Using Variable Decoding Weight for Language Model in Statistical Machine Translation

Behrang Mohit[†,*] **Rebecca Hwa**[†,‡]
Intelligent Systems Program[†], Computer Science Department[‡]
University of Pittsburgh
Pittsburgh, PA 15260
{behrang,hwa}@cs.pitt.edu

Alon Lavie
Language Technology Institute
Carnegie Mellon University
Pittsburgh, PA 15213
alavie@cs.cmu.edu

Abstract

This paper investigates varying the decoder weight of the language model (LM) when translating different parts of a sentence. We determine the condition under which the LM weight should be adapted. We find that a better translation can be achieved by varying the LM weight when decoding the most problematic spot in a sentence, which we refer to as a *difficult segment*. Two adaptation strategies are proposed and compared through experiments. We find that adapting a different LM weight for every difficult segment resulted in the largest improvement in translation quality.

1 Introduction

The dominant paradigm in Statistical Machine Translation (SMT) is a log-linear model. It combines multiple information sources by treating each underlying component, such as the language model or the distortion model, as a feature. The strength of each component's influence on the translation process is determined by the decoding weight associated with it. Typically, these weights are tuned during training; once optimized on a development set, they remain fixed for all future inputs. Thus, if the language model were deemed to be generally helpful, the decoder must always weigh its scores highly, even when considering some portion of text for which the language model is known to have poor coverage.

This paper investigates varying the decoder weight of the Language Model (LM) while translating a sentence. We address two challenges. The first is to define under what conditions would the language model warrant a different decoder weight setting. The second is to develop methods of assigning the appropriate weight values.

We argue that the right scope for a language model weight to excise its influence is at the subsentential level. Even a long and complex sentence may contain parts that are relatively straightforward. Our approach is to first identify the portion of a source sentence whose translation may be problematic for the target language model (we refer to this special portion as a *segment*); then use an adapted LM weight for the translation of the special segment but use the default LM weight for the rest of the sentence.

We previously defined such special segment as *Difficult to Translate Phrase* (DTP) (Mohit and Hwa, 2007). We also described an SVM classifier that can predict whether a segment of source text will be problematic for the MT system with good accuracy.

In an extended work, we adapted the language model of a SMT system for the translation of the DTPs (Mohit et al., 2009). For each DTP we automatically selected the relevant subset of the training data and constructed an adapted language model. Moreover, we translated the difficult phrase with the adapted model. In this paper, we continue our segment-specific system customization framework for testing our idea of modifying the language model weights.

To find a better LM weight for the special segments, we consider two options. One is to treat all the special segments as a group and learn an appropriate weight for the group; another is to predict a weight value for each segment. The first option can

[*]Currently at Carnegie Mellon University in Qatar, P.O.Box 24866, Doha, Qatar, behrang@cmu.edu

be performed in a straight-forward manner similar to how the default weight is learned; we estimate the weight on a development set of problematic segments; we then use the learned LM weight for translating problematic segments in the test set. The second option is a more challenging learning problem because we have to train a function that can assign an appropriate weight to every segment. While this problem maps most directly to regression learning, we have found a greater success by formulating it in terms of preference ranking over a set of candidate weights.

We have conducted experiments to compare different methods of determining the adapted LM weight. For a robust evaluation, we train the Arabic to English PB-SMT system with a small (1M words) and later a medium (50M words) size corpora. We observed an improvement of 1.16 BLEU score over baseline when we use a classifier to identify the most problematic source segment and then use ranking to determine the LM weight for the highlighted segment. Our studies suggest that promising improvements in the translation quality can be achieved by judiciously varying the LM weight.

2 Overview

SMT decoders use a set of features to compute a decoding score for each translation hypothesis. These features are collected from different resources such as the translation model, the language model, linguistic properties of the source-language text, etc. The influence of each feature is decided by its associated weight value; they are combined to determine the decoding score. For example, in the Table 1 which presents a standard Phrase-Based SMT (PB-SMT) formulation, the (λ)s are the decoding weights for the Language Model (LM), Translation Model (TM), Word Penalty (WP) and Distortion (d) features. The translation model feature function(ϕ) holds four model parameters, and each parameter gets an entry in the λ_ϕ weight vector.

