Rapid Ramp-up for Statistical Machine Translation: Minimal Training for Maximal Coverage

Hemali Majithia, Philip Rennert, and Evelyne Tzoukermann
StreamSage Inc.
1016 16th Street, Washington, DC, 20036.

{majithia,rennert,tzoukermann}@streamsage.com

Abstract

This paper investigates optimal ways to get maximal coverage from minimal input training corpus. In effect, it seems antagonistic to think of minimal input training with a statistical machine translation system. Since statistics work well with repetition and thus capture well highly occurring words, one challenge has been to figure out the optimal number of "new" words that the system needs to be appropriately trained. Additionally, the goal is to minimize the human translation time for training a new language.

In order to account for rapid ramp-up translation, we ran several experiments to figure out the minimal amount of data to obtain optimal translation results.

1 Introduction

When a new language or dialect is suddenly of interest, how can we best train an MT system to translate this new language? Any training corpus in a language of low density language resource will necessarily be small, because however large the English corpus, human translator time to produce material in the new source language will be very limited.

At StreamSage, we have been working on statistical machine translation using publicly available tools. Part of the initial project has been to evaluate an optimal methodology for rapid ramp-up machine translation. For achieving these goals, we ran a series of experiments to determine (a) out of a large parallel corpus – in this case the European Parliament data, what is the minimal amount of training data to build a "mature" system; (b) the effect of morphological analysis on translation, that is if Spanish words are reduced to their lemmas, the translation probabilities should be stronger to determine word mappings; (c) the effect of using a bilingual dictionary to boost word mappings, and (d) finally for a new low density language, how to determine a new training set from the mature system, minimal in word and sentence count, and maximal in coverage of word frequencies. The last item is the objective of this paper.

2 Related Work

Most work on rapid development of MT so far has focused on acquiring large amount of data in the form of resources. In our research, we focus on obtaining an optimal parallel corpus for rapid development of statistical machine translation (SMT) from a large English corpus. We argue that, SMT with less but optimal data will perform as well as a SMT with more data.

Germann et al. [11] address the problem of rapidly building a Statistical Machine Translation system, concluding, that parallel corpus collection is the primary obstacle for the rapid construction of a SMT system. This is this exact reason that motivated our work here.

An approach described by Oard [10] for rapid development of Statistical Machine Translation focuses on the collection of several resources used individually and in combination for training a SMT system. In effect, it is useful to collect maximal resources about a language. In our work, we attempt to maximize our resource selection technique.

Probst et al. [8] discuss a rule-based machine translation system from an elicited corpus trained on transfer rules. The elicited corpus provides minimum number of sentences for the bilingual speaker to translate. This work is similar in this aspect to our selection of a minimal number of sentences. However, we select the optimal sentences on coverage statistics rather than universal language facts. That way, we can quickly build a new training corpus that is dependent on a particular domain. In addition, we use an SMT system, which foresees different problems, such as the statistical acquisition of words.

3 Procedure

In our previous work on the European parliament corpus, about 31,000 bilingual training sentences have been needed to produce "mature" MT system performance. By mature, we mean a MT system capable of producing acceptable translations that are understandable to the user. Note that the European parliament is quite restricted in domain, thus translations are acceptable on the small amount of data. Assuming that a translator can comfortably translate 85 sentences per day[1], five translators working 7 days nonstop will produce one-tenth of the training data needed for mature performance — only 2975 sentences[2]. How, in such a case, should one choose the material for human translation to produce the best resulting translation performance from the trained MT system?

