Low Cost Portability for Statistical Machine Translation based on N-gram Coverage

Matthias Eck, Stephan Vogel and Alex Waibel
Interactive Systems Laboratories
Carnegie Mellon University
Pittsburgh, PA, 15213, USA
matteck@cs.cmu.edu, vogel+@cs.cmu.edu, waibel@cs.cmu.edu

Abstract

Statistical machine translation relies heavily on the available training data. However, in some cases, it is necessary to limit the amount of training data that can be created for or actually used by the systems. To solve that problem, we introduce a weighting scheme that tries to select more informative sentences first. This selection is based on the previously unseen n-grams the sentences contain, and it allows us to sort the sentences according to their estimated importance. After sorting, we can construct smaller training corpora, and we are able to demonstrate that systems trained on much less training data show a very competitive performance compared to baseline systems using all available training data.

1 Introduction

The goal of this research was to decrease the amount of training data that is necessary to train a competitive statistical machine translation system regardless of the actual test data or its domain. "Competitive" here means that the system should not produce significantly worse translations compared to a system trained on a significantly larger amount of data.

It is important to note that we assume that the test data (and its domain) is not known at the time we select the actual training data. This means the test data has no influence on the selection process.

Statistical machine translation can be described in a formal way as follows:

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

Here 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 used in the decoder.

Statistical machine translation searches for the best target sentence from the space defined by the target language model and the translation model.

Statistical translation models are usually either phrase- or word-based and include most notably IBM1 to IBM4 and HMM (Brown et al., 1993; Vogel et al., 1996; Vogel et al., 2003). Some recent developments focused on online phrase extraction (Vogel et al., 2004).

All models use available bilingual training data in the source and target languages to estimate their parameters and approximate the translation probabilities.

One of the main problems of Statistical Machine Translation is the necessity to have large parallel corpora available. This might not be a big issue for major languages, but it certainly is a problem for languages with less resources. To improve the data situation for these languages, it is necessary to hire human translators at enormous costs who translate corpora that can later be used to train statistical machine translation systems.

Our idea focuses on sorting the available source sentences that should be translated by a human translator according to their approximate importance. The importance is estimated using the unseen n-grams of a sentence to be sorted given the sentences that were selected earlier. This means the algorithms will try to get the best possible unseen n-gram coverage for each newly selected sentence. We refine this later by taking the length of the respective sentence into account as well.

2 Motivation

There are three inherently different motivations for the goal of limiting the amount of necessary training data for a competitive system that are best described by their actual applications.

2.1 Application 1: Reducing Human Translation Cost

The main problem of portability of SMT systems to new languages is the involved cost to generate parallel bilingual training data as it is necessary to have sentences translated by human translators.

An assumption could be that a 1 million word corpus needs to be translated to a new language in order to build a decent SMT system.

A human translator could charge in the range of approximately 0.10-0.25 USD per word depending on the involved languages and the difficulty of the text. The translation of a 1 million word corpus would then cost between 100,000 and 250,000 USD.

The concept here is to select the most important sentences from the original 1 million word corpus and have only those translated by the human translators. If it would still be possible to get a similar translation performance with a significantly lower translation effort, a considerable amount of money could be saved.

This could especially be applied to low density languages with limited resources (compare Lavie et al., 2004; McEnery et al., 2000).

2.2 Application 2: Translation on Small Devices

Another possible application is the usage of statistical machine translation on portable small devices like PDAs or cell phones. Those devices tend to have a limited amount of memory available which limits the size of the models the device can actually hold and a larger training corpus will usually result in a larger model. The more recent approaches to online phrase extraction for SMT make it necessary to have the corpus available (and in memory) at the time of translation (Callison-Burch et al., 2005; Zhang and Vogel, 2005).

Given the upper example, a small device might not be able to hold a 1 million word bilingual corpus but e.g. only a corpus with 200,000 words. The question is now which part of the corpus (especially which sentences) should be selected and put on the device to get the best possible translation system.

2.3 Application 3: Standard Translation System

Even on larger devices that do not have rigid limitations of memory, the approach could be helpful. The complexity of online phrase extraction and standard training algorithms depends mainly on the size of the bilingual training data. Limiting the size of the training data with the same translation performance on these devices would speed up the translations.

