaluation metrics and data selection methods. In summary, in order to obtain better discriminative training accuracy, four main aspects are relevant in discriminative training algorithms: 1) Training criteria that are closely related to SMT translation quality, for example, using MERT instead of maximum likelihood training; 2) Suitable oracle selection strategies, given that the reference or maximum BLEU candidates are not always a suitable target for parameter updates; 3) More accurate automatic evaluation metrics for positive/negative labelling of training instances ; 4) better training data, whereby data is more closely related to the segments that will be translated in the future, but also generally 'clean' and more useful. In the following chapters, we design and improve discriminative training algorithms based on existing training criteria, and propose solutions to improve discriminative training focusing on challenges 2-4.

3

Automatic Evaluation Metrics with Better Human Correlation

Automatic evaluation metrics are fundamentally important for machine translation, allowing comparison of systems performance, measurements of progress over time, and efficient training. As we previously discussed, metrics are one of the main components in discriminative training, which is used as gold scoring function to discriminate between good and bad translations. The BLEU metric has been used as the default gold scoring function in most state-of-art SMT systems (Koehn *et al.*, 2007; Li *et al.*, 2009), but other metrics such as METEOR (Banerjee & Lavie, 2005), TER (Snover *et al.*, 2006) and TESLA (Liu *et al.*, 2010) have also showed promising results for this purpose. Previous work has shown that the effectiveness of discriminative training heavily depends on the evaluation metric used (Callison-Burch *et al.*, 2011). Current evaluation metrics can be grouped into two classes: heuristic approaches, like BLEU, and those using supervised learning trained on human judgement data. Trained metrics normally provide better correlation with human judgements. In addition, they are flexible to the inclusion of various features. However, they are highly dependent on training data, which is often specific to each language pair. Heuristic approaches are less flexible but do not require training data, and therefore can be considered language-independent.

This chapter introduces two novel automatic evaluation metrics, ROSE and SIMPBLEU. ROSE (Section 3.1) is a trained metric that uses only simple features that are portable across languages and fast to compute. It is sentence level, as opposed to document level, which allows it to be used in a wider range of settings. SIMPBLEU

(Section 3.2) is a heuristic metric based on BLEU. It explores new variants of BLEU which address the limitations of the original metric, resulting in a more flexible metric that is not only more reliable, but also allows for more accurate discriminative training. Our experimental settings to test these metrics and their results are described in Section 3.3.

3.1 Regression and Ranking-based Evaluation

BLEU is still the most commonly used metric in automatic machine translation evaluation. However, several drawbacks have been identified for this metric (Chiang *et al.*, 2008a; Callison-Burch *et al.*, 2006; Banerjee & Lavie, 2005), most notably that it omits recall (substituting this with a penalty for overly short output) and that it is not easily applicable at sentence level. Metrics such as METEOR (Banerjee & Lavie, 2005) account for both precision and recall, but their relative weights are difficult to determine.

In contrast to heuristic metrics, trained metrics use supervised learning to learn directly from human judgements. This allows the combination of different features and can better fit specific tasks, such as evaluation focusing on fluency, adequacy, relative ranks or post-editing effort. Previous work includes approaches using classification (Corston-Oliver *et al.*, 2001), regression (Albercht & Hwa, 2008; Specia *et al.*, 2009; Specia & Gimenez, 2010), and ranking (Duh, 2008). This work achieved good results and better correlations with human judgements than purely heuristic metrics.

Automatic metrics must find a balance between several key issues: 1) applicability to texts of different sizes (documents and sentences); 2) ease of portability to different languages; 3) runtime requirements and 4) correlation with human judgements. Previous work has typically ignored at least one of these issues. For example, BLEU applies only to documents, while trained metrics tend to be specific to a given language pair and are slow to compute.

This section presents ROSE, a trained metric which is loosely based on BLEU, but seeks to further simplify its components so that it can be used for sentence-level evaluation. This contrasts with BLEU which is defined over documents, and must be coarsely approximated to allow sentence level application. The increased flexibility of ROSE allows the metric to be used in a wider range of situations, including during decoding. ROSE uses a linear model with a small number of simple features. The

model is trained using regression or ranking against data with human judgements. The benefits of using only simple features are that ROSE can be trivially ported between target languages, and that it can be run very quickly. Features include precision and recall over different sized n-grams, and the difference in word counts between the candidate and the reference sentences, which is further divided into counts of content words, function words and punctuation. An extended version also includes features over POS tag sequences but is less portable across languages than standard ROSE.

3.1.1 Model

ROSE is defined as a linear model and its weights are trained by a Support Vector Machine (SVM) (Joachims, 1999). It is formulated as

$$S = WF(e, r), \tag{3.1}$$

where W is the feature weights vector and $F(c, r)$ is a function which takes candidate translation (e) and reference (r), and returns the feature vector. S is the response variable, measuring the "goodness" of the candidate translation. A higher score means a better translation, although the magnitude is not always meaningful.

We propose two methods for training: a linear regression approach, ROSE-reg, trained to match a human evaluation score, and a ranking approach, ROSE-rank, trained to match the relative ordering of pairs of translations assigned by human judges. Unlike ROSE-reg, ROSE-rank only gives a relative score between sentences, indicating that translation A is better than translation B, or vice versa. The features used in ROSE will be listed in Section 3.1.2, and the regression and ranking methods are described in Section 3.1.3.

3.1.2 ROSE Features

The features used in ROSE are listed in Table 3.1. These include counts over string n-gram matches, word counts and counts for POS n-grams.

N-gram string matching features are used to measure how closely the candidate sentence resembles the reference in terms of the words used. Both precision and recall are considered. Word count features measure length differences between the candidate and the reference, which is further divided into function words, punctuation and content

words. POS features are defined over POS n-gram matches between the candidate and the reference and are aimed at measuring similarity in terms of sentence structure. We describe these features in detail in what follows.

String Matching Features

The string matching features include precision, recall and F1-measure over n-grams of various sizes. N-gram precision measures how many n-grams match between the sequences of words in the candidate sentence and the reference sentences:

$$p_n = \frac{Count_{clip}(\text{n-gram}_{matched})}{Count(\text{n-gram}_{candidate})} \tag{3.2}$$

where $Count_{clip}(\text{n-gram}_{matched})$ are the matched counts of n-grams (to the reference) in the candidate sentence and $Count(\text{n-gram}_{candidate})$ is the total number of n-gram occurrence in the candidate.

Recall is also taken into account in ROSE, so clipping was deemed unnecessary in the precision calculation: repeating words in the candidate will increase precision but at the expense of recall. Recall is calculated as

$$p_n = \frac{Count(\text{n-gram}_{matched})}{Count(\text{n-gram}_{reference})}, \tag{3.3}$$

where $Count(\text{n-gram}_{reference})$ is the total number of n-grams occurrence in the reference. If multiple references are available, the n-gram precision follows the same strategy as BLEU: the n-grams in the candidate can match any of the references. For recall, ROSE will match the n-grams in each reference separately, and then choose the reference with maximum recall.

Word Count Features

The word count (WC) features measure the length ratio between the candidate and reference sentences.

$$WC = \frac{\text{num. of words in candidate}}{\text{num. of words in reference}} \tag{3.4}$$

In a sentence, content words are more informative than function words (grammatical words) and punctuation. Therefore, the number of content words in the candidate

Description
n-gram precision, n=1...4
n-gram recall, n=1...4
n-gram F-measure, n=1...4
Average n-gram precision
Word count
Function word count
Punctuation count
Content word count
n-gram POS precision, n=1...4
n-gram POS recall, n=1...4
n-gram POS f-measure, n=1...4
n-gram POS string mixed precision, n=1...4

Table 3.1: ROSE Features. The dashed line separates the core features from the extended POS features (for ROSE-regpos and ROSE-rankpos)

is a important indicator in evaluation. Besides measuring the length difference for the entire candidate and reference sentences, we measure the ratio of *function words*, *punctuation* and *content words* between the candidate and the reference. We normalise this difference by the length of the reference, which allows comparability between short versus long sentences. If multiple references are available, we choose the ratio that is closest to 1.

Part-of-Speech Features

The string matching and word count features only measure similarities on the lexical-level, and not over sentence structure or synonyms. To add this capability we include Part-of-Speech (POS) tag features, which work in similar ways to the string matching features, but using POS tags instead of words. They measure precision, recall and F-measure over POS tag n-grams (n=1...4). In addition, we include features that mix tokens and POS tags.

The string/POS tags mixed features are used for handling synonyms. One limitation of string n-gram matching is that it is not able to deal with words in the candidate

Example
Reference: A/DT red/ADJ vehicle/NN
Candidate 1: A/DT red/ADJ car/NN
Candidate 2: A/DT red/ADJ rose/NN
Candidate 3: A/DT red/ADJ red/ADJ

Table 3.2: Example of the use of mixed features for evaluation. Using string matching only, the three candidates obtain the same score, but with mixed POS features we determine that Candidate 3 is worse than Candidates 1 and 2

which are synonymous of words in the reference. One approach for doing so is to use an external resource such as WordNet, like in METEOR (Banerjee & Lavie, 2005). However this would limit the portability of the metric. Instead, we use POS tags as a proxy. In most cases synonyms share the same POS. We can reward such potential synonyms by considering n-grams over a mixture of string and POS tags: either string or POS in candidate matching a reference's string or POS will be treated as a matching.

Example
Reference: A/DT red/ADJ vehicle/NN
Candidate 1: A/DT red/ADJ car/NN
Candidate 2: A/DT red/ADJ rose/NN
Candidate 3: A/DT red/ADJ red/ADJ

Table 3.3: Example of mixed string/POS matches. Green tokens indicate matches and red tokens indicate mismatches. Using mixed feature we can determine that Candidate 3 is worse than Candidates 1 and 2

For example, considering the example in Table 3.2, all three candidates match 2 unigrams and mismatch the last word in the reference. If only string n-gram matching is used (as illustrated in Table 3.3), all candidates will receive the same score (2/1/0 uni/bi/trigrams), which means these three candidates have same translation quality. However, Candidate 1 is a better translation, because *car* is a synonym of *vehicle*, and they share the same POS. In our mixed features we match either the POS tag or the

actual string, so that Candidates 1 and 2 will be scored 3/2/1 for uni/bi/trigrams and Candidate 3 remains scored as 2/1/0. In this case, Candidate 1 will be ranked higher than Candidate 3. Although the mixed feature cannot distinguish the difference between Candidates 1 and 2, they do not require WordNet, so they are more applicable across languages.

3.1.3 Training

The linear model combining the various features described above is trained on human evaluation data in two different ways: ranking and regression. In both cases we used the SVM-light tool (Joachims, 1999), which implements SVM Regression and SVM Ranking algorithms. In the ranking model, the training data is a set of candidate translations labelled with their relative rankings, as given by human annotators. For regression, we use a different dataset: human annotations of post-editing effort (this will be further described in Section 3.3). The SVM regression algorithm learns weights with minimum magnitude that limit prediction error to within an accepted range with a soft-margin formulation (Smola & Scholkopf, 2004).

3.2 BLEU Deconstructed

Trained metrics such as ROSE show better correlation with human judgements than BLEU (see, for example, the WMT 2011 results (Callison-Burch *et al.*, 2011)). However trained metrics require human labelled data for training and are less reliable when applied to languages and domains different from the training set. Therefore they have not been commonly adopted for discriminative training.

BLEU was designed for evaluating MT output against multiple references, and over large documents. However, evaluating translations at sentence level with a single reference is much more common in MT research. Popular evaluation campaigns such as those organised by the WMT workshop only provide one reference for test and development corpora. In addition, many state-of-the-art discriminative training algorithms require sentence-level evaluation metrics (Liang *et al.*, 2006; Chiang *et al.*, 2008b; Hopkins & May, 2011). Often this means using a sentence-based approximation of BLEU, which can unduly bias the system and affect overall performance. BLEU has been shown to perform less well when applied at the sentence or sub-sentence levels, and when using

only one reference (Chiang *et al.*, 2008a; Callison-Burch *et al.*, 2006; Banerjee & Lavie, 2005). One reason is that in this setting BLEU has many zero or low counts for higher order n-grams, and this has a disproportional effect on the overall score. As previously mentioned, another problem with BLEU is its brevity penalty, which has been shown to be a poor substitute for recall.

Previous research has sought to address these problems. Doddington (2002) suggests using arithmetic mean instead of geometric mean. Zhang *et al.* (2004) shows that unigram and bigram precision contribute over 95% of overall precision, and state that adding higher order n-gram precision introduces a bias towards fluency over precision. This led us to question the effect of removing or substituting some components in BLEU, especially for sentence-level evaluation. In this section, we provide experimental analysis of each component in BLEU aiming to design better evaluation metrics for sentence-level MT evaluation and MT system tuning with a single reference.

3.2.1 Limitations of the BLEU Metric

We reviewed the BLEU metric in detail in Chapter 2. In this section we will discuss the limitations of this metric. First, in a short document or sentence, there is a high probability of obtaining zero trigram or 4-gram precision, which makes the overall BLEU score equal zero due to the use of geometric mean. Similarly, very low (but non-zero) counts disproportionately affect the final score. A common method to alleviate this effect is smoothing the counts (Lin & Och, 2004; Owczarzak *et al.*, 2006; Koehn *et al.*, 2008; Hanneman *et al.*, 2008), e.g. adding α both to the numerator and denominator of Equation 2.28. This avoids zero precision scores and zero overall BLEU score. However, different α values will affect the accuracy of the approximation, and it is unclear what a reasonable value to use is.

BLEU supports multiple references, which makes it hard for it to obtain an estimate of recall. Therefore, recall is replaced by the brevity penalty (BP). Banerjee & Lavie (2005) state that the BP is a poor substitute for recall. Banerjee & Lavie (2005); Liu *et al.* (2010); Song & Cohn (2011) include recall in their metrics and achieve better correlation with human judgements compared to BLEU.

Lin & Och (2004) analysed BLEU at the sentence level using Pearson's correlation with human judgements over 1 to 9 grams. In order to apply BLEU for sentence level, they add one to the count of each n-gram. Results show that BLEU with only unigram

precision has the highest correlation with adequacy (0.87), while adding higher order n-gram precision factors decreases the adequacy correlation and increases fluency. Overall they recommend using up to 5-gram precision to achieve the best balance. Zhang *et al.* (2004)'s experiments show that unigram and bigram precision contribute over 95% of the overall precision. They also found that adding higher n-gram precision leads to a bias towards fluency over adequacy. However, it is not clear whether fluency or adequacy is more important, with recent evaluation favouring ranking judgements that implicitly consider both fluency and adequacy (Bojar *et al.*, 2014; Macháček & Bojar, 2013; Callison-Burch *et al.*, 2012, 2011, 2010, 2009).

These limitations affect the possible applications of BLEU, particularly for SMT discriminative training. In discriminative training, the references are given, and we want the decoder to produce translations with high BLEU score. Current solutions rank translations in N-best lists (Liang *et al.*, 2006; Och, 2003) or explicitly search for the maximum BLEU translation and use this for discriminative updates (Arun & Koehn, 2007; Liang *et al.*, 2006; Tillmann & Zhang, 2006; Chiang *et al.*, 2008b). In order to efficiently search for the maximum BLEU translation we need to be able to evaluate BLEU over partial sentences. However, clipping and high order n-grams make it unfeasible to apply BLEU during decoding. Thus the process relies on coarse approximations.

3.2.2 Simplified BLEU

In an attempt to better understand and simplify BLEU so that it meets our requirements, we analyse each component of BLEU and seek a solution to improve the above mentioned shortcomings, especially for sentence-level evaluation. We test the effect of the brevity penalty as well as a recall-based BLEU. Further, we test how each component contributes to BLEU. We will use following notation for each component:

- P: Precision
- R: Recall
- A: Arithmetic mean
- G: Geometric mean

- B: Brevity penalty
- 1, 2, 3, 4 : 1 to 4-grams
- C: Clipping

Note that our short-hand for standard BLEU is PGBC4, while a metric for clipped recall over unigrams and bigrams with no brevity penalty is labelled RGC2.

3.3 Experiments with ROSE and SIMPBLEU

This section introduces the settings for the experiments with the ROSE and SIMPBLEU as standalone metrics. We will compare the evaluation results of several ROSE and SIMPBLEU variants (these variants are listed in Table 3.4) with standard BLEU against human judgements. The baseline BLEU version is David Chiang's implementation[1].

