Language Model Adaptation for Statistical Machine Translation with Structured Query Models

Bing Zhao Matthias Eck Stephan Vogel
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA, 15213, USA
{bzhao, matteck, vogel+}@cs.cmu.edu

Abstract

We explore unsupervised language model adaptation techniques for Statistical Machine Translation. The hypotheses from the machine translation output are converted into queries at different levels of representation power and used to extract similar sentences from very large monolingual text collection. Specific language models are then build from the retrieved data and interpolated with a general background model. Experiments show significant improvements when translating with these adapted language models.

1 Introduction

Language models (LM) are applied in many natural language processing applications, such as speech recognition and machine translation, to encapsulate syntactic, semantic and pragmatic information. For systems which learn from given data we frequently observe a severe drop in performance when moving to a new genre or new domain. In speech recognition a number of adaptation techniques have been developed to cope with this situation. In statistical machine translation we have a similar situation, i.e. estimate the model parameter from some data, and use the system to translate sentences which may not be well covered by the training data. Therefore, the potential of adaptation techniques needs to be explored for machine translation applications.

Statistical machine translation is based on the noisy channel model, where the translation hypothesis is searched over the space defined by a translation model and a target language (Brown et al, 1993). Statistical machine translation can be formulated as follows:

$$t^* = \arg\max_t P(t \mid s) = \arg\max_t P(s \mid t) \cdot P(t)$$

where t is the target sentence, and s is the source sentence. $P(t)$ is the target language model and $P(s|t)$ is the translation model. The argmax operation is the search, which is done by the decoder.

In the current study we modify the target language model $P(t)$, to represent the test data better, and thereby improve the translation quality. (Janiszek, et al. 2001) list the following approaches to language model adaptation:

- Linear interpolation of a general and a domain specific model (Seymore, Rosenfeld, 1997).
- Back off of domain specific probabilities with those of a specific model (Besling, Meier, 1995).
- Retrieval of documents pertinent to the new domain and training a language model on-line with those data (Iyer, Ostendorf, 1999, Mahajan et. al. 1999).
- Maximum entropy, minimum discrimination adaptation (Chen, et. al., 1998).
- Adaptation by linear transformation of vectors of bigram counts in a reduced space (DeMori, Federico, 1999).
- Smoothing and adaptation in a dual space via latent semantic analysis, modeling long-term semantic dependencies, and trigger combinations. (J. Bellegarda, 2000).

Our approach can be characterized as unsupervised data augmentation by retrieval of relevant documents from large monolingual corpora, and interpolation of the specific language model, build from the retrieved data, with a background language model. To be more specific, the following steps are carried out to do the language model adaptation. First, a baseline statistical machine translation system, using a large general language model, is applied to generate initial translations. Then these translations hypotheses are reformulated as queries to retrieve similar sentences from a very large text collection. A small domain specific language model is build using the retrieved sentences and linearly interpolated with the background language model. This new interpolated language model in applied in a second decoding run to produce the final translations.

There are a number of interesting questions pertaining to this approach:

- Which information can and should used to generate the queries: the first-best translation only, or also translation alternatives.
- How should we construct the queries, just as simple bag-of-words, or can we incorporate more structure to make them more powerful.
- How many documents should be retrieved to build the specific language models, and on what granularity should this be done, i.e. what is a document in the information retrieval process.

The paper is structured as follows: section 2 outlines the sentence retrieval approach, and three bag-of-words query models are designed and explored; structured query models are introduced in section 3. In section 4 we present translation experiments are presented for the different query. Finally, summary is given in section 5.

2 LM Adaptation via Sentence Retrieval

Our language model adaptation is an unsupervised data augmentation approach guided by query models. Given a baseline statistical machine translation system, the language model adaptation is done in several steps shown as follows:

- ❖ Generate a set of initial translation hypotheses $H = \{h_1 \ldots h_n\}$ for source sentences *s,* using either the baseline MT system with the background language model or only the translation model
- ❖ Use H to build query
- ❖ Use query to retrieve relevant sentences from the large corpus
- ❖ Build specific language models from retrieved sentences
- ❖ Interpolate the specific language model with the background language
- ❖ Re-translate sentences *s* with adapted language model

Figure-1: Adaptation Algorithm

The specific language model $P_A(w_i \mid h)$ and the general background model $P_B(w_i \mid h)$ are combined using linear interpolation:

$$\hat{P}(w_i \mid h) = \lambda P_B(w_i \mid h) + (1-\lambda) P_A(w_i \mid h) \qquad (1)$$

The interpolation factor λ can be simply estimated using cross validation or a grid search.