Since the objectives of the experiment consist of coming up with an optimal set of sentences to train a new language pair to be translated, we decided to evaluate the training set using our statistical machine translation (SMT) system [1]. As part of the initial project, we have built a SMT system for Spanish using publicly available tools. The system has been constructed using the GIZA++ [2, 3] toolkit, and parallel corpora are trained on the IBM-1, HMM, and IBM-4 models for 5 iterations each. The ISI-Rewrite Decoder [4, 5] and the CMU Language Modeling Toolkit [6] were used to generate translations. As said earlier, the goal is to determine how to choose the training sentences that would yield optimal MT system performance at each stage of rapid ramp-up. Therefore, the algorithm consists in the following steps:

1. Start with a vocabulary of 31,000 sentences. This corresponds to 900,000 tokens, or 19,000 words.
2. Select an initial set of sentences.
3. Translate and score the initial sentences using the test corpus.
4. Select next set of sentences.
5. Repeat steps 3 through 5.

What is the optimal next set of sentences that should be selected is the challenge that is posed here. Several trials have been performed using a fixed number of sentences, after which the MT system was trained on. For each trial, a fixed number of sentences were chosen (approximately 500 sentences) by the method being tested. The MT system was then trained on those sentences. For comparison, the MT was also trained on two special cases: first, all 31,000 sentences in our training corpus, representing the gold standard of training to mature MT system performance. Second, random sentences totalling to about the same number of words as the training data being tested, representing an unsophisticated choice of material for human translation.

Results from the trained MT system under these various conditions were evaluated with NIST (http://www.nist.gov/speech/tests/mt/resources/scoring.htm) and BLEU [7], and METEOR [9] scores. We considered only set of one reference translations to scores our MT system. BLEU averages the precision for n-grams. NIST is very similar to BLEU but instead of n-gram precision, the information gain from each n-gram is taken into account. METEOR scores the translation using unigram precision and recall. NIST and METEOR scores were fairly similar even though METEOR puts more emphasis on unigram and bigram matching. BLEU assigns a zero score when no 4-gram exactly matched the human translated sentence, which occurred often enough in our ramp-up situation to significantly bias results. We would like a less restricted evaluation metric and hence prefer the METEOR scores for comparison.

Data

The proceedings of the European Parliament (http://people.csail.mit.edu/koehn/publications/europarl/) served as an experimental corpus, and we focused with Spanish as the simulated “new language”.

The European parliament proceedings from January 17, 2000 through March 30, 2000 provided a total of 31,000 sentences, 900,858 tokens, 18,924 unique words in English and 938,305 tokens, 29,550 unique words in Spanish as our potential training set. As a holdout test set we randomly selected 1000 sentences of the proceedings from April 15, 1996 through December 17, 2001. This corresponded to 27806 tokens and 4399 unique words for English and 28892 tokens and 5280 unique words for Spanish. As test sentences are drawn from a wider time span than training sentences, there may be small effects from differences in content. As well, it is possible that a small number of the test sentences may also be in the training corpus.

[1] Translating 2500 words per day, at an average of 30 words per sentence, comes to 83.33 sentences per day.

[2] Even with extensive financial resources, a large pool of translators may simply be unavailable for a low density language.

4 Experiments and Results

The research tested a number of methods for choosing sentences from the 31,000 candidates in the potential training corpus. Among these methods, we:

a. Maximize the number of unique words. We explored training sentences based on a one-time occurrence of the word,
b. Choose sentences on average length, various discounting value functions and bigram frequency.
c. Perform on-line learning algorithms (choosing the sentences next on which the current translator did worst). We tackled the issue of words that were incorrectly translated by selecting sentences that did not produce a good alignment score.
d. Select support sentences to include the new words in a frequency-chosen sentence. Also use translation and fertility probabilities to detect hardness of words in the sentence and select variable support sentences for the sentence in question.
e. Choose sentences based on maximizing discounted average word frequency.

Interestingly, the first three techniques did not yield significant results. Though the fourth approach was pursued in detail, it didn't work quite as well as the fifth.

Sentences were selected in two steps:

1. choose top frequency sentences based on high average English word frequency;
2. augment first set in choosing support sentences for difficult words. A support sentence is a sentence, which provides a given word enough contextual information for the system to train.

How many support sentences does the system require to learn a new word translation? This has been the focus of the work in this section.