ROSE variants
Regression-based ROSE without mixed POS features (ROSE-reg)
Regression-based ROSE with mixed POS features (ROSE-regpos)
Ranking-based ROSE without mixed POS features (ROSE-rank)
Ranking-based ROSE with mixed POS features (ROSE-rankpos)
SimpBLEU variants
Standard BLEU (PGBC4)
BLEU with arithmetic avg (PABC4)
BLEU with recall (RGBC4)
BLEU with recall and arithmetic avg (RABC4)
BLEU with 1(2,3,4) grams (PGBC1(2,3,4))
BLEU with 1(2,3,4) grams without clipping (PGB1(2,3,4))
BLEU without BP (PGC4) BLEU with recall without BP (RGC4)

Table 3.4: ROSE and BLEU variants

The training data used for ROSE are sentences judged by humans in WMT10 (Callison-Burch *et al.*, 2010). A regression model was trained based on sentences with

[1]https://github.com/tylin/coco-caption/blob/master/pycocoevalcap/bleu/bleu_scorer.py

human annotation for post-editing effort. The three levels used in WMT10 are "OK", "EDIT" and "BAD", which we treat as response values 3, 2 and 1. In total, 2,885 sentences were used in the regression model training. The ranking model was trained based on sentences with human annotation as ranking, with tied results allowed at training time. 1,675 groups of sentences were used for training, where each group contains five sentences manually ranked from 1 (best) to 5 (worst). In order to test ROSE's ability to adapt to a new language without training data, ROSE was only trained with English data.

In order to test the portability of the metrics, our experiments are based on four languages: English (en), French (fr), Spanish (es) and German (de). A function word list was created for each language and used in ROSE for feature extraction. Each function word list contains the 100 most common function words in the language. English POS tags were generated using NLTK (Bird & Loper, 2004), and POS features were only used for into English translation evaluation.

The experiments include two parts: document-level evaluation and sentence-level evaluation. The detailed settings for each part are described in the following two Sections.

3.3.1 Document-level Evaluation

For document-level, we follow WMT08's (Callison-Burch *et al.*, 2008) evaluation procedure, whereby we first rank each document by human using following evaluation methods:

- **Ranking:** Humans judge the candidate sentences by ranking them in order of quality. To apply this to whole documents, documents are ranked according to the proportion of the candidate sentences in a document that are better than all of the other candidates.

- **Constituent Ranking:** The constituent task also involves human ranking judgements, but operates over chosen syntactic constituents, instead of entire sentences.

- **Yes/No:** In this task human judges decide whether or not a particular part of a sentence is acceptable. This is applied to document-level evaluation by calculating the proportion of YES sentences in the document.

Then we compare each BLEU variant evaluation results against human rankings using Spearman's ρ correlation:

$$\rho = 1 - \frac{6 \sum_{i=1}^{n} d_i^2}{n(n^2 - 1)}, \tag{3.5}$$

where d_i measures the difference between the rank value assigned to sentence i by the system versus the human, and the n is number of sentences in the document. ρ varies between 1 and -1, where higher scores indicate that the automatic evaluation metric has higher correlation with human judges. For example, $\rho = 1$ means that the automatic evaluation metric ranks the documents in the same order as human judges do, and $\rho = -1$ means the ranking is opposite to that of human judges.

Our test corpora are from all systems submitted in WMT08 for the "test2008" test set, with each document including more than 500 sentences. We selected Spanish, French and German into and out-of English. The final score is the average of the BLEU variant Spearman's ρ correlation with the human rankings in three tasks of 'Ranking', 'Constituent' and 'Yes/No':

SVM kernel	**es-en**	**fr-en**	**de-en**	**avg**
Linear	0.76	0.93	0.58	0.75
Polynomial	0.76	0.92	**0.59**	**0.76**
RBF	**0.77**	**0.95**	0.54	0.75

Table 3.5: Document-level evaluation of ROSE-reg with different SVM kernel functions. The results were computed as the average Spearman's ρ correlation for the yes/no, ranking and constituent tasks. Boldface numbers indicate the best correlation in each language

In the document-level experiment, we first compute ROSE-reg with three SVM kernel functions. Results are shown in Table 3.5. The performance is similar with all kernel functions. However, the linear kernel results in faster training and prediction times, making the metric more applicable in decoding. Therefore, a linear kernel function was used in ROSE for the follow up experiments.

Table 3.6 shows the document-level evaluation performance of ROSE (without POS features) and SIMPBLEU variants. The best performing metric is SIMPBLEU with arithmetic average (PABC4). PGB4 (BLEU without clipping) performs the same as BLEU, and thus we can say that the effect of clipping on BLEU is not noticeable here.

	BLEU	ROSE-reg	ROSE-rank	RGBC4	PABC4	PGB4
es-en	0.80	0.76	0.76	**0.81**	0.80	0.80
fr-en	**0.95**	0.93	**0.95**	0.93	0.94	**0.95**
de-en	0.59	0.58	0.57	0.58	**0.68**	0.59
en-es	0.78	0.73	0.74	0.75	**0.81**	0.78
en-fr	**0.94**	**0.94**	0.89	**0.94**	**0.94**	**0.94**
en-de	0.72	**0.80**	0.78	0.72	0.77	0.72
avg.	0.80	0.79	0.78	0.79	**0.82**	0.80

Table 3.6: Document-level evaluation results (Spearman's ρ correlation). The results were computed as the average Spearman's ρ correlation for the yes/no, ranking and constituent tasks

However, in some of the following experiments (Table 3.9) we found that clipping affects lower n-gram SIMPBLEU variants.

Rank-task	PGBC4(BLEU)	ROSE-reg	ROSE-rank
es-en	0.66	0.57	**0.85**
fr-en	**0.97**	**0.97**	0.96
de-en	0.69	0.69	**0.76**
avg.(into en)	0.77	0.74	**0.86**
en-es	**0.85**	0.75	0.69
en-fr	**0.98**	**0.98**	0.93
en-de	0.88	0.93	**0.94**
avg.(from en)	**0.90**	0.89	0.88

Table 3.7: Document-level evaluation results (Spearman's ρ correlation) of ranking task only

The overall performance of ROSE variants and recall-based SIMPBLEU are slightly better than BLEU at document level, but note that ROSE-rank has much better ρ correlation than BLEU for the into English ranking task. These results are shown in Table 3.7, which lists the ROSE evaluation results for the ranking task only. From the table, ROSE-rank wins over BLEU with almost 0.1 ρ correlation gain (0.86 vs 0.77) for into English translation evaluation. However, BLEU is still slightly better than ROSE-rank for out-of English translation evaluation (0.90 vs 0.88). This may be because

ROSE-rank is trained only on into English human ranking data.

All Task	ROSE-reg	ROSE-regpos	ROSE-rank	ROSE-rankpos
es-en	0.76	0.78	0.76	0.76
fr-en	0.93	0.94	0.95	0.95
de-en	0.58	0.59	0.57	0.52
avg.	0.76	0.77	0.76	0.74

Table 3.8: Document-level evaluation results (Spearman's ρ correlation) of ROSE with POS features for into English translation evaluation

Table 3.8 shows the evaluation performance of ROSE with POS features. According to the table, the performance of the regression-based ROSE is improved across all three language pairs by adding the POS features. For the ranking-based model, adding POS features leads to no variation for Spanish into English and French into English evaluation, and to worse results for German into English. The reason behind this is not clear, but these results show that the regression model is more reliable than the ranking model with POS feature sets.

Table 3.9 shows the evaluation performance of BLEU variants with different n-gram orders with and without clipping. According to the table, with clipping the BLEU variants using tri- and four-grams have better correlation with humans on the English evaluation, but for other languages, evaluation using only unigram and bigram leads to better performance. In order to obtain the best evaluation performance we recommend using different n-gram orders in different languages. For English evaluation the best BLEU performance should include all four grams, but for French, Spanish and German we suggest only using unigrams in BLEU. For other languages a safe option is to use up to trigrams, as the overall best performance in our test was obtained with the BLEU variant using up to trigram precision.

If we remove clipping in BLEU, except for English into Spanish translation, the performance drops when reducing the order of n-grams. This may be because although over generating words increases the lower order n-gram precision, it can degrade the higher order n-gram precision. Clipping is thus more important for lower order n-grams. In addition, the test data is produced by BLEU-tuned systems, so sentences with over

generated words will most likely already have been penalised. Therefore, we do not suggest removing clipping from BLEU.

	BLEU Variants with clipping			
	PGBC4(BLEU)	**PGBC3**	**PGBC2**	**PGBC1**
es-en	**0.79**	**0.79**	0.79	0.77
fr-en	**0.95**	**0.95**	0.94	0.93
de-en	**0.59**	**0.59**	**0.59**	0.55
en-es	0.74	0.80	0.80	**0.81**
en-fr	**0.93**	**0.93**	**0.93**	**0.93**
en-de	0.71	0.71	**0.72**	**0.72**
avg.	0.79	**0.80**	0.79	0.79
	BLEU Variants without clipping			
	PGB4	**PGB3**	**PGB2**	**PGB1**
es-en	**0.79**	**0.79**	0.79	0.76
fr-en	**0.95**	**0.95**	0.94	0.92
de-en	**0.59**	**0.59**	0.59	0.56
en-es	0.77	0.80	0.80	**0.81**
en-fr	**0.93**	**0.93**	0.93	0.88
en-de	**0.71**	0.70	0.70	0.70
avg.	0.79	**0.79**	0.79	0.78

Table 3.9: SIMPBLEU's document-level evaluation results (Spearman's ρ correlation) testing 1-4 grams and clipping

3.3.2 Sentence-level Evaluation

For sentence-level evaluation we follow the procedure from WMT09 (Callison-Burch *et al.*, 2009), which uses Kendall's τ correlation (Equation 3.6) to measure metrics' quality:

$$\tau = \frac{\text{number of concordant pairs - number of discordant pairs}}{\text{total pairs}}, \tag{3.6}$$

where ranked lists of translations according to humans judgements and metric's score are compared by counting the number of concordant and discordant relative ordering of pairs of translations, ignoring pairs with ties in either human or metric rankings.

We use Kendall's τ correlation to compare the sentence rankings produced by BLEU and all of our variants against human rankings. The human rankings were collected from WMT09, from all languages into English evaluation task data. After removing tied rankings, the test set contains $1,702$ pairs of ranked sentences.

We also test sentence-level evaluation for document evaluation performance: we evaluate each sentence in the document and take the average sentence scores for a document-level score. An add α smoothing is used for sentence-level BLEU, where the default α value in this experiment is 1. In subsequent experiments we also test different α values for smoothing and their effect on sentence-level BLEU performance.

Metric	**Kendall's tau**
BLEU-smoothed	0.17
ROSE-reg	0.12
ROSE-regpos	0.16
ROSE-rank	**0.20**
ROSE-rankpos	0.17

Table 3.10: Sentence-level evaluation (Kendall's τ correlation) of ROSE. Boldface numbers indicate the best Kendall's τ correlation

Table 3.10 shows the sentence-level evaluation results of ROSE variants. According to the table, ROSE-rank has the best overall score in all versions of ROSE and BLEU. This result corroborates the document-level evaluation results: with appropriate training data, ranking-based ROSE is able to reach better human correlation than BLEU in ranking tasks. The results also show that adding the POS tag features helps the regression model (ROSE-regpos), but it degrades the performance of the ranking model (ROSE-rankpos).

Table 3.11 shows the SIMPBLEU sentence-level evaluation results. According to the table, SIMPBLEU variants with arithmetic average (i.e., Precision with ABC4, which we refer to as PABC4) has the best performance, increasing BLEU's Kendall's τ from 0.1774 to 0.2103. Compared to ROSE, PABC4 shows even better performance than ROSE-rank (0.2103 vs 0.206). We also found that PGBC4's (BLEU precision with GBC4) performance slightly decreases without clipping at sentence level. Without the brevity penalty, BLEU's performance drops both in the precision and recall variants,

	Precision	Recall
GBC4	0.17	0.15
GB4	0.17	0.15
ABC4	**0.21**	0.16
AC4	0.19	0.11
BC4	0.16	0.11
ABC4(no-smooth)	0.19	0.15

Table 3.11: Sentence-level Kendall's τ correlation of SIMPBLEU. Except for ABC4 (no smoothing), add one smoothing is used with all metrics

with a larger drop for the recall variant. This shows that the BP is not only a recall replacement in BLEU, it is also an important feature for evaluation. Non-smoothed arithmetic BLEU variants are comparable (only slightly worse) to the smoothed variant.

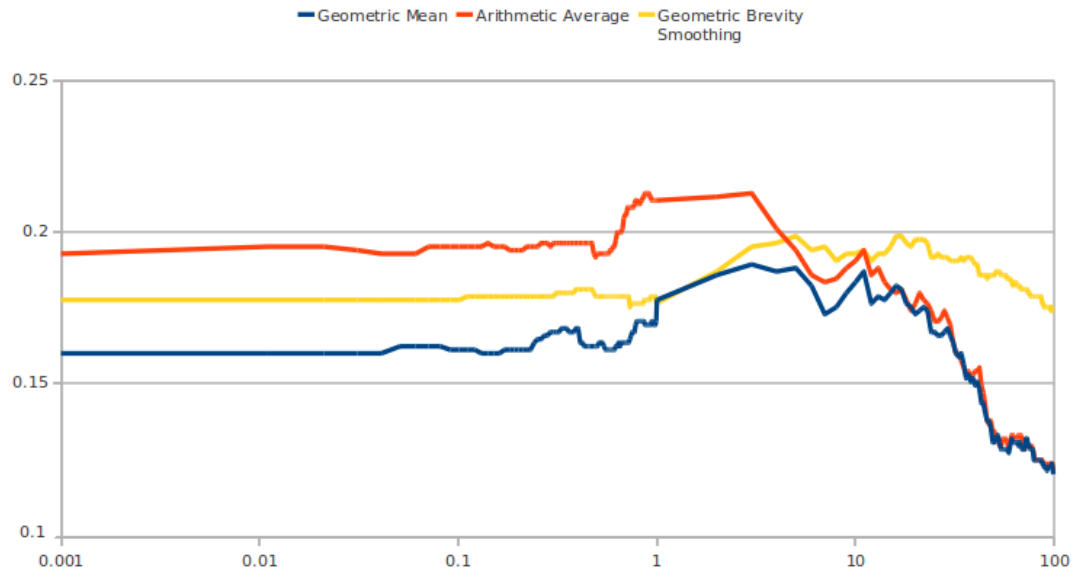


Figure 3.1: Smoothed BLEU Kendall's τ with smoothing values from 0.001 to 100

A question that naturally follows is how important smoothing of counts is to sentence-level evaluation. Figure 3.1 shows the results of smoothing α values from 0.001 to 100. In our experiments, the average sentence length is 23 words. When α is less than 1, smoothing has no effect on the performance of any BLEU variant. The

best smoothing value was found to be 3. The performance tends to drop beyond this value. Results also show that arithmetic average has better correlation than geometric mean with small smoothing values, and tends to reach similar performances with large smoothing α values. Smoothing with brevity penalty (Equation 3.7) leads to better performance than without brevity penalty smoothing, and the results are more reliable as the smoothing value increases.

$$BP_{\text{smooth}} = e^{(1-\frac{r+\alpha}{c+\alpha})} \tag{3.7}$$

The reason we recommend smoothed BP is that adding a smoothing value is equivalent to adding to the sentence length of the test and reference sentences, which affect the length ratio. We added the same smoothing value to the BP component to neutralise this effect. Smoothing the BP component is sensible when considering sentence-level application, as the effect of a single sentence on document-level BP is very small.

Table 3.12 compares the performance of BLEU variants by using different n-gram sizes for sentence-level evaluation. At sentence-level, geometric mean variants of BLEU have the best performance when using unigrams and bigrams, even without clipping, but for the arithmetic average variant, adding trigrams improves performance.

grams	**PGBC**	**PGB**	**PABC**
1-4 grams	0.17	0.17	**0.21**
1-3 grams	0.19	0.18	**0.21**
1-2 grams	**0.21**	**0.19**	0.20
1 grams	0.18	0.16	0.18

Table 3.12: Sentence-level SIMPBLEU evaluation (Kendall's τ correlation) in 1 - 4 grams

The results in Table 3.13 are the Spearman's correlation for document-level evaluation by using sentence-level BLEU variants. According to these results, the overall accuracy at sentence-level is lower than that of BLEU variants at document-level. Arithmetic average variants still outperform geometric mean variants.