Another problem is that the still widely used 32 bit machines like the Intel Pentium 4 and AMD Athlon XP series can only address up to 4 gigabytes of memory. There are already bilingual corpora in excess of 4 gigabytes available, and therefore, it is necessary to select the most important sentences from these corpora to be able to hold them in memory. (The last issue will certainly be resolved by the widespread introduction of 64 bit machines which can theoretically address 17 million terabytes of memory.)

3 Previous Work

In general this research can be regarded as an example of active learning. This means the machine learning algorithm does not just passively train on the available training data but plays an active role in selecting the best training data. Active learning, as a standard method in machine learning, has been applied to a variety of problems in natural language processing, for example to parsing (Hwa, 2004) and to automatic speech recognition (Kamm and Meyer, 2002).

It is important to note the difference between this approach and approaches to Translation Model Adaptation (Hildebrand et al., 2005) or simple sub-sampling techniques that are based on the actual test data. Here, we assume that the test data is not known at selection time, so the intention is to get the best possible translation system for every possible test data.

4 Description of Sentence Sorting

4.1 Algorithm

The sentences are sorted according to the following very simple algorithm.

```
For all sentences that are not in the sorted list:
        Calculate weight of sentences
        Find sentence with highest weight
        Add sentence with highest weight to sorted list
```

4.2 Weighting of Sentences

The interesting part is the calculation of the weight of each sentence. The weight of a sentence will generally depend on the previously selected sentences.

As mentioned before, state-of-the-art statistical machine translation systems are using word- and especially phrase-based translation models. This means they align words and phrases in the source and target sentences of the training data and use these as the building blocks for the final translations. It is obvious that the performance of a translation system will then largely depend on the word and phrase coverage of the test data. This led us to the first idea to try to optimize the word and phrase (or generally n-gram) coverage by just using the number of previously unseen n-grams in a sentence as its weight.

We tried this approach for n-grams up to trigrams. This gives the following easy terms for the calculation of the weight.

$$\text{weight}_{\text{j}}(sentence) = \sum_{n=1}^{j} \#(\text{unseen } n-\text{grams})$$

The parameter *j* here determines the n-grams that are considered and was set to values of 1, 2 and 3 in the experiments. These simple weighting schemes already show improvements over the baseline systems as shown in the later parts of the paper, but they have various shortcomings.

They focus only on improving the coverage but do not take the actual translation cost of the sentence into account (Translators generally charge per word and not per sentence). This leads to the fact that longer sentences tend to get higher weights than shorter sentences because they might contain more unseen n-grams. The focus on coverage [1] is certainly very helpful but longer sentences are more difficult for the training of statistical translation models.

(When training the translation model IBM1, for example, every possible word alignment between sentences is considered.)

To fix these shortcomings, we changed the weighting terms to incorporate the actual length of a sentence by dividing the number of unseen n-grams by the length of the sentence (in words):

$$\text{weight}_{\text{j}}(sentence) = \frac{\sum_{n=1}^{j} \#(\text{unseen } n-\text{grams})}{|sentence|}$$

This changes the weight to – informally speaking – "new n-grams per word to translate."

As noted earlier, the algorithms for training translation models in statistical machine translation usually work better (and faster) on shorter sentences. For this reason, we also tried to divide the number of unseen n-grams by the square of the length of a sentence which prefers even shorter sentences.

[1] Please note that although these weighting schemes focus exclusively on improving the n-gram coverage, they are not optimal in this sense. There are situations where it is possible to order sentences differently and reach 100% coverage earlier than when using this algorithm; but this is merely a theoretical issue and certainly does not significantly affect the results.

Overall the weighting terms can be written as:

$$\text{weight}_{\text{i, j}}(sentence) = \frac{\sum_{n=1}^{j} \#(\text{unseen } n-\text{grams})}{|sentence|^{i}}$$

The newly introduced parameter *i* determines the exponent of the sentence length and was set to 0, 1 and 2 in the experiments. It is certainly possible to use higher values for *i* and *j* but the results indicated that higher values would not produce better results. Using this notation we can write the three weighting terms as $\text{weight}_{0,\text{j}}$, $\text{weight}_{1,\text{j}}$, and $\text{weight}_{2,\text{j}}$.