As an alternative to using translations for the baseline system, we will also describe an approach, which uses partial translations of the source sentence, using the translation model only. In this case, no full translation needs to be carried out in the first step; only information from the translation model is used.

Our approach focuses on query model building, using different levels of knowledge representations from the hypothesis set or from the translation model itself. The quality of the query models is crucial to the adapted language model's performance. Three bag-of-words query models are proposed and explained in the following sections.

2.1 Sentence Retrieval Process

In our sentence retrieval process, the standard *tf/idf* (*term frequency* and *inverse document frequency*) term weighting scheme is used. The queries are built from the translation hypotheses. We follow (Eck, et al., 2004) in considering each sentence in the monolingual corpus as a document, as they have shown that this gives better results compared to retrieving entire news stories.

Both the query and the sentences in the text corpus are converted into vectors by assigning a *term weight* to each word. Then the cosine similarity is calculated proportional to the inner product of the two vectors. All sentences are ranked according to their similarity with the query, and the most similar sentences are used as the data for building the specific language model. In our experiments we use different numbers of similar sentences, ranting from one to several thousand.

2.2 Bag-of-words Query Models

Different query models are designed to guide the data augmentation efficiently. We first define "bag-of-words" models, based on different levels of knowledge collected from the hypotheses of the statistical machine translation engine.

2.2.1 First-best Hypothesis as a Query Model

The first-best hypothesis is the Viterbi path in the search space returned from the statistical machine translation decoder. It is the optimal hypothesis the statistical machine translation system can generate using the given translation and language model, and restricted by the applied pruning strategy. Ignoring word order, the hypothesis is converted into a bag-of-words representation, which is then used as a query:

$$Q_{T1} = (w_1, w_2, \cdots w_l) = \{(w_i, f_i) \mid w_i \in V_{T1}\}$$

where w_i is a word in the vocabulary V_{T1} of the Top-1 hypothesis. f_i is the frequency of w_i's occurrence in the hypothesis.

The first-best hypothesis is the actual translation we want to improve, and usually it captures enough correct word translations to secure a sound adaptation process. But it can miss some

informative translation words, which could lead to better-adapted language models.

2.2.2 N-Best Hypothesis List as a Query Model

Similar to the first-best hypothesis, the n-best hypothesis list is converted into a bag-of-words representation. Words which occurred in several translation hypotheses are simply repeated in the bag-of-words representations.

$$Q_{TN} = (w_{1,1}, w_{1,2}, \cdots w_{1,l_1}; \cdots; w_{N,1}, w_{N,2}, \cdots w_{N,l_N}) = \{(w_i, f_i) \mid w_i \in V_{TN}\}$$

where V_{TN} is the combined vocabulary from all n-best hypotheses and f_i is the frequency of w_i's occurrence in the n-best hypothesis list.

Q_{TN} has several good characteristics: First it contains translation candidates, and thus is more informative than Q_{T1}. In addition, the confidently translated words usually occur in every hypothesis in the n-best list, therefore have a stronger impact on the retrieval result due to the higher term frequency (*tf*) in the query. Thirdly, most of the hypotheses are only different from each other in one word or two. This means, there is not so much noise and variance introduced in this query model.

2.2.3 Translation Model as a Query Model

To fully leverage the available knowledge from the translation system, the translation model can be used to guide the language model adaptation process. As introduced in section 1, the translation model represents the full knowledge of translating words, as it encodes all possible translations candidates for a given source sentence. Thus the query model based on the translation model, has potential advantages over both Q_{T1} and Q_{TN}.