	PGBC4(BLEU)	RGBC4	PABC4	PGB4
es-en	0.79	0.79	0.79	0.79
fr-en	**0.93**	0.92	0.92	0.93
de-en	0.55	**0.59**	0.58	0.55
en-es	0.76	0.75	0.76	0.76
en-fr	0.93	0.93	0.93	0.93
en-de	0.66	**0.80**	0.70	0.66
avg.	0.77	**0.78**	0.78	0.77

Table 3.13: Sentence-level evaluation for document ranking (Spearman's ρ correlation)

3.4 SIMPBLEU for Discriminative Training

Up until now we have applied our SIMPBLEU as standalone metrics to human evaluation data, testing whether our variant metrics result in better ranking of MT outputs. However, it remains to be tested whether the metrics can also work effectively as a loss function for tuning a translation system. This can be seen as an extrinsic way of testing the metric, which will encounter a much wider variety of outputs than those present in MT evaluation data. For instance, empty sentences, overly long output, etc.

In this experiment we investigate parameter tuning of the following SMT system: a Moses phrase-based approach (Koehn *et al.*, 2007) which we tune using cmert-0.5, David Chiang's implementation of MERT. We use the following (default) features:

- reordering model
- language models
- translation models, including forward and backward lexical probabilities, word count and phrase count
- word penalty.

The training data to build models for this experiment was the Europarl-v6 German to English corpus. For tuning, the dev-newstest2010 German to English development set from WMT10 (Callison-Burch *et al.*, 2010) was used. For test, the test set from WMT11 (Callison-Burch *et al.*, 2011) was used. For manual evaluation, we randomly

picked 50 unique output sentences from five versions of the system (where differences stem from the stochastic tuning process) for human ranking from best to worst.

The human ranking was done using Amazon Mechanical Turk and MAISE (Zaidan, 2011). For each ranking, source and reference sentences were provided, plus five candidate translations placed in random order. Each unit was ranked five times, by five different annotators. Annotation agreement was then measured as the average Cohen's Kappa coefficient (Cohen, 1960), which is a normalised agreement measure (Equation 3.8):

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)}, \tag{3.8}$$

where P(A) is percentage of times annotators agree with each other, and P(E) is the probability of agreement by chance. In our experiment $P(E) = \frac{1}{3}$ (candidate A can be ranked better than B, equal to B, or worse than B, so the probability of agreement by chance is $\frac{1}{3}$). We also calculate intra-agreement for each annotator, such that annotators with low intra-annotator agreement can be filtered out from our experiments.

	PABC4	PGBC4	PGBC2	PGB4	RGBC4
PABC4	–	0.27	0.26	0.25	0.29
PGBC4	**0.31**	–	0.29	0.28	0.28
PGBC2	**0.33**	**0.29**	–	0.21	0.26
PGB4	**0.28**	**0.29**	**0.23**	–	0.24
RGBC4	**0.33**	**0.32**	**0.29**	**0.28**	–

Table 3.14: German-to-English head-to-head: figures represent how often metric in column header beat metric in row. E.g. PABC4 ranked better than PGBC4 31% of the times, while PGBC4 ranked better than PABC4 only 27% of the times, so they tied 42% of the times. In this case: $P(A) = 0.608$ and $K = 0.396$

We had 42 annotators from the Amazon Mechanical Turk platform, producing a total of 250 rankings. Only 143 rankings were kept after filtering out six annotators with low intra annotator agreement. According to the filtered results (Table 3.14), arithmetic average BLEU ranks 31% better than geometric mean BLEU. PGBC2 and PGBC4 have the same performance. BLEU with clipping is slightly better than the version without clipping (0.29 vs 0.28).

According to results we showed above, BLEU can be improved for sentence-level evaluation by substituting geometric mean with arithmetic average, precision with recall and adjusting the order of n-grams. We now test the significance of the three BLEU variants by using the paired bootstrap re-sampling method introduced in (Koehn, 2004b). We randomly draw 50 groups (each group contains five ranked sentences) from the WMT09 test set to produce a test corpus, and test Kendall's τ of BLEU variants in this test corpus. We repeat this process 1000 times. Significance values are obtained as the percentage of BLEU variants that are better than standard BLEU.

Three pairs of significance test results are listed in Table 3.15. All three BLEU variants are better than standard BLEU in more than 80% of the time; the two arithmetic variants are better than standard BLEU in more than 90% of the time.

Metric	**Significance (%)**
PABC4	92.6
PABC3	90.9
PGBC2	84.0

Table 3.15: Paired sentence-level significance tests against standard smoothed BLEU

We also perform a binomial test for variants PABC4 & PGBC4. We obtain p-value equal 0.5811, i.e., there is no significant difference between arithmetic average and standard BLEU.

3.5 SIMPBLEU in WMT Evaluation

The Workshop on Statistical Machine Translation (WMT) provides a platform for evaluating and comparing evaluation metrics. In 2012 we submitted the SIMPBLEU PABC3 variant to the WMT competition. The WMT12 official results are listed in Tables 3.16-3.19. In all tasks (that these tables cover these various settings), our PABC3 shows better correlation with human scores than BLEU. In addition, PABC3 also proved very competitive against other metrics and ranked best for document-level out-of English evaluation test sets.

In WMT13 we submitted a precision (PABC3) and a recall (RABC2) SIMPBLEU variants to the workshop (Table 3.20-3.23). Our recall variant outperformed the preci-

	cs-en	de-en	es-en	fr-en	avg
SEMPOS	**0.94**	0.92	0.94	0.80	**0.90**
AMBER	0.83	0.79	**0.97**	0.85	0.86
TERRORCAT	0.71	0.76	**0.97**	**0.88**	0.83
PABC3	0.89	0.70	0.89	0.82	0.82
TER	0.89	0.62	0.92	0.82	0.81
BLEU	0.89	0.67	0.87	0.81	0.81
POSF	0.66	0.66	0.87	0.83	0.75
BLOCKERRCATS	0.64	0.75	0.88	0.74	0.75
WORDBLOCKEC	0.66	0.67	0.85	0.77	0.74
XENERRCATS	0.66	0.64	0.87	0.77	0.74
SAGAN-STS	0.66	n/a	0.91	n/a	n/a

Table 3.16: WMT12 document-level Spearman's ρ correlation between automatic evaluation metrics and human judgements for translations into English

	en-cs	en-de	en-es	en-fr	avg
PABC3	**0.83**	0.46	0.42	**0.94**	**0.66**
BLOCKERRCATS	0.65	0.53	0.47	0.93	**0.64**
ENXERRCATS	0.74	0.38	0.47	0.93	0.63
POSF	0.80	0.54	0.37	0.69	0.60
WORDBLOCKEC	0.71	0.37	0.47	0.81	0.59
TERRORCAT	0.65	0.48	0.58	0.53	0.56
AMBER	0.71	0.25	0.50	0.75	0.55
TER	0.69	0.41	0.45	0.66	0.55
METEOR	0.73	0.18	0.45	0.82	0.54
BLEU	0.80	0.22	0.40	0.71	0.53
SEMPOS	0.52	n/a	n/a	n/a	n/a

Table 3.17: WMT12 document-level Spearman's ρ correlation between automatic evaluation metrics and human judgements for translations out-of English

sion variant, and achieved very promising results in the competition: RABC2 SIMP-BLEU ranked the best evaluation metric at document level for out-of English evaluation, and the best evaluation metric at sentence level for both into and out-of English evaluations.

	en-cs	en-de	en-es	en-fr	avg
SPEDE07-PP	**0.26**	**0.28**	**0.26**	**0.21**	**0.25**
METEOR	0.25	0.27	0.25	**0.21**	0.24
AMBER	0.24	0.25	0.23	0.19	0.23
TERRORCAT	0.18	0.19	0.18	0.19	0.19
PABC3	0.19	0.17	0.19	0.13	0.17
XENERRCATS	0.17	0.18	0.18	0.13	0.17
POSF	0.16	0.18	0.15	0.12	0.15
WORDBLOCKEC	0.15	0.16	0.17	0.13	0.15
BLOCKERRCATS	0.07	0.08	0.08	0.06	0.07
SAGAN-STS	n/a	n/a	0.21	0.20	n/a

Table 3.18: WMT12 sentence-level Kendall's τ correlation between automatic evaluation metrics and human judgements for translations into English

	en-cs	en-de	en-es	en-fr	avg
METEOR	**0.26**	0.18	0.21	0.16	**0.20**
AMBER	0.23	0.17	**0.22**	0.15	0.19
TERRORCAT	0.18	**0.19**	0.18	**0.18**	0.18
PABC3	0.20	0.13	0.18	0.10	0.15
ENXERRCATS	0.20	0.11	0.17	0.09	0.14
POSF	0.15	0.13	0.15	0.13	0.14
WORDBLOCKEC	0.19	0.10	0.17	0.10	0.14
BLOCKERRCATS	0.13	0.04	0.12	0.01	0.08

Table 3.19: WMT12 sentence level Kendall's τ correlation between automatic evaluation metrics and human judgements for translations out-of English

For WMT14, we did not submit our metrics to the official competition, but we tested them after the evaluation campaign. For WMT14, the organisers changed the document ranking method by introducing the Trueskill algorithm (Sakaguchi *et al.*, 2014). In previous years, document were ranked based on the percentage of sentences in the document rated better than in other documents. However, Sakaguchi *et al.* (2014) argued that this approach does not consider the effects of document groupings. A 'lucky' document may end up rated highly only because it is always compared with

	fr-en	de-en	es-en	cs-en	ru-en	avg
METEOR	0.984	0.961	**0.979**	0.964	0.789	**0.935**
DEPREF-ALIGN	**0.995**	**0.966**	0.965	0.964	0.768	0.931
UMEANT	0.989	0.946	0.958	**0.973**	0.775	0.928
MEANT	0.973	0.926	0.944	**0.973**	0.765	0.916
SEMPOS	0.938	0.919	0.930	0.955	**0.823**	0.913
DEPREF-EXACT	0.984	0.961	0.937	0.936	0.744	0.912
RABC2	0.978	0.936	0.923	0.909	0.789	0.909
BLEU-MTEVAL-INTL	0.989	0.902	0.985	0.936	0.695	0.883
PABC3	0.989	0.846	0.832	0.918	0.704	0.858
BLEU-MTEVAL	0.989	0.895	0.888	0.936	0.670	0.876
BLEU-MOSES	0.993	0.902	0.879	0.936	0.651	0.872
CDER-MOSES	**0.995**	0.877	0.888	0.927	0.659	0.869
NLEPOR	0.945	0.949	0.825	0.845	0.705	0.845
LEPOR	0.945	0.934	0.748	0.800	0.779	0.841
NIST-MTEVAL	0.951	0.875	0.769	0.891	0.649	0.827
NIST-MTEVAL-INTL	0.951	0.875	0.762	0.882	0.658	0.826
TER-MOSES	0.951	0.833	0.825	0.800	0.581	0.798
WER-MOSES	0.951	0.672	0.797	0.755	0.591	0.753
PER-MOSES	0.852	0.858	0.357	0.697	0.677	0.688

Table 3.20: WMT13 document-level Spearman's ρ correlation between automatic evaluation metrics and human judgements for translations into English

poorer document. The Trueskill algorithm is inspired by the Xbox Live online gaming ranking system (Herbrich *et al.*, 2007), where each document is considered a player. Every player (document) has the same skill level and uncertainty at the beginning, which follows a Gaussian distribution. The Gaussian mean represents the player's current skill, while its variance represents the uncertainty. Trueskill first selects the player with highest uncertainty (high variance) and picks its opponent with similar skill (mean) to compete (sentence-level comparison). After each round of competition, Trueskill updates the skill estimates and uncertainty according to their current skill level and uncertainty. Large skills difference and low variance will result in the system making larger skill updates. Low skill difference and high variance will conversely results in smaller skill updates. Therefore, a system will compete against other systems

	en-fr	en-de	en-es	en-cs	en-ru	avg
RABC2	0.924	**0.925**	0.830	0.867	0.710	**0.851**
LEPOR	0.904	0.900	0.841	0.748	0.855	0.850
NIST-MTEVAL-INTL	**0.929**	0.846	0.797	0.902	0.771	0.849
CDER-MOSES	0.921	0.867	**0.857**	0.888	0.701	0.847
NLEPOR	0.919	0.904	0.852	0.818	0.727	0.844
NIST-MTEVAL	0.914	0.825	0.780	0.916	0.723	0.832
PABC3	0.909	0.879	0.780	0.881	0.697	0.829
METEOR	0.924	0.879	0.780	**0.937**	0.569	0.818
BLEU-MTEVAL-INTL	0.917	0.832	0.764	0.895	0.657	0.813
BLEU-MTEVAL	0.895	0.786	0.764	0.895	0.631	0.794
TER-MOSES	0.912	0.854	0.753	0.860	0.538	0.783
BLEU-MOSES	0.879	0.786	0.759	0.895	0.574	0.782
WER-MOSES	0.914	0.825	0.714	0.860	0.552	0.773
PER-MOSES	0.873	0.686	0.775	0.797	0.591	0.744

Table 3.21: WMT13 document-level Spearman's ρ correlation between automatic evaluation metrics and human judgements for translations out-of English

	fr-en	de-en	es-en	cs-en	ru-en	avg
RABC2	**0.303**	**0.318**	**0.388**	0.260	0.234	**0.301**
METEOR	0.264	0.293	0.324	**0.265**	**0.239**	0.277
Pearson correlationDEPREF-ALIGN	0.257	0.267	0.312	0.228	0.200	0.253
DEPREF-EXACT	0.258	0.263	0.307	0.227	0.195	0.250
PABC3	0.238	0.236	0.287	0.208	0.174	0.229
NLEPOR	0.225	0.240	0.281	0.176	0.172	0.219
SENTBLEU-MOSES	0.229	0.218	0.266	0.197	0.170	0.216
LEPOR	0.235	0.221	0.236	0.187	0.177	0.211
UMEANT	0.161	0.166	0.202	0.160	0.108	0.160
MEANT	0.158	0.160	0.202	0.164	0.109	0.159

Table 3.22: WMT13 sentence-level Kendall's τ correlation between automatic evaluation metrics and human judgements for translations into English

that have similar performance.

Another change in the document-level ranking from WMT14 was that human cor-

	en-fr	en-de	en-es	en-cs	en-ru	avg
RABC2	**0.261**	**0.254**	**0.231**	**0.192**	**0.245**	**0.236**
METEOR	0.236	0.203	0.175	0.160	0.203	0.195
PABC3	0.219	0.197	0.187	0.148	0.175	0.185
NLEPOR	0.200	0.199	0.163	0.139	0.188	0.178
SENTBLEU-MOSES	0.214	0.177	0.171	0.139	0.173	0.175
LEPOR	0.206	0.179	0.178	0.084	0.205	0.170

Table 3.23: WMT13 sentence-level Kendall's τ correlation between automatic evaluation metrics and human judgements for translations out-of English

relation was calculated using Pearson's correlation coefficient rather than Spearman's ρ correlation. The organisers argue that Spearman's ρ disregards the absolute differences in the scores and that his may be unfair to some metrics. In contrast, Pearson's correlation is able to take into account the score difference between human and metric judgements. The correlation metric in WMT14 is calculated by Equation 3.9:

$$\kappa = \frac{\sum_1^n (Human_i - \overline{Human})(Metric_i - \overline{Metric})}{\sqrt{\sum_1^n (Human_i - \overline{Human})^2}\sqrt{\sum_1^n (Metric_i - \overline{Metric})^2}}, \tag{3.9}$$

where $Human$ is the human score vector and $Metric$ is the metric score vector, $\overline{Human}$ and $\overline{Metric}$ are their means, respectively.

Here we present the WMT14 document-level evaluation results using Trueskill ranked documents. The evaluation method is same as in Section 3.3.1, where we compare our SIMPBLEU variants against other metrics submitted to WMT14. Table 3.24 and Table 3.25 show the SIMPBLEU variants Pearson's correlations in comparison with other metrics submitted to WMT14. Our RABC2 still shows strong correlation with human judges, and is ranked the best metric in the out-of English translation tasks.