The weights are estimated using the Minimum Error Rate Training (MERT) (Och, 2003). MERT tunes the SMT system based on an iterative translation task performed on a development set. In each iteration, the MERT estimates a new set of decoding weights. It then check the effects of the new weights on the translation quality of the development data, using automatic evaluation metrics such as BLEU (Papineni et al., 2002). This evaluation is usually performed at the corpus level. After the training converges, the decoder computes scores for all translation hypotheses using the tuned weights, regardless of their characteristics.

The goal of our work is to improve translation by using different decoder weights for different parts of a source sentence. Specifically, we propose to adapt the language model weight for decoding the problematic *segment* of a source sentence. This is a sequence of five to fifteen words whose characteristics are significantly different from the average case so as to cause problems for the translation process. We limit the scope of our study to modifying only the language model (LM) weight to gain a better understanding of the effects of varying decoder weights. This is because LM is less interdependent than other features, yet its influence on the decoding is stronger than features like word penalty and phrase distortion.

2.1 Difficult Segments

The first step of our work on LM weight adaptation is to characterize the type of source segments whose translations might improve if the decoder were to use a different LM weight value. Ideally, we would like to correlate each source segment to the performance of the language model; that is, we would like to identify source segments whose translations cannot be scored accurately by the language model. However, this correlation cannot be made directly. Because the language model is applied on the translation candidates in the target language, its performance is somewhat dependent on other components of the translation process. As a proxy to the direct correlation, we look for source segments that are expected to receive poor translations from the underlying PB-SMT system. In our earlier works we referred to these segments as Difficult-to-Translate Phrases (DTPs) (Mohit and Hwa, 2007). Here we refer to them as difficult *segments* to avoid potential confusion with the phrases in a PB-SMT translation table. Difficult segments are sub-sentences with five to fifteen source language words with no syntactic constraint. Here we follow the same classification framework to locate difficult segments. We identify

$$e_{best} = argmax_e \prod_{i=1}^{n} \phi(\bar{f}_i|\bar{e}_i)^{\lambda_\phi} d(start_i - end_{i-1} - 1)^{\lambda_d} \prod_{i=1}^{|e|} p_{LM}(e_i|e_1...e_{i-1})^{\lambda_{LM}} |e|^{\lambda_{WP}}$$

Table 1: Decoding weights in PB-SMT's formulation

some common features of these segments and build a classifier to predict whether a segment would be difficult for the MT. Although our original model of potential causes of translation difficulties includes factors other than language modeling errors, but the LM features are influential enough for finding segments with LM-related problems. In most experiments in this work (except section 6), the most difficult segment is gold-standard. That means that for each sentence, the reference translations are used to find the segment with the lowest translation quality.

2.2 Implementation

We modify the standard PB-SMT pipeline to facilitate LM weight adaptation. Our basic PB-SMT system uses the SRI package (Stolcke, 2002) as the target language model and Phramer (Olteanu et al., 2006), an open source implementation similar to Pharaoh (Koehn, 2004), as the decoder. We modify the decoder to allow the usage of different LM weights for different parts of a sentence; it accepts two decoding weights: One is used for the difficult segment, and the other one is used for the translation of the rest of the sentence. In order to apply this separation of the decoding weights, we constrain the choice of phrases in hypothesis expansion at the boundaries of the difficult segment. We allow the decoder to shift the difficult segment's boundaries with one word to use more of the phrase table entries.

We assume that every sentence holds at least one segment that is more difficult than other segments. To reduce the complications of decoding complication, we limit our decoder modification and consequently our weight adaptation to the most difficult segment of each sentence. To apply the adapted LM weight to the most difficult segment of each sentence, we first need to locate the difficult segments on each test sentence. Figure 1 describes the data flow. We have implemented a control component to interact with the difficulty classifier. We pass different segments of the test sentence to the classifier; the segment that received a difficult label with the highest confidence is selected as the problematic segment to receive weight adaptation. To determine whether the adaptation is helpful, we apply standard MT evaluation on the test data using BLEU (Papineni et al., 2002).