To utilize the translation model, all the n-grams from the source sentence are extracted, and the corresponding candidate translations are collected from the translation model. These are then converted into a bag-of-words representation as follows:

$$Q_{TM} = (w_{s_1,1}, w_{s_1,2}, \cdots w_{s_1,n_1}; \cdots; w_{s_I,1}, w_{s_I,2}, \cdots w_{s_I,n_I}) = \{(w_i, f_i) \mid w_i \in V_{TM}\}$$

where s_i is a source n-gram, and *I* is the number of n-grams in the source sentence. $w_{s_i,j}$ is a candidate target word as translation of s_i. Thus the translation model is converted into a collection of target words as a bag-of-word query model.

There is no decoding process involved to build Q_{TM}. This means Q_{TM} does not incorporate any background language model information at all, while both Q_{T1} and Q_{TN} implicitly use the background language model to prune the words in the query. Thus Q_{TM} is a generalization, and Q_{T1} and Q_{TN} are pruned versions. This also means Q_{TM} is subject to more noise.

3 Structured Query Models

Word proximity and word order is closely related to syntactic and semantic characteristics. However, it is not modeled in the query models presented so far, which are simple bag-of-words representations. Incorporating syntactic and semantic information into the query models can potentially improve the effectiveness of LM adaptation.

The word-proximity and word ordering information can be easily extracted from the first-best hypothesis, the n-best hypothesis list, and the translation lattice built from the translation model. After extraction of the information, structured query models are proposed using the *structured query language*, described in the Section 3.1.

3.1 Structured Query Language

This query language essentially enables the use of proximity operators (ordered and unordered windows) in queries, so that it is possible to model the syntactic and semantic information encoded in phrases, n-grams, and co-occurred word pairs.

The InQuery implementation (Lemur 2003) is applied. So far 16 operators are defined in InQuery to model word proximity (ordered, unordered, phrase level, and passage level). Four of these operators are used specially for our language model adaptation:

Sum Operator: #sum($t_1 \ldots t_n$)

The terms or nodes ($t_1 \ldots t_n$) are treated as having equal influence on the final retrieval result. The belief values provided by the arguments of the sum are averaged to produce the belief value of the #sum node.

Weighted Sum Operator: #wsum($w_1 : t_1$, …)

The terms or nodes ($t_1 \ldots t_n$) contribute unequally to the final result according to the weight (w_i) associated with each t_i.

Ordered Distance Operator: #N($t_1 \ldots t_n$)

The terms must be found within N words of each other in the text in order to contribute to the document's belief value. An n-gram phrase can be modeled as an ordered distance operator with N=n.

Unordered Distance Operator: #uwN($t_1 \ldots t_n$)

The terms contained must be found in any order within a window of N words in order for this operator to contribute to the belief value of the document.

3.2 Structured Query Models

Given the representation power of the structured query language, the Top-1 hypothesis, Top-N Best hypothesis list, and the translation lattice can be converted into three Structured Query Models respectively.

For first-best and n-best hypotheses, we collect related target n-grams of a given source word according to the alignments generated in the Viterbi decoding process. While for the translation lattice, similar to the construction of Q_{TM}, we collect all the source n-grams, and translate them into target n-grams. In either case, we get a set of target n-grams for each source word. The structured query model for the whole source sentence is a collection of such subsets of target n-grams.

$$Q_{st} = \{\vec{t}_{s_1}, \vec{t}_{s_2}, \cdots, \vec{t}_{s_I}\}$$

$\vec{t}_{s_i}$ is a set of target n-grams for the source word s_i:

$$\vec{t}_{s_i} = \{\{t_i, \cdots\}_{1-gram}; \{t_i t_{i+1}, \cdots\}_{2-gram}; \{t_{i-1} t_i t_{i+1}\}_{3-gram} \cdots\}$$

In our experiments, we consider up to trigram for better retrieval efficiency, but higher order n-grams could be used as will. The second simplification is that every source word is equally important, thus each n-gram subset $\vec{t}_{s_i}$ will have an equal contribution to the final retrieval results. The last simplification is each n-gram within the set of $\vec{t}_{s_i}$ has an equal weight, i.e. we do not use the translation probabilities of the translation model. If the system is a phrase-based translation system, we can encode the phrases using the ordered distance operator (#N) with N equals to the number of the words of that phrase, which is denoted as the #phrase operator in InQuery implementation. The 2-grams and 3-grams can be encoded using this operator too.