3.6 Summary

In this chapter we focused on Problem 3 discussed in Chapter 1 – the design of better automatic evaluation metrics for discriminative training. We introduced the ROSE and SIMPBLEU metrics, both of which achieve better human correlation than BLEU. We

	en-fr	en-hi	en-cs	en-ru	en-de	avg
RABC2	0.943	0.985	0.974	0.927	0.335	**0.833**
APAC	0.950	0.940	0.973	0.926	0.346	0.827
PABC3	0.937	0.981	0.967	0.911	0.318	0.823
CDER	0.949	0.949	0.982	0.938	0.278	0.819
METEOR	0.941	0.975	0.976	0.923	0.263	0.816
AMBER	0.928	**0.990**	0.972	0.926	0.241	0.811
NIST	0.941	0.981	0.985	0.927	0.200	0.807
ELEXR	0.885	0.962	0.979	0.938	0.260	0.805
TBLEU	0.932	0.968	0.973	0.912	0.239	0.805
BLEU	0.937	0.973	0.976	0.915	0.216	0.803
TER	0.954	0.829	0.978	0.931	0.324	0.803
PER	0.936	0.931	**0.988**	**0.941**	0.190	0.797
BLEU NRC	0.933	0.971	0.974	0.901	0.205	0.797
WER	**0.960**	0.516	0.976	0.932	**0.357**	0.748

Table 3.24: WMT14 system-level (Trueskill) Pearson's correlation between automatic evaluation metrics and human judgements for translations out-of English

applied SIMPBLEU as part of a loss function for MERT training and the human judges ranked SIMPBLEU-tuned translations better than BLEU-tuned translation.

ROSE's overall performance was close to that of BLEU at document and sentence levels. However, it performed better on tasks it was specifically trained for, such as ROSE-rank at document-level and ROSE-regpos for the syntactic constituents task. Results also showed that when training data is not available for the language pair of interest, ROSE trained on data for a different language pair could still produce reasonable scores (only slightly worse than BLEU). Smoothed BLEU slightly outperformed ROSE for sentence-level evaluation. This might be due to the fact that the rankings in the training data are not expert judgements, and consequently can be very noisy for model learning.

In SIMPBLEU, we analysed and tested components in BLEU. In order to address the shortcomings of BLEU at sentence level, we experimented with variants of the metric. Results showed that sentence-level BLEU variants underperform their document-level counterparts, and that precision-based metrics often have better performance

	fr-en	hi-en	cs-en	ru-en	de-en	avg
DISCOTK-PARTY-TUNED	0.977	**0.943**	0.956	0.975	0.870	**0.944**
LAYER	0.973	0.893	**0.976**	0.941	0.854	0.927
DISCOTK-PARTY	0.970	0.921	0.862	0.983	0.856	0.918
UPC-STOUT	0.968	0.915	0.898	0.948	0.837	0.913
VERTA-W	0.959	0.867	0.920	0.934	0.848	0.906
VERTA-EQ	0.959	0.854	0.927	0.938	0.842	0.904
TBLEU	0.952	0.932	0.954	0.957	0.803	0.900
BLEU NRC	0.953	0.823	0.959	0.946	0.787	0.894
BLEU	0.952	0.832	0.956	0.909	0.789	0.888
UPC-IPA	0.966	0.895	0.914	0.824	0.812	0.882
PABC3	0.959	0.856	0.940	0.799	0.848	0.880
CDER	0.954	0.823	0.826	0.965	0.802	0.874
APAC	0.963	0.817	0.790	0.982	0.816	0.874
REDSYS	**0.981**	0.898	0.676	0.989	0.814	0.872
REDSYSSENT	0.980	0.910	0.644	**0.993**	0.807	0.867
NIST	0.955	0.811	0.784	0.983	0.800	0.867
DISCOTK-LIGHT	0.965	0.935	0.557	0.954	0.791	0.840
METEOR	0.975	0.927	0.457	0.980	0.805	0.829
TER	0.952	0.775	0.618	0.976	0.809	0.826
RABC2	0.958	0.549	0.935	0.806	**0.876**	0.825
WER	0.952	0.762	0.610	0.974	0.809	0.821
AMBER	0.948	0.910	0.506	0.744	0.797	0.781
PER	0.946	0.867	0.411	0.883	0.799	0.781
ELEXR	0.971	0.857	0.535	0.945	-0.404	0.581

Table 3.25: WMT14 system-level (Trueskill) Pearson's correlation between automatic evaluation metrics and human judgements for translations into English

than recall-based metrics. In addition, arithmetic mean outperformed geometric mean consistently across languages and for sentence and document-level evaluation. The smoothing parameters are less important for sentence-level evaluation with arithmetic mean; for geometric mean, smoothing both in n-gram precision and brevity penalty leads to better performance.

Our experiments also showed that higher n-grams appear to be unnecessary, and

clipping is only important when using lower order n-grams. However, this did not hold for tuned models, suggesting that the human evaluation data from WMT is heavily biased towards similar SMT models (those trained on BLEU).

4

Development Data Selection For Unseen Test Sets

The quality of discriminative training in Statistical Machine Translation is heavily dependent on the quality of the development corpus used, and on its similarity to the test set. This chapter introduces a novel development corpus selection algorithm – the LA selection algorithm. It focuses on the selection of development corpora to achieve better translation quality on unseen test data and to make training more stable across different runs, particularly when manually created development sets are not available, and for selection from noisy and potentially non-parallel, large scale web-crawled data. The LA selection algorithm does not require knowledge of the test set, nor the decoding of the candidate pool before the selection. In our experiments, development corpora selected by the LA algorithm lead to improvements of over 2.5 BLEU points when compared to random development data selection from the same larger datasets.

4.1 Introduction

Discriminative training quality is closely related to the quality of training samples in the development corpus and, to a certain extent, to the proximity between this corpus and the test set(s). In their experiments, Hui *et al.* (2010) demonstrate that by using different development corpora to train the same SMT system, translation performance can vary up to 2.5 BLEU points using a standard phrase-based system (Koehn *et al.*, 2007). Gains obtained from a sensible choice of development data have been shown to

be even greater in a cross-domain setting (Pecina *et al.*, 2012). Building a 'suitable' development corpus is an important problem in SMT discriminative training.

A suitable development corpus should aid discriminative training to achieve better quality, and thus yield better translations. Previous research on selecting training samples for the development corpus can be grouped into two categories: i) selecting samples based on the test set (transductive learning), or ii) selecting samples without knowing the test set (inductive learning). Research in the first category focuses on how to find samples similar to the ones on which the system will be tested. Li *et al.* (2010), Lu *et al.* (2008), Zheng *et al.* (2010), and Tamchyna *et al.* (2012) measure similarity based on information retrieval methods, while Zhao *et al.* (2011) select similar sentences based on edit distance. These similarity-based approaches have been successfully applied to the local discriminative algorithm proposed in Liu *et al.* (2012). The disadvantage of these approaches is that the test set needs to be known before model building, which is most often an artificial scenario.

Our research belongs to the second category. Previous work on development data selection for unknown test sets include Hui *et al.* (2010); Cao & Khudanpur (2012). Hui *et al.* (2010) suggest that training samples with high oracle BLEU scores[1] will lead to better training quality. Cao & Khudanpur (2012) confirmed this finding and also demonstrated that better training data exhibits high variance in terms of BLEU scores and feature vector values between oracle and non-oracle hypotheses, arguing that these are more easily separable by the learning machine algorithms used for tuning. Both of the above approaches achieved positive results, but require decoding the candidate development data to obtain BLEU scores and feature values, which may be too slow if the pool for data selection is extremely large.

Another potential way of improving training quality based on a development corpus is to increase the size of this corpus. However, high-quality, sentence-aligned parallel corpora are expensive to obtain. In contrast to data used for rule extraction in SMT, data used for SMT discriminative training is required to be of better quality for reliable training. Development data is therefore often created by professional translators. In addition, increasing the development corpus size also increases the computational cost and the time required to train a model. It is therefore also important to determine

[1]Oracle BLEU scores are those computed for the closest candidate translation to the reference in the N-best list of the development set.

how much data is sufficient to build a suitable development corpus. Web crawled or crowd-sourced data is much cheaper than professionally translated data, and research exploiting these types of data (Zaidan & Callison-Burch, 2011; Uszkoreit *et al.*, 2010; Smith *et al.*, 2010; Resnik & Smith, 2003; Munteanu & Marcu, 2005; Smith *et al.*, 2013a) has already been successfully applied to machine translation, both in phrase extraction and discriminative training. However, they do not provide a direct comparison between their selected data and professionally built development corpora.

In order to address these problems, in this thesis we introduce a novel development corpus selection algorithm, the **LA Selection** algorithm. **LA Selection** combines sentence length, bilingual alignment and other textual clues, as well as data diversity for sample sentence selection. It does not rely on knowledge of the test sets, nor on the decoding of the candidate sentences. Our results show that the proposed selection algorithm achieves improvements of over 2.5 BLEU points compared to random selection. We also present experiments with development corpora for various datasets to shed light on the following aspects that might have an impact on translation quality as well as on the stability of the results over different runs: namely, that sentence length has substantial effect on the development corpus, and that if the right selection process is chosen, large development corpora offer fewer benefits over smaller ones.

The remainder of this chapter is structured as follows: We will describe our novel LA selection algorithm in Section 4.2. Experiments and results are presented in Sections 4.3 and 4.4, respectively, where we also discuss the training quality and scalability across different corpus sizes.

4.2 LA Selection Algorithm

The proposed development corpus selection algorithm is comprised of two main steps: (i) selecting training sentence pairs by sentence **L**ength, and (ii) selecting training sentence pairs by **A**lignment and other textual clues. We call it **LA selection**. It also has an additional step to reward diversity in the set of selected sentences with respect to the words they contain. The LA algorithm assumes that a good training sample should have a 'reasonable' length, be paired with a good quality translation, as primarily indicated by the word alignment clues between the sentences in the candidate pair, and enhance the existing set in terms of diversity.

Algorithm 4.1 LA Development Data Selection Algorithm

Require: Data Pool $D = (f^t, r^t, a^t)_{t=1}^{T}$, Number of words N, length limits λ_{low} and λ_{top}

1: $Selected = ()$, $Tmp = ()$
2: **for** $d_i = (f^i, r^i, a^i)$ in D **do**
3: **if** $length(f^i) > \lambda_{low}$ and $length(f^i) < \lambda_{top}$ **then**
4: Extract features $featureList$ from d^i
5: Calculate feature score $featureScore$ according to $featureList$
6: Add $(featureScore, d^i)$ to Tmp
7: **end if**
8: **end for**
9: Sort Tmp according to $featureScore$ from high to low
10: **while** Selected length $LS < N$ **do**
11: **for** d^i in Tmp **do**
12: **if** $maxSimi(f^i, Selected[f^j]_{j=J-200}^{J}) < 0.3$ and $simi(f^i, r^i) < 0.6$ **then**
13: Add (f^i, r^i) to $Selected$
14: $LS = LS + length(f^i)$
15: **end if**
16: **end for**
17: **end while**
18: **return** $Selected$

LA selection is shown in Algorithm 4.1. Assume that we have T sentence pairs in our data set D. Each sentence pair d_i in D contains a foreign sentence f^i, a translation of the foreign sentence r^i and the word alignment between them a^i. We first filter out sentence pairs below the low length threshold λ_{low} and above the high length threshold λ_{top} (Line 3). Sentence length has a major impact on word alignment quality, which constitutes the basis for the set of features we use in the next step. Shorter sentences tend to be easier to align than longer sentences, so our algorithm would naturally be biased to selecting shorter sentences. However, as we show later in our experiments, sentences that are either too short or too long often have a negative effect on training quality. Therefore, it is important to set both upper and lower thresholds on sentence length. Based on empirical results, we suggest set $\lambda_{low} = 10$ and $\lambda_{top} = 50$, as we will discuss in more detail in Section 4.4.1.

After filtering out sentences using the length thresholds, we extract the feature

+/-	**Alignment Features**
+	Source alignment ratio (LAR)
+	Target alignment ratio (RAR)
+	Source & target alignment ratio (TAR)
-	Top three largest fertilities ratio (AFer.1 ... 3)
+	Source largest contiguous span ratio (SLCSR)
+	Target largest contiguous span ratio (TLCSR)
-	Source largest discontiguous span ratio (SLDSR)
-	Target largest discontiguous span ratio (SLDSR)
	Text-only Features
+	Source and target length ratio (STLR)
-	Target function word penalty (TFWP)

Table 4.1: Features used to score candidate sentence pairs

values for each remaining candidate sentence pair. The features used in this thesis are listed in Table 4.1. The first column of the Table contains the sign of the feature value, where a negative sign indicates that the feature will return a negative value, and a positive sign indicates that the feature will return a positive value. The actual features, which we describe below, are given in the second column. These include word alignment features, which are computed based on GIZA++ alignments for the candidate development set, and simpler textual features. The alignment features used here are primarily adapted from Munteanu & Marcu (2005):

The **alignment ratio** is the ratio between the number of aligned words and length of the sentence in words:

$$\text{Alignment Ratio} = \frac{\text{No. Aligned Words}}{\text{Sentence Length}}$$

A low alignment ratio means that the data is most likely non-parallel or represents a highly non-literal translation. In both cases, these sentence pairs are likely to prove detrimental to discriminative training.

Word fertility is the number of foreign words aligned to each word. The **word fertility ratio** is the ratio between word fertility and sentence length. We use the top three largest fertility ratios as three features:

$$\text{Fertility Ratio} = -\frac{\text{Word Fertility}}{\text{Sentence Length}}$$

This feature can detect a common feature of unsupervised alignment algorithms: garbage collection, whereby the aligner uses a rare word in one sentence to erroneously account for many difficult words to align in the parallel sentence.

Our definition of **contiguous span** differs from that of Munteanu & Marcu (2005): we define it as a substring in which all words have an alignment to words in the other language. A **discontiguous span** is defined as a substring in which none of the words has an alignment to any word in the other language. The **contiguous span ratio**, CSR, is the length of the longest contiguous span over the length of the sentence:

$$CSR = \frac{LC}{\text{Sentence Length}}$$

The **discontiguous span ratio**, $DCSR$, is the length of the longest discontiguous span over the length of the sentence:

$$DCSR = -\frac{LDC}{\text{Sentence Length}}$$

where LC is the length of the contiguous span and LDC is the length of the discontiguous span.

In addition to the word alignment features, we use **source and target length ratio**, LR, to measure how close the source and target sentences in the pair are in terms of length:

$$LR = \begin{cases} \frac{TL}{SL} & \text{if } SL > TL \\ \\ \frac{SL}{TL} & \text{if } TL > SL \end{cases}$$

where TL is target sentence length and SL is source sentence length.

Finally, the **target function words penalty**, FP, penalises sentences with a large proportion of function words or punctuation tokens:

$$FP = e^{-\frac{nfw}{TL}}$$

where nfw is the number of function words or punctuation tokens, and TL is the target sentence length. We only consider a target language penalty for practical reasons, but a source language penalty could also be used.

Once we obtain these feature values for all candidate sentence pairs, we apply two approaches to calculate an overall score for the candidate. The first is a heuristic

approach, which simply sums up the scores of all features for each sentence (with some features negated as shown in Table 4.1). The second approach uses machine learning to combine these features, similar to what was done in Munteanu & Marcu (2005) to distinguish between parallel and non-parallel sentences. Here, a binary SVM classifier is trained to predict samples that are more similar to professionally created sentences. The labelling of the data was therefore done by comparing professionally created translations against badly aligned translations from web-crawled data. The heuristic approach achieved better performance than the machine learning approach, as we will discuss in Section 4.4.3.

Line 9 through Line 17 in Algorithm 4.1 describe the sentence pair selection procedure based on this overall feature score. The candidate sentence pair and its features are stored in the Tmp list, and sorted from high to low according to their overall feature scores. The algorithm takes candidate sentence pairs from the Tmp list until the number of words in the selected development corpus $Selected$ reaches the limit N. If the candidate sentence pair passes the test in Line 12, the sentence pair will be added to the selected corpus $Selected$.

Line 12 has two purposes: first, it aims at increasing the diversity of the selected development corpus. Based on our experiments, candidate sentence pairs with similar feature scores (and thus similar rankings) may be very similar sentences, with most of their words being identical. We therefore only select a sentence pair whose source sentence has less than 0.3 BLEU similarity when compared to the source sentences in the last 200 selected sentence pairs to reduce computational complexity (the computational cost increases linearly with the number of selected words, so we set this threshold to ensure fast runtime). The second purpose is to filter out sentence pairs that are not translated, i.e., sentence pairs with the same words in the source and target sides. Untranslated sentence pairs are a problem in web-crawled data, therefore we filter out sentence pairs whose source and target have a BLEU similarity score of over 0.6. [1]

[1]Source and target sentences with high BLEU similarity have a high number of matching words and n-grams; this is unlikely in a translation.