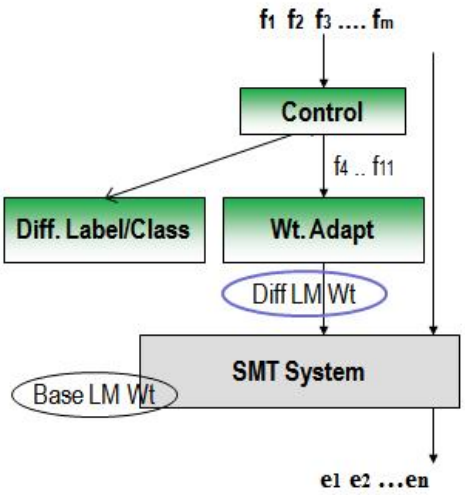

Figure 1: The SMT pipeline with difficulty classifier and LM weight adaptation

2.3 Data

In all our experiments, the underlying PB-SMT system translates from Arabic to English. For training the MT system, we use a one million words Arabic-English parallel corpus released by the Linguistic Data Consortium (LDC)[1]. The corpus can be obtained from the Linguistic Data Consortium under catalog ID LDC2004T17, LDC2004T18. To train a trigram target language model, we use

[1]In section 6, we cumulatively expand the training data to 50 million words.

the English side of the parallel corpus. For the exploratory experiments on weight adaptation for difficult segments, we use the NIST-2003 multi-translation Arabic-English corpus (LDC2003T18). This corpus offers 1037 sentences with multiple reference translation. Using GIZA++ word alignments, we extract 3360 parallel segments from this corpus. Since each sentence includes multiple translations, we obtain the word alignments between the source sentences and each of the reference translations and later use the source segment to match its multiple translations. The extracted segments can be evaluated against the references and tagged with gold-standard labels of "easy" or "difficult." Most of this dataset is used to train the classifier for predicting difficult segments. A subset of 655 segments are used as training, development and test data for the weight adaptation experiments. For each of these segments, the translation experiments are conducted at the sentence level. Furthermore, we evaluate the translation quality both at the segment and sentence levels.

3 LM Weight For All Difficult Segments

We begin by investigating whether the decoder can take advantage of using two LM weights. One is the baseline set of decoding weights. These weights come from running MERT on a development set of sentences. We also train a new LM weight that will be used for all the difficult segments. Starting with the baseline weight set, we run a modified version of the MERT in which only the LM weight is allowed to change while the other decoding features remain fixed. This re-tuning is performed over a development set of 100 difficult segments. For this part of our study, instead of using the difficulty classifier in the manner described in Figure 1, we use the gold-standard tagging of difficult segments for both the development and the test sentences.

Table 2 compares the effect of weight tuning on the chosen difficult segments. We see that the LM weight is decreased from the baseline value of 0.37 to the optimized value of 0.28. The new optimized weight results in an increment of about 3 BLEU score for the development set. A smaller (1.5 score) improvement holds when we use the learned LM weight for the unseen test set of 301 difficult segments. This suggest that the influence of the baseline LM should be decreased for the difficult segments' translations.

	Wt. Val	**Dev**	**Test**
Baseline	0.37	19.93	15.96
Diff. Seg.	0.28	22.81	17.44

Table 2: Comparison of the usage of baseline and difficult-segment specific LM weights (BLEU evaluation at the *segment* level)

We next consider the effect of a segment level weight adaptation over the entire sentence. Table 3 compares three cases: translating the entire test set using the default weights, replacing the LM weight with the weight adapted specifically for the difficult segments for the entire sentence, and using a combination of the adapted weight on the difficult segments and the default weight on the rest of the sentence. As expected, there is little change in performance when applying the adapted weight indiscriminately over the entire test set. On the other hand, when we tailor the LM weight for only the difficult segment, there is a small gain in the overall BLEU score. These results indicate that i) "translation difficulty" is a useful proxy for identifying problematic spots for the language model; and ii) adapting weights at a sub-sentential level is a promising strategy.

	Wt. Val	**Test**
Baseline	0.37	22.56
Diff-Specific	0.28	22.69
Combined	0.28/0.37	23.17

Table 3: Comparison of the usage of baseline and difficult segment-specific LM weights (BLEU evaluation at the *sentence* level)

4 An Analysis of LM Weight Changes

The results of the first study suggest that difficult segments as a group require less influence from the language model. In order to better understand the interaction between the LM decoder weight and each individual segment, we conduct an oracle experiment to find the best LM weight for each segment.

This allows us to perform further analysis to determine whether it is worthwhile to adapt LM weights for each individual segment.