Thus our final structured query model is a sum operator over a set of nodes. Each node corresponds to a source word. Usually each source word has a number of translation candidates (unigrams or phrases). Each node is a weighted sum over all translation candidates weighted by their frequency in the hypothesis set. An example is shown below, where #phrase indicates the use of the ordered distance operator with varying n:

```
#q=#sum( #wsum(2 eu  2 #phrase(european union) )
         #wsum(12 #phrase(the united states)
                1 american 1 #phrase(an american) )
         #wsum(4 are 1 is )
         #wsum(8 markets  3 market))
         #wsum(7 #phrase(the main)  5 primary ) );
```

4 Experiments

Experiments are carried out on a standard statistical machine translation task defined in the NIST evaluation in June 2002. There are 878 test sentences in Chinese, and each sentence has four human translations as references. NIST score (NIST 2002) and Bleu score (Papineni et. al. 2002) of mteval version 9 are reported to evaluate the translation quality.

4.1 Baseline Translation System

Our baseline system (Vogel et al., 2003) gives scores of 7.80 NIST and 0.1952 Bleu for Top-1 hypothesis, which is comparable to the best results reported on this task.

For the baseline system, we built a translation model using 284K parallel sentence pairs, and a trigram language model from a 160 million words general English news text collection. This LM is the background model to be adapted.

With the baseline system, the n-best hypotheses list and the translation lattice are extracted to build the query models. Experiments are carried out on the adapted language model using the three bag-of-words query models: Q_{T1}, Q_{TN} and Q_{TM}, and the corresponding structured query models.

4.2 Data: GigaWord Corpora

The so-called GigaWord corpora (LDC, 2003) are very large English news text collections. There are four distinct international sources of English newswire:

AFE	Agence France Press English Service
APW	Associated Press Worldstream English Service
NYT	The New York Times Newswire Service
XIE	The Xinhua News Agency English Service

Table-1 shows the size of each part in word counts.

AFE	APW	NYT	XIE
170,969K	539,665K	914,159K	131,711K

Table-1: Number of words in the different GigaWord corpora

As the Lemur toolkit could not handle the two large corpora (APW and NYT) we used only 200 million words from each of these two corpora.

In the preprocessing all words are lowercased and punctuation is separated. There is no explicit removal of stop words as they usually fade out by *tf.idf* weights, and our experiments showed not positive effects when removing stop words.

4.3 Bag-of-Words Query Models

Table-2 shows the size of Q_{T1}, Q_{TN} and Q_{TM} in terms of number of tokens in the 878 queries:

	Q_{T1}	Q_{TN}	Q_{TM}
$\|Q\|$	25,861	231,834	3,412,512

Table-2: Query size in number of tokens

As words occurring several times are reduced to word-frequency pairs, the size of the queries generated from the 100-best translation lists is only 9 times as big as the queries generated from the first-best translations. The queries generated from the translation model contain many more translation alternatives, summing up to almost 3.4 million tokens. Using the lattices the whole information of the translation model is kept.

4.3.1 Results for Query Q_{T1}

In the first experiment we used the first-best translations to generate the queries. For each of the 4 corpora different numbers of similar sentences (1, 10, 100, and 1000) were retrieved to build specific language models. Figure-2 shows the language model adaptation after tuning the interpolation factor λ by a grid search over [0,1]. Typically λ is around 0.80.