The following sections 5 and 6 will give an overview over the experiments that were done using this approach to sort sentences according to their estimated importance. For the first experiments in section 5, we translated English to Spanish. The experiments in section 6 translating Thai to English were done to validate the positive results we saw in the experiments translating English to Spanish.

5 Experiments English-Spanish

5.1 Test and Training Data

The training data for the first translations consisted of 123,416 English sentences with 903,525 English words (tokens). This data is part of the BTEC corpus (Takezawa et al., 2002) with relatively simple sentences from the travel domain. The whole training data was also available in Spanish (852,362 words).

The testing data which was used to measure the machine translation performance consisted of 500 lines of data from the medical domain.

All translations in this task were done translating English to Spanish.

5.2 Machine Translation System

The applied statistical machine translation system uses an online phrase extraction algorithm based on IBM1 lexicon probabilities (Vogel et al., 2003; Vogel et al., 2004). The Language model is a trigram language model with Kneser-Ney-discounting built with the SRI-Toolkit (SRI, 1995-2005) using only the Spanish part of the bilingual training data. This system was also used for the validation experiments translating Thai to English.

We applied the standard metrics introduced for machine translation evaluation, NIST (Doddington, 2001) and BLEU (Papineni et al., 2002).

5.3 Baseline Systems

It was necessary for these experiments to have different baseline systems in order to compare the performance for different training data sizes.

The baseline system that uses all available training data achieved a NIST score of 4.19 [4.03; 4.35] and a BLEU score of 0.141 [0.129; 0.154] (95% confidence intervals).

For the baseline systems, that do not use all available training data, we selected sentences based on the original order of the training corpus. Translation systems trained on this (smaller) data give the scores shown in diagram 1 and 2. The diagrams clearly show that after a rather steep increase of the scores until the translation of approximately 400,000 words, the scores increase only slightly until they reach the final score for the system using all available training data.

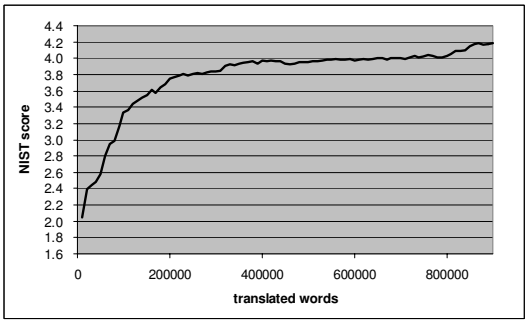

Diagram 1: NIST scores for Baseline systems

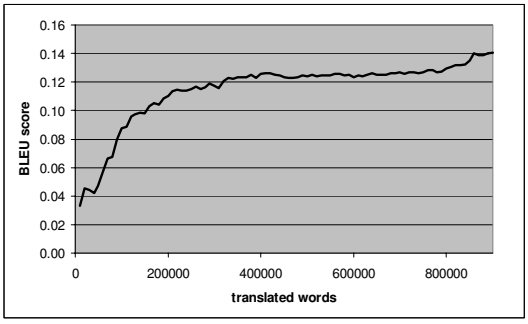

Diagram 2: BLEU scores for Baseline systems

5.4 Coverage Comparison

As our initial weighting terms $weight_{0,1}$, $weight_{0,2}$, and $weight_{0,3}$ focused exclusively on the coverage of uni-, bi- and trigrams, we first examined the actual coverage the sorted sentences would achieve. The diagrams 3 to 6 illustrate the coverage for the sentences in the original order and in the sorted order according to $weight_{0,1}$, $weight_{0,2}$, and $weight_{0,3}$. Please note that the diagrams show the percentage of types covered in the training data, not the percentage of tokens.