Similar to the previous study, we evaluate every segment of each sentence against its reference translation. We then extract a set of the most difficult segments and a set of the "easiest" segments (segments whose translations received the highest BLEU scores). In this study, we compare 301 difficult segments with 253 easy segments. To find the best LM weight for each segment, we perform a brute-forced search, trying values near the default weight in discrete (+/-0.01) steps of 20 increments and 20 decrements. The weight value that results in translation with the highest BLEU score is considered to be the gold standard weight for that segment[2]. In case of ties, we choose the weight that is the closest to the baseline weight. For each segment set, we record the percentage of the gold standard weight changes with respect to the baseline weight. We also record the oracle BLEU score for each set to give us an estimate of the upper-bound for LM weight learning.

Group	**Wt. Change** (-/=/+)	**Baseline** BLEU	**Oracle** BLEU
Diff. Segs	36/49/15	15.96	21.93
Easy Segs	11/72/16	46.09	47.23

Table 4: Comparison of the changes in the Oracle Weights and their effects on translation (segment-level BLEU score)

Table 4 presents our findings. We see that for many segments there is no weight change; for 49% of the difficult segments and 72% of the easy segments, the oracle simply prefers the baseline weight. However, we observe that there is a definite characteristic difference between the two segment groups. The oracle prefers to reduce the LM weights for 36% of the difficult segments. In contrast, the change distribution for the easy segments is almost even and there less tendency towards changing the weight at all. This suggests that the baseline weight usually provides the best possible translation for the easy segments. These results are also reflected in the BLEU score changes. There is a larger score improvement over the baseline when we use the oracle weights on the difficult segments, than the easy segments.

To determine whether the difficult segments share some common problems that are ameliorated by LM weight reduction, we manually examined 70 segments to analyze the effects of weight changing. We observed that the majority of translation improvements are related to under-generation. A common problem is the short length of the difficult segments. The decoder is hesitant to generate function words such as articles, auxiliary verbs and punctuation. This problem is more frequent in the proximity of words for which the translation model is sparse.

In the case of complex verbs such as the past participle or the passive form, problems such as morphology and word alignment errors cause sparseness in the phrase table. Therefore, it is upon the language model to generate the auxiliary verbs. Reduction of the LM weight, reduces the cost of target-language generation in the decoding. This eases the generation of longer hypothesis with more function words, specially the auxiliary verbs. Table 5 gives an example in which the under-generation problem is reduced by using a smaller LM weight.

The under-generation problem also occurs when the language model sparseness forces the decoder to choose incorrect, yet shorter (i.e. few words) translation of a source-language term. Similarly, reducing the LM weight relieves the decoder to choose more expensive, yet more accurate paths.

There are fewer oracle weight changes for the easy segments, and we do not observe any dominant pattern in the improved translations. Most improvements are due to punctuations; there are some infrequent cases of over-generation, in which the decoder generates extra content words. This problem is reduced when we increase the LM-weight which shortens the generation length. Table 6 shows an example of the over-generation problem. In the example, the word *joint* is not needed. With a higher LM weight, the extra word is omitted. The over-generation problem does not seem to occur as frequently among difficult segments.

[2]Due to the smaller scope of a segment, we use BLEU-3 (with BLEU-2 as back-off) for these scoring.

Ref: richard nixon had visited syria , which is still ...
Baseline: richard nyskwn visited syria , which still ...
New LM-Wt: richard nyskwn *had* visited syria , which *is* still ...

Table 5: An example of under-generation in a difficult segment

Ref: kharazi said in a press conference with ...
Baseline: kharazi said in a *joint* press conference he held with ...
New LM-Wt: kharazi said in a press conference he held with ...

Table 6: An example of over-generation

5 Learning Individual LM Weight

From the oracle experiment, we observed that the best LM weight may be different for each difficult segment whereas the variations of LM weights is smaller for the easy segments. Therefore, we now investigate learning to predict the best LM weight for each difficult segment. By finding oracle LM weights of difficult segments, we construct training data for supervised learning methods.