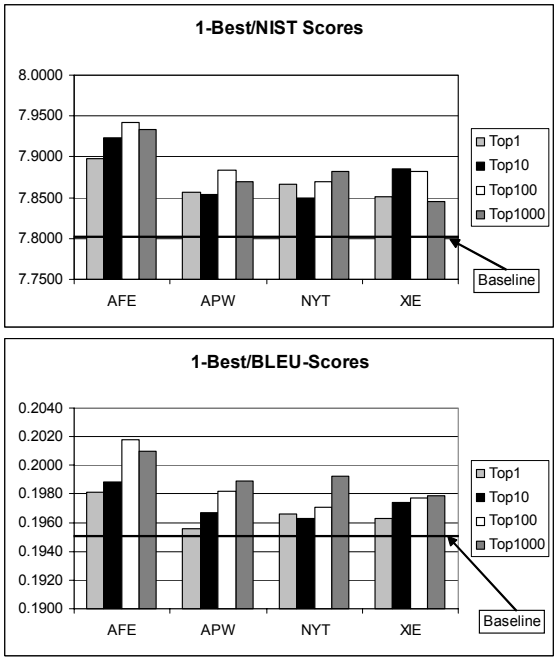

Figure-2: NIST and Bleu scores Q_{T1}

We see that each corpus gives an improvement over the baseline. The best NIST score is 7.94, and the best Bleu score is 0.2018. Both best scores are realized using top 100 relevant sentences corpus per source sentence mined from the AFE.

4.3.2 Results for Query Q_{TN}

Figure-3 shows the results for the query model Q_{TN}. The best results are 7.99 NIST score, and 0.2022 Bleu score. These improvements are statistically significant. Both scores are achieved at the same settings as those in Q_{T1}, i.e. using top 100 retrieved relevant sentences mined from the AFE corpus.

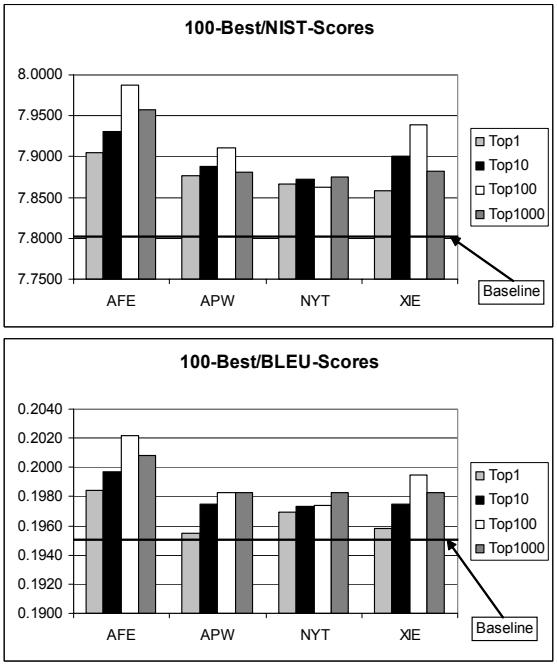

Figure-3: NIST and Bleu scores from Q_{TN}

Using the translation alternatives to retrieve the data for language model adaptation gives an improvement over using the first-best translation only for query construction. Using only one translation hypothesis to build an adapted language model has the tendency to reinforce that translation.

4.3.3 Results for Query Q_{TM}

The third bag-of-words query model uses all translation alternatives for source words and source phrases. Figure-4 shows the results of this query model Q_{TM}. The best results are 7.91 NIST score and 0.1995 Bleu. For this query model best results were achieved using the top 1000 relevant sentences mined from the AFE corpus per source sentence.

The improvement is not as much as the other two query models. The reason is probably that all translation alternatives, even wrong translations resulting from errors in the word and phrase alignment, contribute alike to retrieve similar sentences. Thereby, an adapted language model is built, which reinforces not only good translations, but also bad translations.

All the three query models showed improvements over the baseline system in terms of NIST and Bleu scores. The best bag-of-words query model is Q_{TN} built from the N-Best list. It provides a good balance between incorporating translation alternatives in the language model adaptation process and not reinforcing wrong translations.