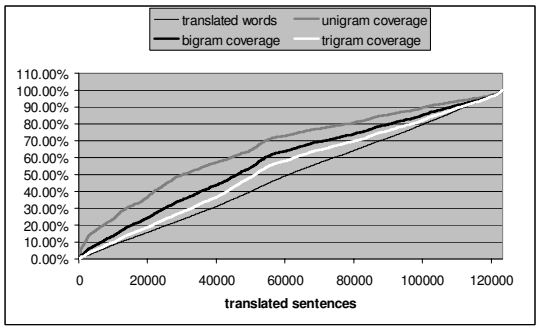

Diagram 3: Coverages for sentences in original order

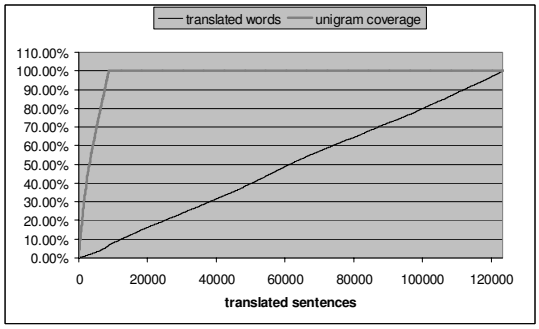

Diagram 4: Unigram coverage for sentences sorted according to $weight_{0,1}$

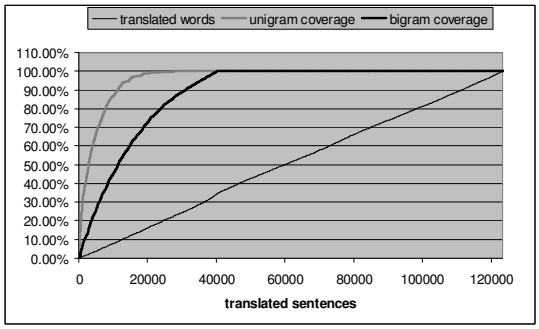

Diagram 5: Uni- and bigram coverage for sentences sorted according to $weight_{0,2}$

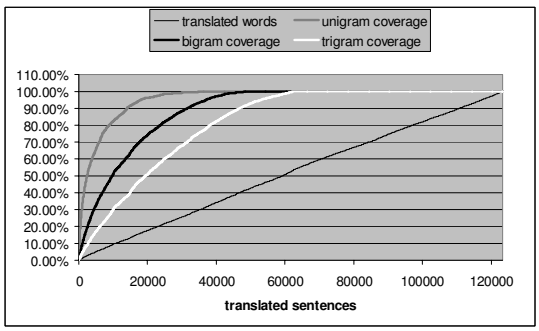

Diagram 6: Uni-, bi- and trigram coverage for sentences sorted according to $weight_{0,3}$

Diagram 3 shows the rather typical behavior of the uni-, bi- and trigram coverage for any training data. Figure 4 shows that it is possible to get 100% coverage of all unigrams in the training data after just translating less than 10% of the sentences and less than 10% of the words. This performance is not really surprising as the weighting term $weight_{0,1}$, that was used to sort the sentences in diagram 4, only focuses on unigrams. The diagrams 5 and 6 illustrate, that similar graphs can be achieved if the weighting term also equally focuses on bi- and trigrams.

5.5 Translation Results

Because of the limited space we will only show graphs for the NIST scores for each experiment. This can be justified as the graphs for the BLEU scores showed basically the same behavior.

Results for term $weight_{0,j}$

Diagram 7 illustrates the NIST scores for systems where the sentences were sorted according to $weight_{0,j}$.

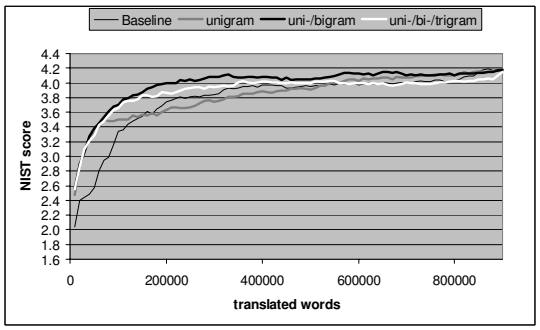

Diagram 7: NIST scores for sentences sorted according to $weight_{0,j}$

If the optimization only uses the number of previously unseen unigrams to rank a sentence, the systems score significantly higher than the baseline for very small amounts of training data. But the steep increase stops very soon, and the systems fall below the baseline but recover towards the end and finish on the same scores. The reason for this pattern is most probably that the optimization achieves a much better coverage for the smaller amounts of training data, but after a while, the baseline system reaches a similar coverage of the testing data and probably has a more meaningful language model with more realistic frequencies.