The most direct formulation of weight adaptation learning is as a regression problem: Learn a function that takes a source segment as input and returns the best LM weight value for it. We believe such a direct approach is unlikely to be successful for several reasons. First, we have a limited number of training instances. Second, the question of "what is the best LM weight" is not a very intuitive question, which makes feature engineering challenging. We might need to explore a more complex feature space than resources allow. Third, using different LM weights may nonetheless produce the same translation. Since our ultimate goal is to improve the translation quality, it is less important for us to predict the exact LM weight specified by the gold standard than to predict some LM weight that helps the decoder to produce a better translation. Therefore, instead of direct regression, we map the weight adaptation problem to one of ranking relative translation qualities. For each source segment, we construct a set of translation candidates generated from decoding with different LM weights. We train a ranker to prefer the candidate that appears to have the best translation. The LM weight associated with the top ranking candidate is considered to be the new adapted value. There are many ways to develop a ranking model. In this work, we use the ranking SVM algorithm, which converts the ranking problem to a series of pairwise preference classifications. The details of the conversion is described in the work of (Cao et al., 2006), who have used this process for ranking in information retrieval.

5.1 The Ranking Model

A training instance of the ranking model is a set of translation candidates and their relative rankings. The ranking is determined according to their actual translation quality (measured by BLEU-3 against a human reference). Duplicate translations are removed so that each ranked list specifies a total order. The learning task is to predict the gold standard ranking based on the features that can be extracted from the translation candidates. In other words, the ranker's objective can be seen as performing a kind of automatic MT evaluation metric that *does not* require references on the translation candidates. Based on related work in MT evaluation (Specia et al., 2009; Albrecht and Hwa, 2007), we use a similar set of features such as:

- N-gram matching with the underlying LM corpus
- N-gram probabilities from the LM
- Target to source-language lexical ambiguity
- Average word movement from source to target-language
- The ratio of punctuation and digit in the source and target phrases

- The average BLEU score: A BLEU evaluation of the hypothesis, using the other competing hypothesis as the pseudo references

- Bigram and trigram POS tags: In order to find patterns of compound nouns and verbs, we use a few POS tag patterns.

At test time, we generate 40 translation candidates for an unseen difficult segment using a window of different LM weights around the MT system's default weight value (eg. +/- 0.01, 0.02, ...). We represent each decoding as a feature vector. The trained ranker takes the list of feature vectors and ranks them. We pick the weight value that produced the top-ranking candidate as the best decoding weight.

5.2 Experiment

To determine whether performing individual LM weight adaptation over segments might improve translation, we conduct a controlled experiment using a set of 500 sentences. We obtain the most difficult segment for each sentence as before. A set of 40 translation candidates is then generated for each segment using different LM weights, and their gold-standard ranking order is established as described above. We keep 50 sentences for tuning the kernel parameters of the SVM; the remaining 450 sentences are split into three-folds for cross validation (300 train, 150 test). We use an SVM-light implementation of the SVM-Rank algorithm (Joachims, 1998) with a polynomial kernel.

Table 7 compares the translation results under different adaptation strategies. As in earlier experiments, we report both the overall BLEU score for the difficult segments as a group as well as that of the entire test set. The first two rows reinforce observations from Section 3 that using a shared weight value adapted for difficult segment can improve translations. The third row compares the individual weight adaptation strategy with the shared weight method. In this case, we observe a larger improvement over the BLEU score of the test set. Additional analysis on this data set shows that for about 40% of the segments, the chosen translation is the same as the baseline. The ranking model chooses the gold standard weight as the top-ranking candidate for 19% of the segments. For 68% of the segments, the ranking model's top choice resulted in translation improvements over the baseline.

LM weight used	Seg. Eval	Sent. Eval
Baseline	15.02	23.04
adapt. by group	16.06	23.30
adapt. of indiv. wt.	17.30	24.06

Table 7: Comparison of the two weight adaptation experiments (BLEU evaluation)

5.3 Discussion

While our objective in performing rank prediction is to enable LM weight adaptation, the process is similar to other work on ranking for SMT (Och et al., 2004). However, there are several aspects that make our approach different:

- Ranking models rely on additional features. To directly perform ranking inside the decoder would require considering more features during decoding. This expands the search space and slows down the decoding efficiency. Our approach limits its focus on one component (the LM) over a smaller scope (a segment) so the search space is more restricted.

- Another way to incorporate a richer feature space is to perform post-decoding reranking (Och et al., 2004). In that case, the candidate list is made up of the n-best hypotheses of the baseline system. In contrast, our candidate list generates alternative translations using different LM decoder weights. This results in a different population of candidates to choose from. We observed that 36% of the translations produced by our oracle weight cannot be found in a 100-best list generated by the baseline MT system.