4.3 Experimental Settings

We build a standard phrase-based SMT systems using Moses with its default features. Word alignment and language model are obtained using GIZA++ and IRSTLM with their default settings. For discriminative training we use the MERT (Och, 2003) algorithm. Two language pairs are used in the experiments: French-English and Chinese-English.

4.3.1 French-English Data

To build a French to English system we used the Common Crawl corpus (Smith *et al.*, 2013b). We filtered out sentences with a length of over 80 words and split the corpus into training (Common Crawl training) and tuning (Common Crawl tuning) sets. The **training** subset was used for phrase table, language model and reordering table training. It contains $3,158,523$ sentence pairs (over 161M words) and average source sentence length of 27 words. The **tuning** subset used was the "Noisy Data Pool" to test our LA selection algorithm. It contains $31,929$ sentence pairs (over 1.6M words), and an average source sentence length of 27 words. We compared the performance of our selected corpora against a concatenation of four professionally created development corpora (Professional Data Pool) for the news test sets distributed as part of the WMT evaluation Callison-Burch *et al.* (2008, 2009, 2010): 'newssyscomb2009', 'newstest2008', 'newstest2009' and 'newstest2010'. Altogether, they contain $7,518$ sentence pairs (over 392K words) with an average source sentence length of 27 words. As **test data**, we took the WMT13 (average source sentence length = 24 words) and WMT14 (average source sentence length = 27 words) news test sets.

4.3.2 Chinese-English Data

To build the Chinese to English translation system we used the non-UN and non-HK Hansards portions of the FBIS (LDC2003E14) training corpus ($1,624,512$ sentence pairs, over 83M words, average source sentence length = 24 words) and **tuning corpus** ($33,154$ sentence pairs, over 1.7M words, average sentence length = 24). The professionally created development corpus in this case is the NIST MT2006 test set[1]

[1]It contains 4 references, but we only apply the first reference to make it comparable to our selection algorithm.

(1,664 sentence pairs, 86K words, average sentence length = 23 words). As **test data**, we used the NIST MT08 test set (average source sentence length = 24 words).

Recall that for both language pairs, the test sets and professionally created development corpora belong to the same domain, namely news, for both French-English and Chinese-English. In addition, the test and development corpora for each language pair have been created in the same fashion, following the same guidelines. Our pool of noisy data, however, includes not only a multitude of domains that differ from news, but also translations created in various ways as well as noisy data.

4.4 Results

Our experiments are split into six parts: Section 4.4.1 examines how sentence length in development corpora affects the training quality; Section 4.4.2 presents an ablation study of features in the LA algorithm; Section 4.4.3 compares our LA selection algorithm against randomly selected corpora and against professionally created corpora; Section 4.4.4 explores different diversity filter thresholds; Section 4.4.5 focuses on the performance of the LA algorithm where features are combined using machine learning; and Section 4.4.6 discusses the effect of development corpus size by testing translation performance with corpora of different sizes.

4.4.1 Selection by Sentence Length

In order to test how sentence length affects the quality of discriminative training, we split the development corpus into six parts according to **source sentence length**[1] ranges (in words): [1-10], [10-20], [20-30], [30-40], [40-50] and [50-60]. For each range, we randomly select sentences to total 30,000 words as a small training set, we train a discriminative model based on the small training set and we test the translation performance on WMT13 and NIST MT08 test set. We repeat the random selection and training procedure five times (Clark *et al.*, 2011) and report average BLEU scores in Tables 4.2 and 4.3.

Table 4.2 shows the results for French-English translation. From this table, we can see that corpora with sentence lengths of [30-40] and [30-50] lead to better translation quality than random selection, with a maximum average BLEU score of 25.62 for

[1]The sentence length we use later in this chapter is also based on the source sentence

	Rand.	**1-10**	**10-20**	**20-30**	**30-40**	**40-50**	**50-60**	**10-50**
avg.	24.36	22.85	23.61	24.43	25.62	24.62	22.94	25.54
std.	0.84	0.65	0.80	0.51	0.40	1.06	0.99	0.84

Table 4.2: Accuracy for random selection of development sentences with respect to sentence length on the French-English WMT13 news test set. Shown is the average BLEU score and the standard deviation measured over five runs

sentence length [30-40], outperforming random length selection by 1.26 BLEU points. Corpora with sentences in [10-20] and [20-30] perform slightly worse than random selection, while corpora with very short or very long sentences perform the worst.

Table 4.3 shows the results for Chinese-English translation. Corpora length in ranges [10-20], [20- 30], [30-40] and [40-50] lead to better translation performance than random selection. As for French-English translation, corpora with very short or very long sentences showed the worst performance, with a lower BLEU score than random selection.

Based on the above results, the best sentence length for discriminative training is not fixed, as it may depend on language pairs and corpus type. However, sentence lengths below 10 words and above 50 words lead to poor results for both language pairs. We conducted another experiment selecting development corpora that excluded sentences with length below 10 and above 50. The results are shown in column [10-50] of both Tables. Compared to random selection, length range [10-50] improved BLEU scores by 1.18 for French-English, and by 0.54 for Chinese-English. We therefore suggest avoiding sentence pairs with fewer than 10 or more than 50 words for discriminative training. Note that our systems were developed on corpora with an average sentence length of around 25 words, which is typical in most freely available training corpora, although the thresholds may differ for corpora with very different sentence lengths.[1]

4.4.2 Selection by LA Features

In this section we will test the features used in the LA selection algorithm in terms of their contribution to the algorithm. The results are shown in Table 4.4. The single

[1]For example, both Europarl and News-Commentary WMT corpora have an average of 25 words on their English side.

	Rand.	1-10	10-20	20-30	30-40	40-50	50-60	10-50
avg.	18.79	18.11	20.00	19.63	18.85	19.29	18.53	19.33
std.	0.83	0.29	1.45	1.00	0.85	1.38	0.81	1.16

Table 4.3: Accuracy for random selection of development sentences with respect to sentence length on the Chinese-English MT08 test set. Shown is the average BLEU score and the standard deviation measured over five runs

	Rand.	AFer.1	AFer.2	AFer.3	TFWP	STLR	SLCSR
avg.	24.36	24.28	24.16	23.17	22.13	23.65	23.08
std.	0.84	0.13	0.24	0.24	0.66	0.11	0.15
	TLCSR	**SLDSR**	**TLDSR**	**LAR**	**RAR**	**TAR**	**All Features**
avg.	22.94	23.13	23.32	23.05	23.17	25.09	25.88
std.	0.59	0.22	0.42	0.09	0.88	0.20	0.16

Table 4.4: Accuracy for development sentences selection with respect to LA feature only on the French-English MWT13 test set. Shown is the average BLEU score and the standard deviation measured over five runs

most informative feature is the Total Align Ratio (TAR), leading to an average BLEU score of 25.09. Apart from this feature, no other feature on its own outperformed random selection. However, when combining all of these features we achieved a BLEU score improvement of 0.79 over that achieved by TAR alone, in addition to relatively low standard deviation.

Some features are clearly more informative than others. For example, the largest fertility ratio (AFer1) leads to better BLEU performance than the second and third largest fertility ratios (AFer2 and AFer3). Although our heuristic LA selection combines all features uniformly, it is likely that non-uniform weighting could lead to even better results.

4.4.3 Selection by LA Algorithm

In what follows we compare the performance of our LA selection algorithm against randomly selected and professionally created corpora. We set $\lambda_{low} = 10$ and $\lambda_{top} = 50$ and select a development corpus with no more than $30,000$ words. The results are

reported in Tables 4.5, 4.6 and 4.7, again reporting statistics over five runs.

Table 4.5 shows the results for the French-English WMT13 test set. Note that LA selection improves BLEU by 1.36 points compared to random selection, and also improves over using sentence length (10-50) only as selection criterion. The performance of the LA selected corpus is slightly lower (0.1 BLEU) than that of the professionally created corpus (Prof.), but the system is much more robust with much lower standard deviation (std). The fact that the results are so close is surprising: The professionally created development sets were drawn from the same domain as the test sets (news), and were created using the same translation guidelines as the test set. Findings for the Fr-En WMT14 test set (Table 4.6) and Chinese-English MT08 (Table 4.7) are similar. Systems trained on corpora selected by LA increase by 1.21 and 2.53 BLEU points over random selection, respectively. For the WMT14 test set, the corpus selected by LA show slight improvements over the professionally created corpus (26.40 vs. 26.31) with a lower variance.

	Rand.	**10-50**	**LA10-50**	**Prof.**
avg.	24.36	25.54	25.72	25.82
std.	0.84	0.84	0.01	0.23

Table 4.5: Accuracy comparing LA selection method with benchmark strategies on the French-English WMT13 news test. Shown are BLEU scores and std. dev. when using development corpora selected by length (10-50), the LA selection algorithm (LA10-50), randomly (Rand.), and a corpus created by professionals (Prof.)

	Rand.	**10-50**	**LA10-50**	**Prof.**
avg.	25.19	25.31	26.40	26.31
std.	0.30	0.14	0.04	0.16

Table 4.6: Accuracy comparing LA selection method with benchmark strategies on the French-English WMT14 news test. Shown are BLEU scores and std. dev. when using development corpora selected by length (10-50), the LA selection algorithm (LA10-50), randomly (Rand.), and a corpus created by professionals (Prof.)

	Rand.	10-50	LA10-50	Prof.
avg.	18.79	19.33	21.32	23.49
std.	0.83	1.16	0.83	0.31

Table 4.7: Accuracy comparing LA selection method with benchmark strategies on the Chinese-English MT08 test. Shown are BLEU scores and std. dev. when using development corpora selected by length (10-50), the LA selection algorithm (LA10-50), randomly (Rand.), and a corpus created by professionals (Prof.)

4.4.4 Diversity Filter

This experiment tests the effect of the different diversity filter thresholds, which range from 0.2 to 1. A low diversity filter threshold results in a selected development corpus with high diversity (i.e., more unique words) but a lower overall score for the other features in the LA selection method. A threshold equal to 1 means no diversity filtering is performed. The results are shown in Tables 4.8 and 4.9, where sentence lengths are allowed to range from 1 to 80 words.

Table 4.8 shows the results for Chinese-English. Notice that the corpora with the least diversity filtering leads to the worst performance, with thresholds of 0.8-1.0 achieving BLEU scores of around 20. Decreasing the threshold until 0.6 leads to BLEU score increases of up to 21.49. This result is almost 1.5 points (absolute) higher than with no filtering. The BLEU score drops as the threshold continues to decrease. This can be attributed to the fact that increasing the diversity forces the algorithm to select sentences from the pool that are deemed bad by the other features, such as sentence pairs with poor alignments.

Table 4.9 illustrated the results for French-English. The best BLEU score is also achieved when the threshold is 0.6 (BLEU 25.57), but this result is not significantly better than for other thresholds. We believe that this is due to the fact that the French-English data pool has higher diversity than the Chinese-English data, so that incorporating diversity into our selection algorithm has little effect. To validate this claim, we investigated the diversity of the selected sets: without diversity filtering, the French-English selected corpus contains $6,229$ unique words while the Chinese-English corpus only contains $5,234$ unique words (see the 'words' row in Tables 4.8 and 4.9). When increased to the best performance threshold of 0.6, the Chinese-English corpus

added 460 new words, while for French-English only 48 new words were added. This confirms that the French-English corpus has higher diversity to start with, and that the diversity strategy for French-English had only limited effect.

	0.2	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
avg.	20.49	20.98	20.99	21.24	21.49	20.64	20.08	19.88	20.07
std.	0.29	0.20	0.28	0.15	0.25	0.24	0.33	0.58	0.53
words	6213	5917	5808	5756	5694	5598	5360	5303	5234

Table 4.8: Performance with differing diversity thresholds, evaluated on the Chinese-English MT08 test set. Lower values of the threshold indicate greater diversity. We report average BLEU, std. deviation and number of unique words in selection

	0.2	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1.0**
avg.	25.48	25.37	25.39	25.45	25.57	25.45	25.49	25.49	25.55
std.	0.22	0.19	0.16	0.19	0.18	0.17	0.08	0.07	0.15
words	7417	6473	6298	6280	6277	6275	6262	6244	6229

Table 4.9: Performance with differing diversity thresholds, evaluated on the French-English WMT13 news test set. Lower values of the threshold indicate greater diversity. We report average BLEU, std. deviation and number of unique words in selection

4.4.5 Machine Learned Approach

We also experiment with using the SVM classifier to combine features in the LA selection algorithm, as previously discussed. The classifier was trained using the SVMlight[1] toolkit with RBF kernel with its default parameter settings. We selected $30,000$ words from the professionally created WMT development corpus as positive training samples, and used the sentence pairs from our corpus with the lowest LA selection score as negative training examples, selecting $30,000$ words worth of data as our negative examples in order to balance the two classes. Results for sentence selection using the examples with the highest classification scores (i.e. distance for hyperplane in direction of positive class) are shown in Tables 4.10 and 4.11.

[1] `http://svmlight.joachims.org/`

The LA selection method with the SVM classifier outperforms random selection, but does worse than our heuristic approach. A reason may be the quality of the training data: both our positive and negative training examples will contain considerable noise. The professionally created WMT corpora include some odd translations, so the alignment features will be less reliable. Moreover, this is a harder problem than the one introduced in Munteanu & Marcu (2005), since their pool of candidate samples contained either parallel or non-parallel sentences, which are easier to label and to distinguish based on word alignment features. Our pool of candidate samples are parallel, with our selection procedure aiming to select the highest quality translations from this pool.

	WMT13 test set		
	SVM	**RANDOM**	**LA heuristic**
avg.	25.42	24.36	25.54
std.	0.08	0.84	0.01
	WMT14 test set		
	SVM	**RANDOM**	**LA heuristic**
avg.	26.08	25.19	26.40
std.	0.08	0.30	0.04

Table 4.10: Performance of SVM-trained LA selection versus heuristic LA selection on the French-English WMT13 and WMT14 news test sets. Also shown is random selection, reporting average BLEU and std. deviation over five independent runs

	NIST test set		
	SVM	**RANDOM**	**LA heuristic**
avg.	20.33	18.79	20.92
std.	1.45	0.83	0.43

Table 4.11: Performance of SVM-trained LA selection versus heuristic LA selection on the Chinese-English NIST08 test set. Also shown is random selection, reporting average BLEU and std. deviation over five independent runs

4.4.6 Effect of Development Corpus Size

Now we consider the question of how much development data is needed to train a phrase-based SMT system. To test this, we experiment with corpora containing between $10,000$ words (about 500 sentences) and $150,000$ words ($7,500$ sentences), with an incremental step of $10,000$ words. We run MERT training five times on each increment and report the average BLEU scores. The test set is the WMT13 news test.

Figure 4.1 shows how BLEU changes as we increase the development corpus size. The three lines represent the BLEU scores of three systems: Random selection from the French-English tuning dataset (blue line), LA selection from the same pool (red line), and professionally created WMT development corpus (green line). Note that performance increases as corpora sizes increase for all techniques up to $70,000$ words ($2,000$-$3,000$ sentence), after which performance is stable. The professionally created corpus achieves the best performance regardless of corpus size. Note, however, that the LA selection technique is only slightly worse, with less than 0.1 BLEU difference, for corpora sizes $\geq 30,000$ words. Random selection clearly performs poorly compared to both.

Figure 4.2 shows the standard deviation over five runs for the same experiment. Random selection presents the largest standard deviation (greater than 0.6 BLEU) for training corpora of sizes below $50,000$ words. The maximum standard deviation is 1.93 at $30,000$ words. With larger development corpus sizes, the standard deviation of random selection is still higher than that of LA selected and professional data. LA selection has a much lower average standard deviation, which is mostly lower even than for the professionally created data.[1] This is important for real application settings, where repeated runs are not practical and robust performance from a single run is imperative.

These results confirm the findings of Hui *et al.* (2010). Increasing the amount of data is not the best solution when creating a development corpus. Better data – rather than more data – leads to better training quality. A development corpus with 30k to 70k words is enough to produce stable translation results in our setting.