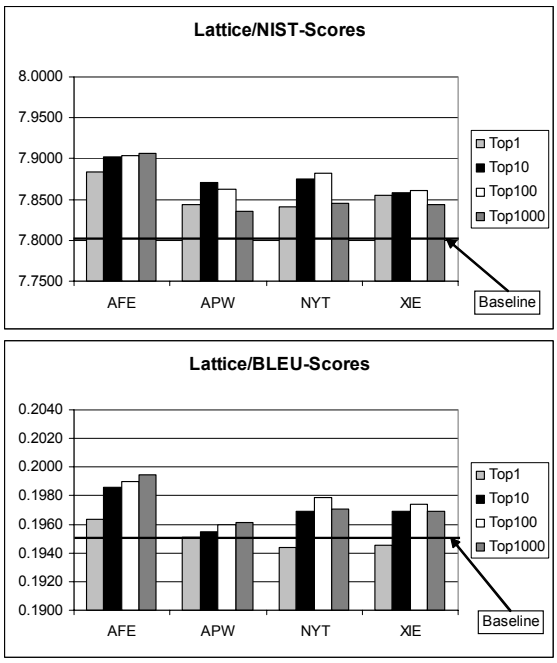

Figure-4: NIST and Bleu scores from Q_{TM}

4.4 Structured Query Models

The next series of experiments was done to study if using word order information in constructing the queries could help to generate more effective adapted language models. By using the structured query language we converted the same first-best hypothesis, the 100-best list, and the translation lattice into structured query models. Results are reported for the AFE corpus only, as this corpus gave best translation scores.

Figure-5 shows the results for all three structured query models, built from the first-best hypothesis ("1-Best"), the 100 best hypotheses list ("100-Best"), and translation lattice ("TM-Lattice"). Using these query models, different numbers of most similar sentences, ranging from 100 to 4000, where retrieved from the AFE corpus. The given baseline results are the best results achieved from the corresponding bag-of-words query models.

Consistent improvements were observed on NIST and Bleu scores. Again, optimal interpolation factors to interpolate the specific language models with the background language model were used, which typically were in the range of [0.6, 0.7]. Structured query models give most improvements when using more sentences for language model adaptation. The effect is more pronounced for Bleu then for NIST score.

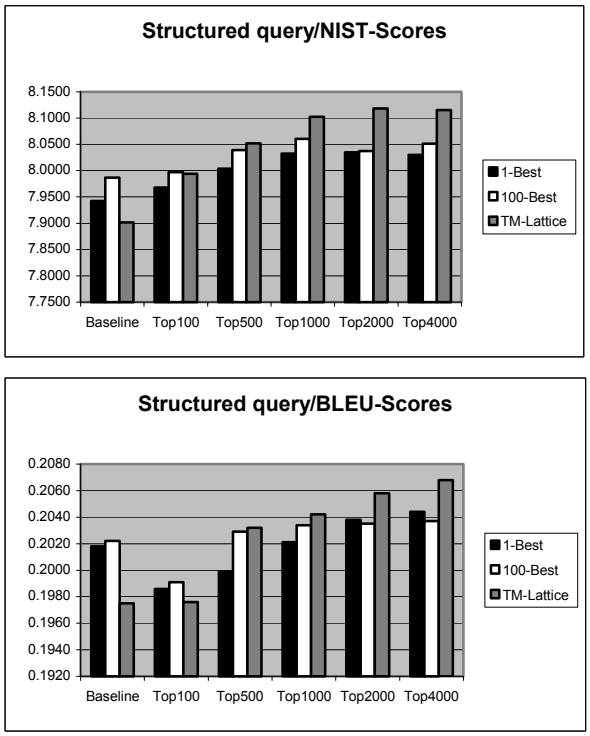

Figure-5: NIST and Bleu scores from the structured query models

The really interesting result is that the structured query model Q_{TM} gives now the best translation results. Adding word order information to the queries obviously helps to reduce the noise in the retrieved data by selecting sentences, which are closer to the good translations,

The best results using the adapted language models are NIST score 8.12 for using the 2000 most similar sentences, whereas Bleu score goes up to 0.2068 when using 4000 sentences for language model adaptation.

4.5 Example

Table-3 shows translation examples for the 17th Chinese sentence in the test set. We applied the baseline system (Base), the bag-of-word query model (Hyp1), and the structured query model (Hyp2) using AFE corpus.

Ref	The police has already blockade the scene of the explosion.
Base	At present, the police had cordoned off the explosion.
Hyp1	At present, police have sealed off the explosion.
Hyp2	Currently, police have blockade on the scene of the explosion.