Our adaptation of the decoding weights is efficient when the components of the SMT systems are static. The individual weight learning framework would not converge if different adapted models are used for each sentence or segment. However, our group weight learning method (section 3) can be used to find an alternative weight when one adapts the language model for individual difficult segments

(Mohit et al., 2009). Moreover, a group weight for all adapted language model can be tuned using a development set of difficult segments (and their associated adapted language models).

In principle the proposed ranking approach is flexible, and can be extended to adapt for more than one weight. The major challenge of the extension is modeling the ranking of a multi-dimensional vector. Dimension reduction frameworks such as Principle Component Analysis might be used to transform the problem to smaller dimensions. Moving to multi-weight learning may also demand a more complex oracle for producing the gold standard multi-dimensional data.

6 A Start-to-Finish Experiment

We test our weight adaptation within a complete SMT pipeline: finding the most difficult segment of a sentence and then predicting the LM weight for it. The data flow is described in Figure 1. For each sentence translation, we use the following procedure:

1. For each source sentence, consider every overlapping segments of five to fifteen words.
   - Translate each segment with the baseline system
   - Label the translation with the difficulty classifier.
   - Compare confidence scores of all translations to identify the most difficult segment.
2. Construct a set of translation candidates for the difficult segment using different LM weights.
3. Rank the set of translation candidates to find the best LM weight.
4. Complete translating the entire sentence, using the adapted LM weight for the difficult segment and the baseline weight for the rest of the sentence.

For this experiment, we use a new test set of 661 sentences. This is the NIST-2002 Arabic-English multi-translation test set (LDC2003T18). In addition to the baseline MT system trained on a relatively small corpus of one million words, we also repeat the experiment with a baseline MT system trained on more data. The second baseline system is trained on a parallel corpus of 50 million words. This training set is a concatenation of three corpora (LDC2004E13, LDC2004E72, LDC2005E46).

System/Corpus Size	1M Words	50M Words
Baseline	18.09	22.51
Wt. Adapt. (Rank)	19.25	23.36

Table 8: Adaptation of LM weight on the Small and Medium Systems (sentence-level BLEU evaluation)

Table 8 presents the results of the start-to-finish experiment. Without knowing for certain whether the segment chosen for adaptation is indeed the most difficult segment, the weight adaptation nonetheless improved the overall translation quality. For the baseline MT system used in previous experiments, we observed a 1.16 improvement in BLEU score (cf. a 2.28 BLEU improvement in Table 7 when the most difficult segments are identified.). When the baseline MT system is trained on a larger parallel corpus, there is still a 0.85 BLEU score improvement. This suggests that individual weight adaptation may still be helpful for larger MT systems.

7 Related Work

The major concepts used in our work are adaptation, re-scoring, automatic MT evaluation, and ranking; they have been widely studied in the MT literature. In this section, we highlight some of the most relevant previous work. In previous work on MT adaptation, many proposed to modify the baseline system to incorporate features from the source language, lexical translations and from the underlying system itself (Snover et al., 2008; Tam et al., 2007; Kim, 2004). In these studies, the components of the baseline SMT system are modified. Furthermore, new models are constructed for the translation of special test sets or individual phrases and sentences. In contrast, our approach does not directly change the baseline components. Instead, we allow the decoder to adjust the influence from a component according to its expected performance. To perform this kind of judgment, we make use of techniques similar to confidence estimation and automatic MT evaluation. Previous work on MT evaluation without references focuses on sentential or corpus level eval-

uation and uses regression to predict a translation quality score (Albrecht and Hwa, 2007; Specia et al., 2009). In contrast, we apply the evaluation metric to segments at a sub-sentential level. We are less interested in the absolute scores than in the quality of the candidates relative to each other. In this way, our approach is more similar to the metric of (Duh, 2008), in which the evaluation is conducted by ranking. Also relevant is the series of work on system modification, such as post-decoding discriminative re-ranking. The discriminative re-ranking benefits from a set of richer but more computationally expensive features. These features range from deeper linguistic knowledge (Och et al., 2004) to system related knowledge like word and phrase level confidence score (Zens and Ney, 2006). Similarly we benefit from additional features in our weight ranking; however, the set of candidate hypotheses generated for weight ranking is different from a decoder's n-best list.