[1]Given a specific pool of sentences, the LA algorithm is deterministic, so any random effects are purely derived from the discriminative training process. For the randomly and professionally selected data, however, are affected by different sets selected from the pool and this may lead to higher standard deviations.

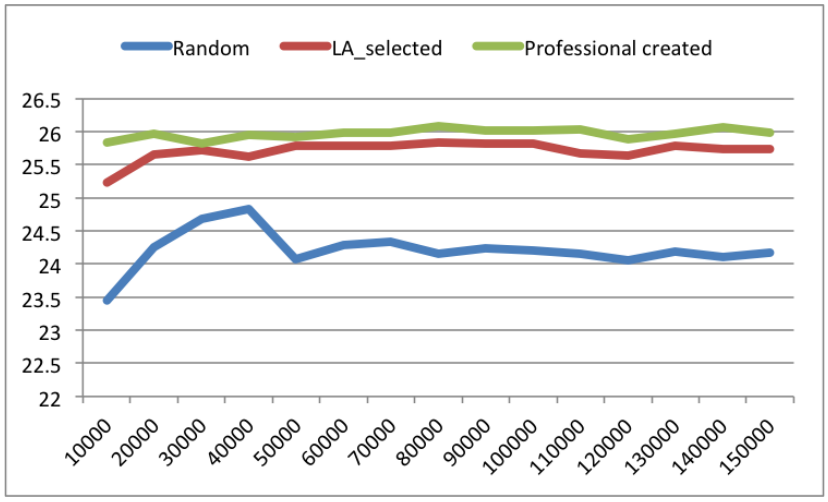


Figure 4.1: Accuracy of development selection algorithms with increasing sizes of development corpora. The horizontal axis shows corpus size, and the vertical axis, BLEU scores, evaluated on the French-English WMT13 news test set. The three curves denote random selection, our proposed LA selection algorithm and professional translation

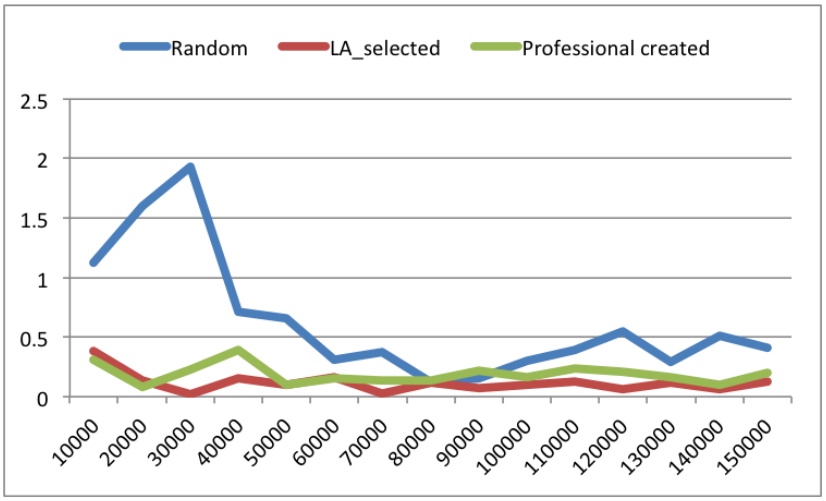


Figure 4.2: Standard deviation of the accuracy for the development selection method with increasing sizes of development corpora, evaluated on the French-English WMT13 news test set. The horizontal axis shows corpus size, and the vertical axis, standard deviation over BLEU results

4.5 Summary

In this chapter we demonstrated how the choice of the development corpus is critical for good discriminative training performance. The standard practice of sourcing expensive human translations is not practical for many SMT application scenarios, and consequently making better use of existing parallel resources is paramount. Length is the most important single criterion for selecting effective sentences for discriminative training: overly short and overly long training sentences often harm the training performance. Using large development sets brings only small improvements in accuracy, and a modest development set of 30k-70k words is sufficient for good performance. Out main contribution in this chapter, the LA sentence selection algorithm, selects high quality and diverse sentence pairs for training. We showed improvements over random selection of up to 2.5 BLEU points (Chinese-English). This approach is competitive with manually selected development sets, despite having no knowledge of the test set, test domain, and without engaging expert translators. In future work, we plan to improve the classification technique for automatically predicting training quality by means of alternative methods for extracting training examples and additional features to distinguish between good and bad translations.

5

Weighted Ranking Optimisation

A number of SMT discriminative training algorithms have been proposed in the past 10 years since the development of Och & Ney (2002)'s model. These go from maximum likelihood training (Och & Ney, 2002) and minimum error rate training (Och, 2003) to margin based training (Tillmann & Zhang, 2006; Watanabe *et al.*, 2007), and most recently pairwise ranking based training (Hopkins & May, 2011). Hopkins & May (2011)'s Pairwise Ranking Optimisation (PRO) shows two main advantages over other training algorithms:

- It scales well for large feature spaces while still reliable in small ones.
- It can be easily adapted to any SMT decoder without changing decoding methods. PRO's framework is similar to MERT, and does not require special decoding strategies such as forced decoding.

As a result, many SMT researchers find that the PRO algorithm is a good substitute for the MERT algorithm.

The PRO algorithm aims to rank translation candidates in correct order. However, since the candidate space for one training sentence is too large, PRO performs random sampling of Ξ candidate pairs for training, with a common number of samples $\Xi = 50$ Hopkins & May (2011). Compared with the entire candidate space, this number is extremely small, and thus the samples should be chosen in a sensible way. In our research, the sample selection considers the following criteria:

- The correctness of the sampled pair is measured by automatic evaluation metrics. These metrics may not always indicate the true correctness of the candidates.

- Because of the limited number of samples and the limitations of LLM itself, the most important candidates should have higher priority in sample selection. For example, the best (metric score) candidate in the N-best list is more important than other candidates since the aim of SMT discriminative training is to adjust the system to produce the best candidate.

- Samples should have different weights according to their importance. For example, samples from unreachable training instances should be less important than other samples. Unreachable sentences indicate the inability of the SMT system to translate them and we may not be able to learn useful information from this kind of sentence. Also, samples that are more closely related to the test sentences should obtain higher weights than other samples to reduce over-fitting.

The sampling strategy of the PRO algorithm only considers the first issue by trying to increase the metric difference between two candidates in order to have confidence in the correctness measurement. The second and third issues are ignored in PRO. In this chapter, we introduce Weighted Ranking Optimisation (WRO) to take these three issues into account. Our WRO algorithm includes two parts: the first part is Weighted Ranking Optimisation Global (WRO-global), which is a global off-line training algorithm. This part is similar to other global training algorithms such as PRO, insofar as the test sentence is unknown and one global weight can be trained by WRO-global to be used to translate any unseen segment. The second part is the Weighted Ranking Optimisation Local (WRO-local), a local online training algorithm which, in contrast to the global algorithm, trains weights for each test sentence.

5.1 Weighted Ranking Optimisation – Global

This section introduces the Weighted Ranking Optimisation Global (WRO-global) algorithm, which is used for off-line global weight optimisation and prior sample selection for WRO-local (which will be introduced in Section 5.2). Before going into the details of WRO-global we first give a brief reminder of PRO algorithm (the details of the PRO algorithm were described in Chapter 2). PRO treats SMT discriminative training as a binary classification problem where the goal is to classify a pair of candidates as best and worst ranked and the ranking is determined by an evaluation metric against a

reference. The candidate pairs are uniformly randomly selected from the N-best list, and the candidate pairs with small BLEU difference (less than 0.05) are discarded to ensure the reliability of the ranking correctness.

PRO has at least two limitations in the sampling phase. First, PRO's random sampling is not the optimum way for selecting samples since the target is not clear. As we only select a small sample from the whole space, a clearer target should give better training quality. We will call these targets **oracles**: in WRO, the oracles are the top 10 percent of all candidates in the N-best list in terms of metric score. The second problem is that all sampled sentences are equally important. Although we select the same number of samples for each training sentence, there are certainly differences among sentences. For example, as discussed in the previous section, reachable sentences can be more important than unreachable ones.

Our WRO algorithm focuses on these two limitations of PRO. The WRO-global procedure is shown in Algorithm 5.1. Similar to PRO, we use N-best list Nb as one of our candidate pools for sample selection. We also create another list called oracle list, Nb_{oracle}. We select the top 10 percent of all candidates in the N-best list with the highest metric score as oracles and store them in the oracle list.

The sampling procedure includes two steps: first, a Γ number of candidate pairs $\{e_s, e'_s\}$ are randomly selected from the two lists, where e_s and e'_s are represented by their corresponding feature values $h(e_s)$ and $h(e'_s)$. Contrary to PRO, WRO focuses on ranking the oracle translations in the correct order among all candidates. In this case, we define the candidate e_s as an oracle that is randomly selected from the oracle list Nb_{oracle}, and e'_s is the non-oracle that is randomly selected from the N-best list Nb. We select e'_s from whole N-best list (if e'_s is also included in the top 10 percent candidates with highest metric score, then the candidate with the better metric score candidate is considered oracle). The selected candidates are then evaluated by an automatic evaluation metric $metric(.)$. The sampled pair with a metric difference (i.e. $metric(e_s) - metric(e'_s)$) below the threshold will be discarded to ensure the confidence of the measurement. After the first step, we choose additional Ξ pairs with the greatest metric difference to generate our training instances.

The training instance and its label generation is the same as in PRO, except that we also add a global weight ($weight_{Global}$) to each training instance to indicate its

Algorithm 5.1 Weighted Ranking Optimisation – Global

Require: Development corpus $D = (f^t, r^t)_{s=1}^{S}$, Initial random weights Λ_0, $\Gamma = 5000$, $\Xi = 50$

for i_{th} iteration $K = (1, 2, \ldots, k)$ iterations **do**
  MegaM Training instances $R = \{\}$
  for s_{th} sentence pair development corpus D **do**
    Calculate global weight according to Equation 5.3
    $r_s = \{\}$
    Generate N best list Nb according to current weight Λ_i
    Copy the top 10 percent best BLEU candidates in Nb to Nb_{top}
    while length$(r_s) < \Gamma$ **do**
      random select candidate e_s from Nb_{top}
      random select candidate e'_s from Nb
      if $|score(e_s,) - score(e'_s,)| >$ threshold **then**
        generate samples x according to Equation 5.1
        add sample x to r_s
      end if
    end while
    Sort s according to $|score(e_s,) - score(e'_s,)|$
    Add Ξ samples with largest BLEU difference in r_s to R
  end for
  Update weights Λ_{i+1} according to R by MegaM Optimiser
end for
return Λ_{i+1}, R

importance. In this case our training instances are:

$$\{+, weight_{Global}, h(e_s) - h(e'_s)\} \text{ if } metric(e) - metric(e') > 0 \tag{5.1}$$

$$\{-, weight_{Global}, h(e_s) - h(e'_s)\} \text{ if } metric(e) - metric(e') < 0 \tag{5.2}$$

The global weight $weight_{Global}$ is used to penalise the training samples generated from the unreachable training sentences. For the datasets in our experiments in this chapter, empirical results have shown that a translation dataset with a BLEU score of 0.4 has acceptable translation quality. Therefore, we downweight the training sentence exponentially if the oracle candidate BLEU score is below 0.4. The $weight_{Global}$

parameter is defined as:

$$weight_{Global} = \begin{cases} 1 & \text{if } BLEU_{Top} \geq 0.4 \\ e^{BLEU_{Top}-0.4} & \text{if } BLEU_{Top} < 0.4 \end{cases}, \tag{5.3}$$

where the $BLEU_{Top}$ is the oracle candidate BLEU score

After the sampling and training instance generation, we can optimise the weights by any off-the-shelf binary classifier that support weighted training instances. In our experiment, we use the MegaM (Daume, 2004) classifier, which is the same one used in PRO.

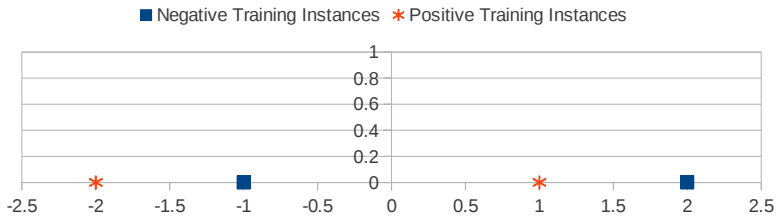
5.2 Weighted Ranking Optimisation – Local

This section introduces Weighted Ranking Optimisation Local (WRO-local), which is a transductive setting of WRO. The local feature weights are trained for each test sentence before translation. The reason we apply transductive learning here is because of the limitations of linear combination of features in Och's Model. Consider the example in Figure 5.1, where we have a development corpus containing two sentences 我要吃鱼 and 是八月十五号, and we generate two translation candidates for each sentence. By applying the global PRO algorithm we can obtain four classifier training samples, two positively labelled and two negatively labelled. We illustrate these training samples in Figure 5.1b, where squares indicate negative training samples and stars indicate positive training samples. From the figure we can see there is no single set of weights able to correctly classify both sentences. One option would be to add more features, but this would require feature engineering. Instead, we apply a transductive setting and optimise local parameters for *each* sentences.

Local based SMT discriminative training was first proposed by Liu *et al.* (2012). Their local training algorithm dynamically updates the development corpus according to the each input test sentence and retrains the feature weights based on the updated development corpus. They assume the existence of a large development data pool and a standard development corpus. Before translating a test sentence, the local training algorithm first compares the string similarity of the test among each sentence in the large development data pool and subsequently retrieves the N most similar sentences in the pool with the standard development corpus to generate the local development corpus.

Source	Candidates	Feature Vector	BLEU	PRO sample
我要吃鱼	Me want eat fish	<1,1>	0.3	{-, <-1,0>}
	I want eat fish	<2,1>	0.8	{+,<1,0>}
是八月十五号	it is august fifteenth	<1,2>	0.9	{+,<-2,0>}
	august 15th	<3,2>	0.5	{-,<2,0>}

a



b

Figure 5.1: Example of PRO training samples, where the x and y axis represent the feature values of the two translations

(Liu *et al.*, 2012)'s local training algorithm delivered better translation results than the global training algorithm. However, this algorithm requires an additional development data pool, which is harder to apply to resource poor language pairs. In our WRO-local approach, instead of retrieving more additional data, we re-weight the existing training sentences in the standard development corpus and we do not require additional development data.

The WRO-local procedure is shown in Algorithm 5.2. In order to improve efficiency in the online translation scenario, the training samples and global weights are pre-generated by WRO-global. As a result, we only need to load the pre-generated training samples and update the local weights. The local weight is the combination of the global weight and the string similarity between the test sentence and the development sentence. We define the local weight calculation as follows:

$$weight_{Local} = \frac{\alpha_1 \times weight_{Global} + \alpha_2 \times SIMI(f_s, t)}{\alpha_1 + \alpha_2}, \tag{5.4}$$

where α_1 and α_2 are the interpolation weights used to adjust the importance between

Algorithm 5.2 Weighted Ranking Optimisation – Local

Require: Develop Corpus $D = (f_1^s, r_1^s)_{s=1}^S$, Global Training Samples $R = (r_s)_{s=1}^S$, Test sentence t

1: **for** s_{th} sentence pair Training data D **do**
2: Compute Similarity score $simi(f_s, t)$
3: **for** Sample x in r_s **do**
4: Calculate $weight_{local} = \frac{\alpha_1 \times SIMI(f_s, t_k) + \alpha_2 \times weight}{\alpha_1 + \alpha_2}$
5: Replace $weight_{global}$ in x with $weight_{local}$
6: **end for**
7: **end for**
8: Update Λ_{local} according to R by MegaM Optimiser
9: **return** Λ_{local}

similarity and global weight. The string similarity is called transductive weight, which is measured by the BLEU metric as:

$$SIMI(f_s, t) = BLEU(f_s, t), \tag{5.5}$$

where the scaling is only used to avoid the effects of computations with very small scores.

5.3 Experiments and Results

For details on the experimental setup and language selection we refer the reader to Section 4.3 in Chapter 4, as the experiments in this section use the same settings with the randomly selected development set. The baseline SMT discriminative training algorithm is the Moses implementation of PRO. The N-best lists used in both PRO and WRO are obtained by running 16 iterations of PRO optimisation with default settings. We repeat the N-best lists generation process 5 times for each test and report the average results. Interpolation weights for local weights (Equation 5.4) α_1 and α_2 are 1 and 10 respectively. We also test the performance of each component in WRO, namely oracle selection, the reachability penalty (Equation 5.3) and the transductive setting (BLEU similarity term for the local weight). The settings used for each WRO variant are shown in Table 5.1.