Table-3 Translation examples

4.6 Oracle Experiment

Finally, we run an oracle experiments to see how much improvement could be achieved if we only selected better data for the specific language models. We converted the four available reference translations into structured query models and retrieved the top 4000 relevant sentences from AFE corpus for each source sentence. Using these language models, interpolated with the background language model gave a NIST score of 8.67, and a Bleu score of 0.2228. This result indicates that there is room for further improvements using this language model adaptation technique.

The oracle experiment suggests that better initial translations lead to better language models and thereby better 2nd iteration translations. This lead to the question if we can iterate the retrieval process several times to get further improvement, or if the observed improvement results form using for (good) translations, which have more diversity than the translations in an n-best list.

On the other side the oracle experiment also shows that the optimally expected improvement is limited by the translation model and decoding algorithm used in the current SMT system.

5 Summary

In this paper, we studied language model adaptation for statistical machine translation. Extracting sentences most similar to the initial translations, building specific language models for each sentence to be translated, and interpolating those with the background language models gives significant improvement in translation quality. Using structured query models, which capture word order information, leads to better results that plain bag of words models.

The results obtained suggest a number of extensions of this work: The first question is if more data to retrieve similar sentences from will result in even better translation quality. A second interesting question is if the translation probabilities can be incorporated into the queries. This might be especially useful for structured query models generated from the translation lattices.

References

J. Bellegarda. 2000, Exploiting Latent Semantic Information in Statistical Language Modeling. In Proceedings of the IEEE, 88(8), pp. 1279-1296.

S. Besling and H.G. Meier 1995. Language Model Speaker Adaptation, Eurospeech 1995, Madrid, Spain.

Peter F Brown., Stephen A Della Pietra., Vincent J. Della Pietra and Mercer Robert L., 1993. The Mathematics of Statistical Machine Translation: Parameter Estimation. Computational Linguistics, 19(2), pp. 263–311.

S.F Chen., K. Seymore, and R. Rosenfeld 1998. Topic Adaptation for Language Modeling using Unnormalized Exponential Models. IEEE International Conference on Acoustics, Speech and Signal Processing 1998, Seattle WA.

Renato DeMori and Marcello Federico 1999. Language Model Adaptation, In Computational Models of Speech Pattern Processing, Keith Pointing (ed.), NATO ASI Series, Springer Verlag.

Matthias Eck, Stephan Vogel, and Alex Waibel, 2004. Language Model Adaptation for Statistical Machine Translation based on Information Retrieval, International Conference on Language Resources and Evaluation, Lisbon, Portugal.

R. Iyer and M. Ostendorf, 1999. Modeling Long Distance Dependence in Language: Topic Mixtures vs. Dynamic Cache Models, IEEE Transactions on Speech and Audio Processing, SAP-7(1): pp. 30-39.

David Janiszek, Renato DeMori and Frederic Bechet, 2001. Data Augmentation and Language Model adaptation, IEEE International Conference on Acoustics, Speech and Signal Processing 2001, Salt Lake City, UT.

LDC, Gigaword Corpora. http://wave.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2003T05

Lemur, The Lemur Toolkit for Language Modeling and Information Retrieval, http://www.cs.cmu.edu/~lemur/

Milind Mahajan, Doug Beeferman and X.D. Huang, 1999. Improved Topic-Dependent Language Modeling Using Information Retrieval Techniques, IEEE International Conference on Acoustics, Speech and Signal Processing 1999, Phoenix, AZ.

NIST Report: 2002, Automatic Evaluation of Machine Translation Quality Using N-gram Co-Occurrence Statistics. http://www.nist.gov/speech/tests/mt/doc/ngram-study.pdf .

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu, 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In Proc of the 40th Annual Meeting of the Association for Computational Linguistics. 2002, Philadelphia, PA.

Kristie Seymore and Ronald Rosenfeld, 1997. Using Story Topics for Language Model Adaptation. In Proc. Eurospeech 1997, Rhodes, Greece.

Stephan Vogel, Ying Zhang, Fei Huang, Alicia Tribble, Ashish Venogupal, Bing Zhao, Alex Waibel, 2003. The CMU Statistical Translation System, Proceedings of MT-Summit IX, 2003, New Orleans, LA.