8 Conclusion

In contrast with the traditional method of using static decoding weights, here we introduced a framework of using dynamic decoding weights. We explored varying the language model's decoder weights based on the characteristics of the source text. Following the insight that the weight adaptation should be performed on the part of a sentence with which the baseline MT system is having problems, we turn our attention to difficult source segments. By limiting the scope of our modification (one weight and one segment), we were able to offset some shortcomings of the language model (for example, its model of short translations is less reliable). In terms of the adaptation strategy, we experimented with both learning one shared weight setting for all the difficult segments and learning to individual weight setting for each difficult segment. Our results suggest that:

- Using dynamic decoding weight for SMT is a promising avenue to explore. We observed translation quality improvements when we used different language model weights for different segments of a sentence.
- Difficult segments share some common characteristics (e.g., compound verb translation) that cause problems for PB-SMT. Some of the damages can be reduced by appropriate adjustment of the LM weight. In contrast, we did not find any common pattern of problems that can be addressed through weight adjustments in the "easy" segments.
- Adjusting the LM weight for each individual difficult segment improves the translation quality more than finding one shared LM weight for all difficult segments. We also showed that the appropriate amount of weight adjustment can be learned for individual segments. In particular, we found preference ranking to have worked well.

Acknowledgments

This research has been supported by US National Science Foundation Grants IIS-0712810 and IIS-0745914. The presentation of this work was made possible by the support of the Qatar Foundation through Carnegie Mellon University's Seed Research program. We thank Kemal Oflazer for facilitating the CMU-Qatar support.

References

Joshua Albrecht and Rebecca Hwa. 2007. Regression for sentence-level mt evaluation with pseudo references. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL-07)*, Prague, Czech Republic.

Yunbo Cao, Jun Xu, Tie-Yan Liu, Hang Li, Yalou Hang, and Hsiao-Wuen Hon. 2006. Adapting ranking svm to document retrieval. In *Proceedings of the ACM Conference of the Special Interest Group on Information Retrieval (SIG-IR)*.

Kevin Duh. 2008. Ranking vs. regression in machine translation evaluation. In *Proceedings of the Third ACL Workshop on Statistical Machine Translation*, Columbus, Ohio.

Thorsten Joachims. 1998. Making large-scale svm learning practical. In Bernhard Schölkopf, Chris Burges, and Alex Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, pages 169–185. MIT Press.

Woosung Kim. 2004. *Language Model Adaptation for Automatic Speech Recognition and Statistical Machine Translation*. Ph.D. thesis, Johns Hopkins University, Baltimore, MD.

Philip Koehn. 2004. Pharaoh: A beam search decoder for phrase-based statistical machine translation models. Technical report, USC Information Sciences Institute, Marina Del Rey, USA.

Behrang Mohit and Rebecca Hwa. 2007. Localization of difficult-to-translate phrases. In *Proceedings of the Second ACL Workshop on Statistical Machine Translation*, Prague, Czech Republic.

Behrang Mohit, Frank Liberato, and Rebecca Hwa. 2009. Language model adaptation for difficult-to-translate phrases. In *Proceedings of the Conference of the European Association of Machine Translation (EAMT-09)*, Barcelona, Spain.

Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. 2004. A smorgasbord of features for statistical machine translation. In *HLT-NAACL 2004: Main Proceedings*, Boston, Massachusetts, USA.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL-03)*.

Marian Olteanu, Chris Davis, Ionut Volosen, and Dan Moldovan. 2006. Phramer - an open source statistical phrase-based translator. In *Proceedings on the Workshop on Statistical Machine Translation*, New York City.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Matthew Snover, Bonnie Dorr, and Richard Schwartz. 2008. Language and translation model adaptation using comparable corpora. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, Honolulu, Hawaii.

Lucia Specia, Marco Turchi, Nicola Cancedda, Marc Dymetman, and Nello Cristianini. 2009. Estimating the sentence-level quality of machine translation systems. In *Proceedings of the Conference of the European Association of Machine Translation (EAMT-09)*, Barcelona, Spain.

A. Stolcke. 2002. Srilm – an extensible language modeling toolkit. In *Proceedings of the International Conference on Spoken Language Processing*.

Yik-Cheung Tam, Ian Lane, and Tania Schultz. 2007. Bilingual-lsa based lm adaptation for spoken language translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic.

Richard Zens and Hermann Ney. 2006. N-gram posterior probabilities for statistical machine translation. In *Proceedings on the Workshop on Statistical Machine Translation*, New York City.