	Description
Baseline PRO	Moses implementation of PRO
Global weight	Samples weighted with global weight only, without oracle selection
OS	Oracle selection only, samples are equally weighted
Trans	Samples weighted only with BLEU similarity (Equation 5.5)
Trans-OS	Samples weighted with BLEU similarity, with oracle selection
WRO-Global	WRO with global weight and OS
WRO-Local	WRO with local weight and OS

Table 5.1: Settings of the PRO and WRO variants tested in our experiments

	Average BLEU Score	Standard Deviation
Baseline PRO	21.27	0.058
Global weight	21.41	0.036
OS	21.53	0.038
Trans	21.69	0.078
Trans-OS	21.71	0.063
WRO-Global	21.51	0.035
WRO-Local	**21.74**	0.085

Table 5.2: BLEU results on the Chinese-English NIST MT08 test set. Boldface figure indicates the best BLEU score among all variants

Table 5.2 shows the performance for Chinese-English translation in each variant of WRO and PRO. WRO-local improves the BLEU score over the baseline PRO by almost 0.5. This procedure is composed of three components: oracle selection (OS), global weight, and transductive weight ('Trans' in the Table – note that the local weight is obtained by a combination of transductive weight and global weight - see Equation 5.4). According to the table, the baseline PRO algorithm can be improved by adding any of the components. The most effective component is the transductive weight, which improves BLEU by more than 0.4 when compared with PRO.

Table 5.3 shows the performance for French-English translation using the WMT13 test set. Similar to Chinese-English results, all WRO variants reach better performance than baseline PRO. Again, WRO-Local has the best average BLEU score (25.78). However, WRO-Local only brings a small improvement over WRO-Global (0.2 BLEU

	Average BLEU Score	Standard Deviation
Baseline PRO	25.65	0.031
WRO-Global	25.76	0.047
Trans	25.67	0.046
Trans-OS	25.72	0.058
WRO-Local	25.78	0.045

Table 5.3: BLEU results on the WMT13 French-English news test set

score improvement). One possible reason may be that the development corpus does not contain sentences that are similar to those in the test set. We compared the document level lexical (uni-gram) similarity between the test set and development set for both Chinese and French. The results show that in the French development corpus, the lexical similarity is 0.118, and in the Chinese development corpus, it is 0.103. In order to further investigate this problem we conducted a more challenging experiment in next section, where we test the training performance on the BTEC test set, which is not from the same domain as our training and development corpus.

5.3.1 Cross-domain Experiments

In this section we test WRO performance when translating an out-of-domain test set. The test set used in this section is the Basic Travel Expression Corpus (BTEC). The domain in this corpus is travel, with informal and colloquial language, while the domain of the FBIS corpus used before is mainly news, using formal written English. The main purpose of this experiment is to test the performance of the local training method in different domains.

The next experiment tests the WRO and baseline PRO BTEC translation performance on three different development corpora, which are: the FBIS tuning corpus (same tuning set as used in the last section), the official BTEC tuning corpus, a mixed corpus containing both BTEC and FBIS sentences. Results are shown in Table 5.4.

The FBIS tuned system has the best overall performance on baseline PRO and WRO-Local when compared against BTEC and MIXED tuned systems. However, for the FBIS tuned systems, WRO-Local has the worst performance against the baseline and against WRO-Global.

	FBIS tuned	
	Average BLEU Score	**Standard Deviation**
Baseline PRO	19.82	0.076
WRO-Global	19.92	0.049
WRO-Local	19.74	0.060
	BTEC tuned	
Baseline PRO	19.14	0.097
WRO-Global	19.25	0.109
WRO-Local	19.24	0.048
	MIXED tuned	
Baseline PRO	19.72	0.076
WRO-Global	**19.94**	0.041
WRO-Local	19.75	0.072

Table 5.4: Cross-domain test results on BTEC test set

	Average BLEU Score	**Standard Deviation**
FBIS(zh-en)	27.83	0.053
WMT13(fr-en)	29.79	0.013
BETC(zh-en)	14.73	0.085

Table 5.5: Development corpus reachability test

In BTEC tuned systems, WRO-Global and WRO-Local have similar performance and thus WRO does not benefit from the local setting. Comparing the BTEC tuned and the FBIS tuned systems, we found that the overall performance of the BTEC tuned system is much worse than that of the FBIS tuned system. Therefore, a development corpus with better similarity cannot guarantee better discriminative training quality.

In the MIXED tuned systems, the baseline PRO performance scored in between BTEC and FBIS tuned systems. However, note that both WRO-Global and WRO-Local achieve the same performance as the FBIS tuned system. The WRO algorithm shows better reliability than PRO.

To investigate the cross-domain discriminative training issue more closely, we conducted a reachability test, reporting the document level oracle BLEU score for each

development corpus. The results are given in Table 5.5. We found that the BTEC development corpus has very low reachability by our system, with an average document level oracle BLEU score of only 14.73. Combined with the results in Table 5.3, we conclude that learning from low reachability sentences is harmful to SMT discriminative training. This conclusion corroborates our findings in Chapter 4.

5.3.2 WRO with LA Selection and SIMPBLEU

This section combines SIMPBLEU, LA selection and WRO (we call it LASW). In the experiments in Section 5.3, the systems are tuned using a randomly selected corpus. In this section we will train systems with LA selected corpora, and replace the scoring function by SIMPBLEU. We refer to settings tuned with an LA selected corpus as -LA, and to settings scored with SIMPBLEU for tuning as -S. For example, PRO-LA refers to a system tuned with the PRO algorithm and LA selected corpus, while WRO-G-SLA refers to a system tuned with WRO-Global, LA selected corpus and scored by SIMPBLEU.

Table 5.6 shows the results for Chinese to English using the LASW setting. Since SIMPBLEU does not support multiple references evaluation, we evaluate the system output on each reference individually and report average scores. In this case, we have four references that can be very different from each other, hence the results for SIMPBLEU contain much higher standard deviation than those for BLEU. As shown in Table 5.6, PRO tuned with an LA selected corpus (PRO-LA) leads to better scores according to both BLEU and SIMPBLEU. This result is consistent with those in the previous chapter – tuning with the LA selected corpus results in a higher translation accuracy using both MERT and PRO algorithms. Additionally, WRO-Global and WRO-Local with the LASW setting (WRO-G-SLA and WRO-L-SLA) lead to better performance than without the LASW setting. WRO-Local with the LA selected corpus and SIMPBLEU (WRO-L-SLA) improves by 0.62 in BLEU and 1.54 in SIMPBLEU over a PRO tuned system.

Table 5.7 shows the results for French to English using the LASW setting. Again, the LA selected PRO tuned system improves performance in both BLEU and SIMPBLEU (25.74 vs 25.65 and 61.12 vs 60.77). The LASW setting with WRO cannot further improve BLEU scores over the original setting, with both Global and Local

	BLEU avg.	BLEU std.	SIMPBLEU avg.	SIMPBLEU std.
PRO	21.27	0.058	43.69	1.428
PRO-LA	21.36	0.058	44.17	1.512
WRO-Global	21.51	0.035	43.76	1.428
WRO-G-SLA	21.74	0.043	45.05	1.575
WRO-Local	21.74	0.085	44.19	1.462
WRO-L-SLA	**21.89**	0.050	**45.23**	1.571

Table 5.6: NIST08 Chinese-English LASW setting: results measured with BLEU and SIMPBLEU. Boldface figures indicate the best BLEU/SIMPBLEU score among all variants

	BLEU avg.	BLEU std.	SIMPBLEU avg.	SIMPBLEU std.
PRO	25.65	0.031	60.77	0.089
PRO-LA	25.74	0.265	61.12	0.307
WRO-Global	25.76	0.047	60.98	0.055
WRO-GSLA	25.32	0.212	61.00	0.342
WRO-Local	**25.78**	0.045	61.10	0.050
WRO-LSLA	25.43	0.093	**61.57**	0.034

Table 5.7: WMT13 French-English LASW setting: results measured with both BLEU and SIMPBLEU. Boldface figures indicate the best BLEU/SIMPBLEU score among all variants

LASW WRO achieving worse BLEU scores (0.44 and 0.35 lower). This may be because the original WRO setting is optimised for BLEU, while the LASW setting is optimised for SIMPBLEU. When measuring performance with SIMPBLEU, we confirm this assumption: both the LASW settings of WRO achieve better scores than the original setting. The best system is WRO-Local with LA selected corpus and SIMPBLEU (WRO-L-SLA), whose SIMPBLEU score increases 0.8 over standard PRO.

5.3.3 Summary

In this chapter we proposed a novel discriminative training algorithm: WRO. WRO is a ranking-based optimisation algorithm inspired by PRO. Our algorithm improves the sampling strategy used in PRO, which targets the learning of correct rankings for oracles rather than from random samples. The selected samples are weighted by the

global and local weights to indicate their importance. In WRO-local, the weights are adjusted according to each individual test sentence and a unique set of parameters is optimised for that test sentence. With our WRO-local, the limitation of linear feature combination (Problem 7 in Section 1) is minimised.

In our experiments, WRO improves translation quality when compared with PRO in both in-domain and out-of-domain test scenarios and demonstrates better reliability with different training data. In future work, the similarity measure function should not only depend on word level similarity but should also consider the relationship between words, e.g., a syntax-based similarity measurement. Another way to improve performance is to adjust the weighting method. We also experimented with combining LA and SIMPBLEU with WRO, which led to further improvements over WRO.

Finally, in our experiment only 14 features were used for decoding, while state-of-the-art research has shown that exploring thousands of features can be beneficial. Changing parameters with a small feature set has limited effect. Future work should investigate high dimensional feature spaces.

6

Conclusions

This thesis proposed improvements in SMT discriminative training in order to produce translation outputs that are more human acceptable. Our main contributions spanned over three directions: by developing new evaluation metrics, quantifying the effect of development data selection and designing better discriminative training algorithms. We introduced our approaches for three components in Chapters 3 to 5.

In Chapter 3, we proposed new evaluation metrics to replace BLEU in discriminative training: the ROSE and SIMPBLEU metrics. ROSE is a trained metric, which can be customised using two training methods – a regression-based method and a ranking-based method. Our experiments showed that except in ranking tasks, the regression-based metric is more reliable than the ranking-based one. It is also more tolerant to differences in between the training and test data. However, the ranking-based metric performed the best in evaluation tasks aimed at ranking translations.

SIMPBLEU is a heuristic approach based on BLEU which directly addresses several limitations of BLEU. We analysed the components of BLEU and designed a metric that is more easily adjustable for different training purposes. SIMPBLEU is not only more reliable but also allows for more accurate discriminative training. We found that a precision-based variant of the metric has better correlation with human judgements than a recall-based variant. Clipping is more important in lower order n-gram precision metric variants, but unnecessary for higher order n-gram precision versions of the metric. We also suggested replacing geometric mean in BLEU by arithmetic mean. Our BLEU variants with arithmetic mean achieved better performance than geometric mean for both sentence and document-level evaluation. In addition, the use of arith-

metic mean makes smoothing techniques less important for sentence-level evaluation. Regarding smoothing, we found that different smoothing values affect the metric's accuracy in different ways. Finally, we applied SIMPBLEU to discriminative training. Human evaluation shows that a model trained on SIMPBLEU results in significantly better performance than a model trained on standard BLEU.

In Chapter 4 we analysed the relationship between development corpus and translation performance and designed a novel corpus selection algorithm – LA selection, which can be used for SMT discriminative training data selection without prior knowledge of the test set. The LA selection algorithm focuses on data selection from noisy and potentially non-parallel, large scale web-crawled data. The main features used in LA selection are related to word alignment and sentence length. The algorithm aims to select parallel training sentences that have better word alignment and a reasonable number of words. In our experiments, models trained on LA-selected data led to improvements of up to 2.5 BLEU scores over models trained on randomly selected sentences. The performance achieved with LA-selected data is comparable to that obtained with datasets created manually.

Our findings also pointed out that sentence length is particularly important in discriminative training. Overly long sentences are difficult to to translate, while very short sentences will most likely result in very similar translation candidates in the N-best lists. In both cases, the candidates cannot be well discriminated by the training algorithms. Our recommendation is to avoid using sentences with length below 10 words or above 50 words for discriminative training. Additionally, we examined the effects of the size of development datasets in the training performance. The results showed that using large development sets brings only small improvements in accuracy and a modest development set of 30k-70k words is sufficient for good performance.

In Chapter 5 we described a novel SMT discriminative training algorithm – Weighted Ranking Optimisation (WRO). WRO is a ranking-based optimisation algorithm with an improved strategy to generate training instances: instead of using random samples of candidate pairs, it samples oracle candidate pairs. Therefore, it focuses on optimising system parameters in order to rank the best translation into the correct order. It also assigns weights to each training sentence to penalise training sentences which are difficult to generate by the system.

Finally, we introduced the WRO-Local algorithm, a transductive learning version of WRO. Instead of one global weight for all unseen test sentences, WRO-Local optimises weights for each input test sentence. In our experiments, WRO-Local led to improvements of 0.5 BLEU scores when compared to PRO.

6.1 Future Work

The algorithms proposed in this thesis led to improvements in translation quality in various tasks. Discriminative training is however a very broad and challenging topic. Based on our main findings, we suggest the following as main avenues for future work.

- An SMT automatic evaluation metric is important in discriminative training but currently no specific evaluation metric has been specifically designed for it. Discriminative training algorithms use evaluation metrics that were created to compare systems or measure system progress over time, but this is not the optimal solution. Designing a metric for discriminative training is thus an important future direction. For example, PRO tends to use BLEU as the evaluation metric, but BLEU was not designed for sentence-level evaluation, nor for ranking similar translations. Another aspect is that current evaluation metrics do not support dynamic programming and are not decomposable, hence they cannot be used as (part of) scoring function during decoding. The decoder needs to rely on the model score to produce N-best lists, even if references are already available during decoding. Designing metrics that can be used as (part of) a scoring function could improve the quality of training, or at least improve the quality of the candidate pool that is generated, i.e., the N-best list.

- Trained evaluation metrics are able to combine many features to achieve better evaluation quality. The main problem of this approach is the need of training data, making it difficult to apply metrics for low resource settings. Our experiments showed that it is possible, with some loss in performance, to use training data in one language to build metrics for other languages. Exploring better ways to transfer models across languages/settings is an interesting direction.

- The quality of discriminative training is directly dependent on the training data available in many aspects, such as the target reachability, source length and similarity to test data. We experimented with uniformly combining relevant features reflecting these aspects to select training samples for discriminative training. This led to improved training quality when compared to random data selection. However, we were not able to further improve the training accuracy by using more elaborate ways to combine these features to select data, in particular, by using machine learning algorithms. This is because a good understanding of the relationship between training data and the training quality is still missing. Therefore it is difficult to label instances to train models on the usefulness of data for discriminative training. Exploring this relationship and better methods to improve data selection approach are thus an interesting research directions.