These problems are clearly fixed by incorporating the bi- and trigrams into the optimization process. The scores no longer fall beyond the scores of the baseline systems but stay consistently higher.

The optimization based on uni- and bigrams generally achieves slightly better scores than the optimization based on uni-, bi- and trigrams (the differences for most data sizes are not statistically significant).

The optimization based on uni- and bigrams reaches a NIST score of 4.0 at 200,000 and a NIST score of 4.1 at 320,000 translated words. A score of 4.0 is only about 5% worse, a score of 4.1 is only about 2% worse than the baseline score of 4.1916 (achieved when training on the whole training data). Scores of 4.1 are already in the confidence interval of the baseline system, so it is highly probable that these systems are not significantly worse than the best baseline system.

Results for term $weight_{1,j}$

The difference between the term $weight_{0,j}$ and $weight_{1,j}$ is the incorporation of the length of a sentence. The number of unseen n-grams is divided by the number of words in this sentence to get the weight for the sentence.

Diagram 8 illustrates the according NIST scores.

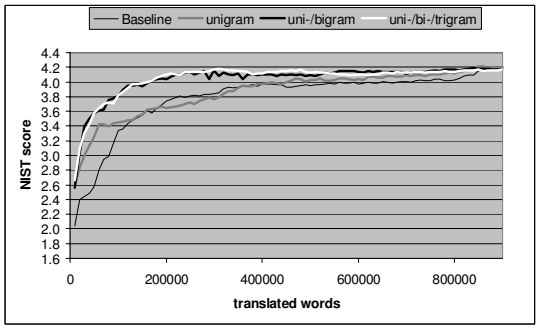

Diagram 8: NIST scores for sentences sorted according to $weight_{1,j}$

A comparison with the Diagram 7 shows that the NIST scores for the sorting of the sentences according to $weight_{1,j}$ are even better than for the term $weight_{0,j}$

The optimization based on unigrams shows a very similar behavior to term $weight_{0,j}$. We notice the same lower scores after about translating 200,000 words and a score recovery towards the end.

The optimizations based on uni- and bigrams and uni-, bi- and trigrams are clearly improved compared to $weight_{0,j}$.

We also do not see any significant differences between the optimization based on uni- plus bigrams and the optimization incorporating

trigrams, too. The performance is very similar with slight advantages for the optimization based on uni- and bigrams only.

In this case a NIST score of 4.0 was already reached at 170,000 translated words while 4.1 was reached at 220,000 translated words.

Results for term weight$_{2,j}$

As explained in section 4.2 we tried to prefer shorter sentences in term weight$_{2,j}$ by dividing the number of unseen n-grams by the square of the number of words in the respective sentence.

Diagram 9 shows that this did not further improve the results achieved using term weight$_{1,j}$ but gave lower scores. The scores stay below 4.1 (NIST) up to over 800,000 translated words.

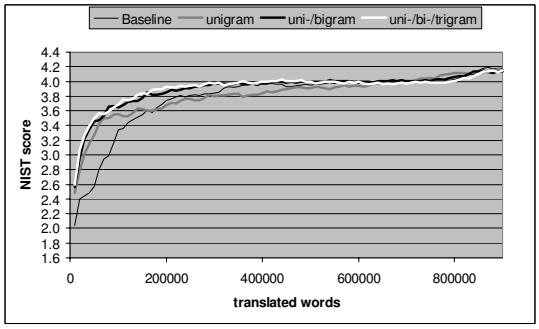

Diagram 9: NIST scores for sentences sorted according to weight$_{2,j}$

Relative Improvements

The maximum relative improvement over a baseline system occurs at 40,000 translated words with 41% in NIST score improvement for the weighting term weight$_{1,2}$, that overall showed the best performance. Diagram 10 gives an overview of the improvements for this term.