4.1 Selecting the top 2000 sentences using high average word frequency

We began by choosing words in sentences with a word discount parameter from the English corpus of 31,000 sentences. A parameter study as shown in Table 1.1 allowed us to compare different discount measures and we pursued the evaluation with a discount of 0.7. Table 1.2 shows the matching statistics of number of sentences and words in English and Spanish as we change the frequency discount parameter.

Initially, we scored sentences according to the average frequency of the words in them. We soon found we were getting short sentences full of stop words. So we applied a discount parameter to down weight a word for each previous occurrence, by adjusting its frequency in the average. For example, under discount parameter of 0.7, if a word had 10,000 occurrences in the corpus but appeared once in the previously chosen sentences, the current sentence was scored as if the word's frequency were 10,000*0.7=7000; if it appeared twice previously, then 4900; and so on. That way, although the first sentences chosen were full of common stopwords, these words were soon discounted to the point that sentences were chosen containing a much greater variety of words.

Second, to weight the algorithm towards choosing longer sentences, in taking the average we divided the word frequency by the number of words plus a constant (which we set equal to 5) — penalizing short sentences over long ones. With this modification, the average length of the training sentences chosen by this algorithm was now close to the average length of sentences across the corpus.

Training set	MT Evaluation Score		
	NIST	BLEU	METEOR
discount = **0.5**	**5.0482**	**0.1635**	**0.3750**
discount = **0.6**	**5.0655**	**0.1655**	**0.3771**
discount = **0.7**	**5.1544**	**0.1717**	**0.3869**
discount = **0.8**	**5.2074**	**0.1729**	**0.3913**
discount = **0.9**	**5.1698**	**0.1682**	**0.3851**

Table 1.1: Evaluation Scores for various Word-frequency discounts

2500 chosen sentences, discount **0.5** **English:** 57690 words, 7820 unique **Spanish:** 63646 words, 9605 unique
2420 chosen sentences, discount **0.6** **English:** 57783 words, 7248 unique **Spanish:** 63936 words, 9245 unique
2280 chosen sentences, discount **0.7** **English:** 57547 words, 6546 unique **Spanish:** 63693 words, 8808 unique
2175 chosen sentences, discount **0.8** **English:** 57709 words, 5910 unique **Spanis**h: **63710 words, 8269 unique**
2000 chosen sentences, discount **0.9** **English:** 57560 words, 5028 unique **Spanish:** 63396 words, 7462 unique

Table 1.2: **Corpus Statistics for various Word-frequency discounts**

4.2 Results for the top 2000 sentences selection

We compared the top 2000 chosen sentences with the highest average discounted word frequency to random training sentences, adjusting the number of random sentences to match the number of words in the set of chosen training sentences. As shown in the Table 2.1, the chosen sentences with word-frequency discount of 0.7 perform better than random sentences. Table 2.2 shows the number of sentences and words in our training set. Notice that we maintain consistency in the number of words between chosen and random sets.

	MT Evaluation Score		
Training set	**NIST**	**BLEU**	**METEOR**
random	**4.8105**	**0.1564**	**0.3643**
chosen	**4.9957**	**0.1628**	**0.3793**
mature	**6.1596**	**0.2334**	**0.4803**

Table 2.1: Evaluation Scores for Word-frequency discount of 0.7

2000 chosen sentences **English:** 50212 words, 6032 unique **Spanish**: 55611 words, 8078 unique
1770 random sentences **English:** 50413 words, 5582 unique **Spanish**: 52465 words, 7091 unique

Table 2.2: Corpus Statistics for Word-frequency discount of 0.7

4.3 Iterative selection of the next sentence set

In contrast to the previous algorithm, which builds a single global ranking of training sentences, we considered iterative improvements to sentence selection and compared them with the approach of choosing sentences by discounted average word frequency.