References

Albercht, J.S. & Hwa, R. (2008). Regression for machine translation evaluation at the sentence level. *Machine Translation*, **22**, 1–27. 30, 44

Arun, A. & Koehn, P. (2007). Online learning methods for discriminative training of phrase based statistical machine translation. In *Proc MT Summit XI*. 51

Arun, A., Haddow, B. & Koehn, P. (2010). A unified approach to minimum risk training and decoding. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, WMT '10, 365–374, Association for Computational Linguistics, Stroudsburg, PA, USA. 21

Banerjee, S. & Lavie, A. (2005). Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. *Proceedings of the ACL-05 Workshop*. 7, 29, 36, 43, 44, 48, 50

Bazrafshan, M., Chung, T. & Gildea, D. (2012). Tuning as linear regression. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, 543–547, Association for Computational Linguistics, Stroudsburg, PA, USA. 24

Bird, S. & Loper, E. (2004). Nltk: The natural language toolkit. In *Proceedings of the ACL demonstration session*, 214–217, Barcelona. 53

Blatz, J., Fitzgerald, E., Foster, G., Gandrabur, S., Goutte, C., Kulesza, A., Sanchis, A. & Ueffing, N. (2004). Confidence estimation for machine translation. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, Association for Computational Linguistics, Stroudsburg, PA, USA. 30, 38

Blunsom, P., Cohn, T. & Osborne, M. (2008). A discriminative latent variable model for statistical machine translation. Columbus, Ohio, USA, 200–208. 6, 26

Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Leveling, J., Monz, C., Pecina, P., Post, M., Saint-Amand, H., Soricut, R., Specia, L. & Tamchyna, A. (2014). Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, 12–58, Association for Computational Linguistics, Baltimore, Maryland, USA. 14, 51

Brown, P.F., Della-Pietra, S.A., Della-Pietra, V.J. & Mercer, R.L. (1993). The mathematics of statistical machine translation : Parameter estimation. *Computational Linguistics*. 1, 2, 11, 13

Callison-Burch, C., Osborne, M. & Koehn, P. (2006). Re-evaluating the role

of bleu in machine translation research. In *In EACL*, 249–256. 44, 50

Callison-Burch, C., Fordyce, C., Koehn, P., Monz, C. & Schroeder, J. (2008). Further meta-evaluation of machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, 70–106, Association for Computational Linguistics, Columbus, Ohio. 53, 80

Callison-Burch, C., Koehn, P., Monz, C. & Schroeder, J. (2009). Findings of the 2009 Workshop on Statistical Machine Translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, 1–28, Association for Computational Linguistics, Athens, Greece. 51, 57, 80

Callison-Burch, C., Koehn, P., Monz, C., Peterson, K., Przybocki, M. & Zaidan, O. (2010). Findings of the 2010 joint workshop on statistical machine translation and metrics for machine translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, 17–53, Association for Computational Linguistics, Uppsala, Sweden, revised August 2010. 51, 52, 61, 80

Callison-Burch, C., Koehn, P., Monz, C. & Zaidan, O. (2011). Findings of the 2011 workshop on statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, 22–64, Association for Computational Linguistics, Edinburgh, Scotland. 29, 43, 49, 51, 61

Callison-Burch, C., Koehn, P., Monz, C., Post, M., Soricut, R. & Specia, L. (2012). Findings of the 2012 workshop on statistical machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, 10–51, Association for Computational Linguistics, Montréal, Canada. 8, 51

Cao, Y. & Khudanpur, S. (2012). Sample selection for large-scale mt discriminative training. In *AMTA*. 10, 40, 41, 74

Chan, Y.S. & Ng, H.T. (2008). Maxsim: A maximum similarity metric for machine translation evaluation. In *Proceedings of ACL*. 36

Chiang, D. (2007). Hierarchical phrase-based translation. *Computational Linguistics*, **33**, 201–228. 14

Chiang, D. (2012). Hope and fear for discriminative training of statistical translation models. *J. Mach. Learn. Res.*, **98888**, 1159–1187. 4, 6

Chiang, D., DeNeefe, S., Chan, Y.S. & Ng, H.T. (2008a). Decomposability of translation metrics for improved evaluation and efficient algorithms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, 610–619, Association for Computational Linguistics, Stroudsburg, PA, USA. 44, 50

Chiang, D., Marton, Y. & Resnik, P. (2008b). Online large-margin training of syntactic and structural translation features. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, 224–233, Association for Computational Linguistics, Stroudsburg, PA, USA. 25, 26, 49, 51

Chiang, D., Knight, K. & Wang, W. (2009). 11,001 new features for statistical machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, 218–226, Association for Computational Linguistics, Stroudsburg, PA, USA. 3

Clark, J.H., Dyer, C., Lavie, A. & Smith, N.A. (2011). Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, 176–181, Association for Computational Linguistics. 81

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, **20:37**. 62

Corston-Oliver, S., Gamon, M. & Brockett, C. (2001). A machine learning approach to the automatic evaluation of machine translation. In *proceedings of the Association for Computational Linguistics*. 30, 38, 44

Daume, H. (2004). Notes on cg and lm-bfgs optimization of logistic regression. *Unpublished*. 95

Doddington, G. (2002). Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, HLT '02, 138–145, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 50

Dreyer, M. & Marcu, D. (2012). Hyter: Meaning-equivalent semantics for translation evaluation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, 162–171, Association for Computational Linguistics. 5, 6

Duh, K. (2008). Ranking vs. regression in machine translation evaluation. In *In Proceedings of the Third Workshop on Statistical Machine Translation*, 191–194, Columbus,Ohio,. 44

Eck, M., Vogel, S. & Waibel, A. (2005). Low cost portability for statistical machine translation based on n-gram frequency and tf-idf. In *Proceedings of IWSLT*. 40

Galley, M., Quirk, C., Cherry, C. & Toutanova, K. (2013). Regularized minimum error rate training. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1948–1959, Association for Computational Linguistics, Seattle, Washington, USA. 5

Gimpel, K. & Smith, N.A. (2012). Structured ramp loss minimization for machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, 221–231, Association for Computational Linguistics, Stroudsburg, PA, USA. 21, 26

Hanneman, G., Huber, E., Agarwal, A., Ambati, V., Parlikar, A., Peterson,

E. & Lavie, A. (2008). Statistical transfer systems for french–english and german–english machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, StatMT '08, 163–166, Association for Computational Linguistics, Stroudsburg, PA, USA. 50

Herbrich, R., Minka, T. & Graepel, T. (2007). Trueskill(tm): A bayesian skill rating system. In *Advances in Neural Information Processing Systems 20*, 569–576, MIT Press. 66

Hildebrand, A.S., Eck, M., Vogel, S. & Waibel., A. (2005). Adaptation of the translation model for statistical machine translation based on information retrieval. In *In Proceedings of the 10th Conference of the European Association for Machine Translation (EAMT)*, pages 133–142. 39

Hopkins, M. & May, J. (2011). Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, 1352–1362, Association for Computational Linguistics, Edinburgh, Scotland, UK. 4, 7, 9, 10, 23, 24, 49, 91

Hui, C., Zhao, H., Song, Y. & Lu, B.L. (2010). An empirical study on development set selection strategy for machine translation learning. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, WMT '10, 67–71, Association for Computational Linguistics, Stroudsburg, PA, USA. 73, 74, 88

Joachims, T. (1999). Making large-scale svm learning practical. *Advances in Kernel Methods - Support Vector Learning,*. 45, 49

Koehn, P. (2004a). Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In *AMTA*. 15

Koehn, P. (2004b). Statistical significance tests for machine translation evaluation. In *Proceedings of 2004 EMNLP*. 63

Koehn, P., Och, F.J. & Marcu, D. (2003). Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, 48–54. 13

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A. & Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*. 43, 61, 73

Koehn, P., Arun, A. & Hoang, H. (2008). Towards better machine translation quality for the german–english language pairs. In *Proceedings of the Third Workshop on Statistical Machine Translation*, StatMT '08, 139–142, Association for Computational Linguistics, Stroudsburg, PA, USA. 50

Kumar, S., Macherey, W., Dyer, C. & Och, F. (2009). Efficient minimum error rate training and minimum bayes-risk decoding for translation hypergraphs and lattices. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of*

the AFNLP: Volume 1 - Volume 1, ACL '09, 163–171, Association for Computational Linguistics, Stroudsburg, PA, USA. 21

Li, M., Zhao, Y., Zhang, D. & Zhou, M. (2010). Adaptive development data selection for log-linear model in statistical machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, 662–670, Association for Computational Linguistics, Stroudsburg, PA, USA. 10, 40, 74

Li, Z. & Eisner, J. (2009). First- and second-order expectation semirings with applications to minimum-risk training on translation forests. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, 40–51, Association for Computational Linguistics, Stroudsburg, PA, USA. 21

Li, Z., Callison-Burch, C., Dyer, C., Ganitkevitch, J., Khudanpur, S., Schwartz, L., Thornton, W.N.G., Weese, J. & Zaidan, O.F. (2009). Joshua: an open source toolkit for parsing-based machine translation. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, StatMT '09, 135–139, Association for Computational Linguistics, Stroudsburg, PA, USA. 43

Liang, P., Bouchard-Côté, A., Klein, D. & Taskar, B. (2006). An end-to-end discriminative approach to machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, 761–768, Association for Computational Linguistics, Stroudsburg, PA, USA. 7, 21, 22, 25, 27, 49, 51

Lin, C.Y. & Och, F.J. (2004). Orange: a method for evaluating automatic evaluation metrics for machine translation. In *Proceedings of the 20th international conference on Computational Linguistics*, COLING '04, Association for Computational Linguistics, Stroudsburg, PA, USA. 35, 50

Liu, C., Dahlmeier, D. & Ng, H.T. (2010). Tesla: Translation evaluation of sentences with linear-programming-based analysis. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, 354–359, Association for Computational Linguistics, Uppsala, Sweden. 29, 36, 43, 50

Liu, D. & Gildea, D. (2005). Syntactic features for evaluation of machine translation. 30, 38

Liu, L., Cao, H., Watanabe, T., Zhao, T., Yu, M. & Zhu, C. (2012). Locally training the log-linear model for smt. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, 402–411, Association for Computational Linguistics, Stroudsburg, PA, USA. 40, 74, 95, 96

Lopez, A. (2009). Translation as weighted deduction. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, 532–540, Association for Computational Linguistics, Athens, Greece. 16

REFERENCES

Lu, Y., Huang, J. & Liu, Q. (2008). Improving statistical machine translation performance by training data selection and optimization. 10, 39, 74

Macháček, M. & Bojar, O. (2013). Results of the WMT13 metrics shared task. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, 45–51, Association for Computational Linguistics, Sofia, Bulgaria. 8, 51

Marcu, D. & Wong, W. (2002). A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, 133–139. 13

Munteanu, D.S. & Marcu, D. (2005). Improving machine translation performance by exploiting non-parallel corpora. *Comput. Linguist.*, **31**, 477–504. 75, 77, 78, 79, 87

Och, F.J. (2003). Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, 160–167, Association for Computational Linguistics, Stroudsburg, PA, USA. 4, 5, 17, 19, 51, 80, 91

Och, F.J. & 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*, ACL '02, 295–302, Association for Computational Linguistics, Stroudsburg, PA, USA. 2, 17, 18, 19, 91

Owczarzak, K., Groves, D., Van Genabith, J. & Way, A. (2006). Contextual bitext-derived paraphrases in automatic mt evaluation. In *Proceedings of the Workshop on Statistical Machine Translation*, StatMT 06, 86–93, Association for Computational Linguistics, Stroudsburg, PA, USA. 50

Papineni, K., Roukos, S., Ward, T. & Zhu, W.J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, 311–318, Association for Computational Linguistics, Stroudsburg, PA, USA. 7, 29, 33

Pecina, P., Toral, A. & van Genabith, J. (2012). Simple and effective parameter tuning for domain adaptation of statistical machine translation. In *Proceedings of COLING 2012*, 2209–2224, The COLING 2012 Organizing Committee, Mumbai, India. 74

Popović, M. & Ney, H. (2009). Syntax-oriented evaluation measures for machine translation output. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, 29–32, Association for Computational Linguistics, Athens, Greece. 36

Quirk, C. (2004). Training a sentence-level machine translation confidence measure. In *In: Proceedings of the international conference on language resources and evaluation*, 825–828, Lisbon, Portugal. 30

Resnik, P. & Smith, N.A. (2003). The web as a parallel corpus. *Comput. Linguist.*, **29**, 349–380. 75

Sakaguchi, K., Post, M. & Van Durme, B. (2014). Efficient elicitation of annotations for human evaluation of machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, 1–11, Association for Computational Linguistics. 65

Smith, D.A. & Eisner, J. (2006). Minimum risk annealing for training log-linear models. In *Proceedings of the COLING/ACL on Main Conference Poster Sessions*, COLING-ACL '06, 787–794, Association for Computational Linguistics, Stroudsburg, PA, USA. 21

Smith, J., Koehn, P., Saint-Amand, H., Callison-Burch, C., Plamada, M. & Lopez, A. (2013a). Dirt cheap web-scale parallel text from the common crawl. In *The 51st Annual Meeting of the Association for Computational Linguistics*. 5, 75

Smith, J.R., Quirk, C. & Toutanova, K. (2010). Extracting parallel sentences from comparable corpora using document level alignment. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, 403–411, Association for Computational Linguistics, Stroudsburg, PA, USA. 75

Smith, J.R., Koehn, P., Saint-Amand, H., Callison-Burch, C., Plamada, M. & Lopez, A. (2013b). Dirt cheap web-scale parallel text from the common crawl. In *Proceedings of the 2013 Conference of the Association for Computational Linguistics (ACL 2013)*. 80

Smola, A.J. & Scholkopf, B. (2004). A tutorial on support vector regression. *STATISTICS AND COMPUTING*, **14**, 199–222. 49

Snover, M., Dorr, B., Schwartz, R., Micciulla, L. & Weischedel, R. (2006). A study of translation error rate with targeted human annotation. 7, 30, 32, 43

Song, X. & Cohn, T. (2011). Regression and ranking based optimisation for sentence level mt evaluation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, 123–129, Association for Computational Linguistics, Edinburgh, Scotland. 8, 50

Song, X., Cohn, T. & Specia, L. (2013). Bleu deconstructed: Designing a better mt evaluation metric. In *CICLING*. 8

Song, X., Specia, L. & Cohn, T. (2014). Data selection for discriminative training in statistical machine translation. In *The Seventeenth Annual Conference of the European Association for Machine Translation*. 9

Specia, L. & Gimenez, J. (2010). Combining confidence estimation and reference-based metrics for segment-level mt evaluation. In *The Ninth Conference of the Association for Machine Translation in the Americas*, Denver,Colorado. 30, 38, 44

Specia, L., Turchi, M., Cancedda, N., Dymetman, M. & Cristianini, N. (2009). Estimating the sentence-level quality of machine translation systems. 30, 38, 44

REFERENCES

Tamchyna, A., Galuščáková, P., Kamran, A., Stanojević, M. & Bojar, O. (2012). Selecting data for english-to-czech machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, WMT '12, 374–381, Association for Computational Linguistics, Stroudsburg, PA, USA. 10, 74

Tillmann, C. & Zhang, T. (2006). A discriminative global training algorithm for statistical mt. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, 721–728, Association for Computational Linguistics, Stroudsburg, PA, USA. 21, 51, 91

Tillmann, C., Vogel, S., Ney, H., Zubiaga, A. & Sawaf, H. (1997). Accelerated dp based search for statistical translation. In *In European Conf. on Speech Communication and Technology*, 2667–2670. 30, 31

Uszkoreit, J., Ponte, J.M., Popat, A.C. & Dubiner, M. (2010). Large scale parallel document mining for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, 1101–1109, Association for Computational Linguistics, Stroudsburg, PA, USA. 75

Watanabe, T., Suzuki, J., Tsukada, H. & Isozaki, H. (2007). Online large-margin training for statistical machine translation. In *In Proc. of EMNLP-CoNLL. 764–773*. 4, 7, 22, 25, 91

Yamada, K. & Knight, K. (2002). A decoder for syntax-based statistical mt. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, 303–310. 14

Yu, H., Huang, L., Mi, H. & Zhao, K. (2013). Max-violation perceptron and forced decoding for scalable mt training. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. 21, 27

Zaidan, O. (2011). Maise: A flexible, configurable, extensible open source package for mass ai system evaluation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, 130–134, Association for Computational Linguistics, Edinburgh, Scotland. 62

Zaidan, O.F. & Callison-Burch, C. (2011). Crowdsourcing translation: professional quality from non-professionals. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, 1220–1229, Association for Computational Linguistics, Stroudsburg, PA, USA. 75

Zens, R., Hasan, S. & Ney, H. (2007). A systematic comparison of training criteria for statistical machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 524–532, Association for Computational Linguistics, Prague, Czech Republic. 21

Zhang, Y., Vogel, S. & Waibel, A. (2004). Interpreting bleu/nist scores: How much improvement do we need to have a

better system. In *In Proceedings of Proceedings of Language Resources and Evaluation (LREC-2004*, 2051–2054. 35, 50, 51

Zhao, Y., Ji, Y., Xi, N., Huang, S. & Chen, J. (2011). Language model weight adaptation based on cross-entropy for statistical machine translation. In *Proceedings of the 25th Pacific Asia Conference on Language, Information and Computation*, 20–30, Institute of Digital Enhancement of Cognitive Processing, Waseda University, Singapore. 74

Zheng, Z., He, Z., Meng, Y. & Yu, H. (2010). Domain adaptation for statistical machine translation in development corpus selection. In *Universal Communication Symposium (IUCS), 2010 4th International*. 10, 74