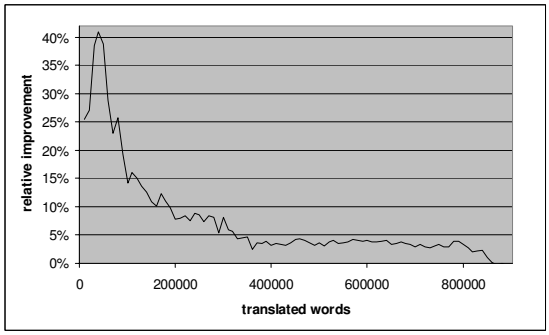

Diagram 10: Relative improvements over the baseline using weight$_{1,2}$ (NIST scores)

One might argue that the improvement of 41% at only 40,000 translated words is irrelevant, as the translations will still be very deficient. This might be the case, but there are applications where even a low quality translation can be helpful (Germann, 2001).

Besides that, some translations are surprisingly good and show considerable improvements over the baseline system (Table 1). The first sentence especially demonstrates the improved coverage; here for the words "heart", "beating" and "normally". The word "beating" is unfortunately not correctly translated but the final result is still much better than the translation of the baseline system.

English source	your heart is beating normally.
Spanish reference	su corazón late normalmente.
Baseline for 40k words	y tu heart está beating normally.
Best system for 40k words	su corazón latía normalmente.
English source	a little bit, but not much.
Spanish reference	un poco, pero no mucho.
Baseline for 40k words	un poco excesivo, pero servirá mucho
Best system for 40k words	un poco, pero no mucho
English source	i have herpes?
Spanish reference	¿tengo herpes?
Baseline for 40k words	podría darme herpes?
Best system for 40k words	tengo herpes?

Table 1: Example translations at 40,000 translated words

6 Experiments Thai-English

We applied the weighting algorithm to another task in order to validate the positive results.

The task here is to translate Thai to English in the medical domain. We used the weighting term weight$_{1,2}$ as it showed the best results in the previous experiments. This means the weight of a sentence is calculated as the number of previously unseen uni- and bigrams divided by the length of the sentence.

6.1 Test and Training Data

The whole training corpus for these experiments had 59,191 sentences with 457,736 English words from the medical domain. The training data was also available in Thai with 422,692 words.

The Test Data consisted of 496 lines, also taken from the medical domain.

6.2 Baseline Systems

Table 2 shows different Baseline scores for these systems. Even with more than 43,000 sentences – more than two-thirds of the whole data – the scores are still 28% (NIST) and 40% (BLEU) lower than if all training data is used.

#sentences	#English words	NIST	BLEU
38,000	306,231	4.30	0.172
43,000	345,773	4.29	0.176
59,191	457,736	5.99	0.294

Table 2: Scores for Baseline systems
Thai – English

6.3 Results

The NIST and BLEU scores in Table 3 clearly indicate that the sorted sentences achieve significantly better results than the baseline systems. The system trained on only 10,000 sentences clearly outperforms the NIST score of the baseline systems trained on 43,000 and 38,000 sentences and reaches an only slightly lower BLEU score.

At 30,000 sentences the NIST score is only 2% lower than the highest score with the BLEU score being only 7% lower.

This shows overall that we can get very similar results on a different task and language pair.

#sentences	#English words	NIST	BLEU
5,000	43,040	4.11	0.104
10,000	82,997	4.84	0.169
20,000	187,595	5.79	0.263
30,000	319,405	5.86	0.274
40,000	395,374	5.92	0.280
59,191	457,736	5.99	0.294

Table 3: Scores for Experiments
Thai - English

7 Future Work

The presented weighting schemes could certainly incorporate other features of the original training data.

It could be useful to include the actual frequency of an n-gram into the weighting of a sentence. Preferring the more frequent words could help because they will cover a higher percentage of the training corpus. On the other hand, less frequent words will have a higher information gain (especially important for the NIST score), so it could also be useful to give higher weight to less frequent words. This will certainly be interesting to investigate.

The presented schemes "try" to cover every n-gram (up to a certain n) once and then do not consider it anymore. It might be helpful to have a goal of covering every n-gram k times to get better estimates of translation probabilities.

Another possible improvement could be to consider the actual coverage situation when weighting the sentences. If the coverage is still very low, it might be more important to cover unigrams (to at least get the words right) while it might be more important to cover bigrams in later stages.

It might be reasonable for some applications to also consider the target language part of the training data when sorting the sentences. This is certainly not possible if the goal is to limit the effort for human translators and the target sentences are not even available at selection time. It could however be incorporated in the selection of training data for small devices because here the translations will already be available.

8 Conclusions

We presented weighting schemes to sort training sentences for statistical machine translation according to their importance for the translation performance. The weighting mainly tries to improve the n-gram coverage while taking the sentence length into account.

The best performance is realized using the number of previously unseen uni- and bigrams and dividing this by the length of the respective sentence.

Using the sorted training data, we were able to achieve similar NIST and BLEU scores with considerably less data.

These results can be used in different applications ranging from low cost portability to translation systems on small devices.

References

Peter E. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. *The mathematics of statistical machine translation: Parameter estimation*. Computational Linguistics, 19(2), pp. 263-311.

Chris Callison-Burch, Colin Bannard, and Josh Schroeder. 2005. *Scaling Phrase-Based Statistical Machine Translation to Larger Corpora and Longer Phrases.* Proceedings of ACL 2005, Ann Arbor, MI, USA.

George Doddington, 2001. *Automatic Evaluation of Machine Translation Quality using n-Gram Coccurrence Statistics*. NIST Washington, DC, USA.

Ulrich Germann. 2001. *Building a Statistical Machine Translation System from Scratch: How Much Bang Can We Expect for the Buck?* Proceedings of the Data-Driven MT Workshop of ACL 2001. Toulouse, France.

Almut Silja Hildebrand, Matthias Eck, Stephan Vogel, and Alex Waibel. 2005. *Adaptation of the Translation Model for Statistical Machine Translation based on Information Retrieval.* Proceedings of EAMT 2005, Budapest, Hungary.

Rebecca Hwa. 2004. *Sample selection for statistical parsing.* Computational Linguistics vol. 30, no. 3.

Teresa. M. Kamm and Gerard G. L. Meyer. 2002. *Selective Sampling of Training Data for Speech Recognition.* Proceedings of HLT 2002, San Diego, CA, USA.

Alon Lavie, Katharina Probst, Erik Peterson, Stephan Vogel, Lori Levin, Ariadna Font-Llitjós, and Jaime Carbonell. 2004. *A Trainable Transfer-based Machine Translation Approach for Languages with Limited Resources*. Proceedings of EAMT 2004, Malta.

Tony McEnery, Paul Baker, Lou Burnard. 2000. *Corpus Resources and Minority Language Engineering.* Proceedings of LREC 2000, Athens, Greece.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. *BLEU: a Method for Automatic Evaluation of Machine Translation*. Proceedings of ACL 2002, Philadelphia, PA, USA.

SRI Speech Technology and Research Laboratory. 1995-2005. *SRI Language Modeling Toolkit*. http://www.speech.sri.com/projects/srilm/

Toshiyuki Takezawa, Eiichiro Sumita, Fumiaki Sugaya, Hirofumi Yamamoto, and Seiichi Yamamoto. 2002. *Toward a Broad-coverage Bilingual Corpus for Speech Translation of Travel Conversation in the Real World.* Proceedings of LREC 2002, Las Palmas, Spain.

Stephan Vogel, Sanjika Hewavitharana, Muntsin Kolss, and Alex Waibel. 2004. *The ISL Statistical Translation System for Spoken Language Translation.* Proceedings of the International Workshop on Spoken Language Translation, Kyoto, Japan.

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

Stephan Vogel, Hermann Ney, and Christoph Tillmann, 1996. *HMM-based Word Alignment in Statistical Translation*. Proceedings of Coling 1996, Copenhagen, Denmark.

Ying Zhang and Stephan Vogel. 2005. *An Efficient Phrase-to-Phrase Alignment Model for Arbitrarily Long Phrase and Large Corpora*. Proceedings of EAMT 2005, Budapest, Hungary.