As in on-line learning algorithms, we first tried to iteratively add those sentences to the training set which the MT system translated the worst. To test the iterative approach, we chose 2000 initial training sentences by the above high average word frequency algorithm. Then we added the sentences in the test data on which the then-best translator scored lowest. To our surprise, this did not help. In fact, MT performance improved more from training on the best-translated test sentences than from training on the worst. From these results we hypothesize that when a translated sentence contains many chunks of new information, the SMT system has difficulty discriminating among them; if a training sentence contains only a moderate amount of new information, the system is more able to learn from it.

4.3.1 Support sentences for new and less frequent words

Then, we considered that in teaching language to a human, new and less frequent words are presented in a variety of contexts. By analogy, we can create training data, which *supports* new words. For a training sentence chosen by high average word frequency, we added sentences, which contain additional uses of the new and less frequent words. This training should improve SMT performance by assisting learning of new words as described below:

To these existing sets, we add additional training sentences in sets, as follows:

1) Choose a training sentence by the high average word frequency algorithm. Call this sentence *S*.
2) Weight all the words in *S*:
   - if there is no occurrence of the word in previous training sentences, assign to this word a weight of 6;
   - if there is one occurrence, assign weight of 5;
   - if there are two occurrences, assign weight of 4; and so on until weight of 1.

   This detects the new and less frequent words in *S*, weighting most highly those least represented in the training data so far.
3) Score all other potential training sentences with their weighted density of these uncommon words in *S*. Density is the weighted total number of occurrences, divided by the length of the sentence. Scoring by density, rather than just the weighted total, avoids a bias towards long sentences.
4) Choose the *h* highest scoring of these sentences, and add them to the training data, along with *S*.

   Trials were run with *h* equal to 2, 3, and 4 support sentences, thus creating training sentences in sets of 3, 4, and 5.

4.3.1.1 Results for Support sentences for new and less frequent words

For NIST, BLEU, and METEOR scores, support sentence sets perform better than random sentences. According to METEOR, the set of 2 support sentences performs better than the set of 3 support sentences as shown in Table 3.1. NIST and BLEU, on the other hand, show better performance

with 3 support sentences. Table 3.2 shows the numbers of training sentences and words in the training set for different sets of support sentences. Again, the attention is put here on the consistency between the number of words across the different training sets.

Training set	MT Evaluation Score		
	NIST	BLEU	METEOR
3 supports	**5.1625**	**0.1701**	**0.3849**
2 supports	**5.1546**	**0.1687**	**0.3874**
variable support	**5.1490**	**0.1682**	**0.3881**
chosen	**5.1813**	**0.1723**	**0.3893**
random	**5.0244**	**0.1685**	**0.3723**

Table 3.1: Evaluation Scores for Word-frequency discount of 0.7 with support sentences

2500 sentences with 3 supports (2000 plus 125 sets of 4) **English**: 59585 words, 6495 unique **Spanish**: 65650 words, 8824 unique
2500 sentences with 2 supports (2000 plus 166 sets of 3) **English**: 59604 words, 6496 unique **Spanish**: 65676 words, 8827 unique
2500 sentences with variable support (2000 plus 500 variable support) **English**: 59689 words, 6493 unique **Spanish**: 65700 words, 8815 unique
2350 chosen sentences (high average word frequency) **English**: 59286 words, 6669 unique **Spanish**: 65589 words, 8983 unique
2100 random sentences **English**: 59590 words, 6056 unique **Spanish**: 61933 words, 7809 unique

Table 3.2: Corpus Statistics for Word-frequency discount of 0.7 with support sentences

4.3.2 Support sentences for difficult words

Our final iterative approach to training sentence choice is an extension to the previous idea. In human language learning, some less frequent words require extra support. *Difficult* words — particularly those with many meanings — must be learned many times in a variety of contexts. This final algorithm, then, adds sentences in variable sets instead of fixed ones. Rather than adding a fixed number of support sentences, the algorithm attempts to add more support sentences for sentences that contain less frequent words that are also more difficult. We added support for difficult *less frequent* words — those that appear 1 through 5 times in the training sentences so far. We cannot calculate the difficulty of new words in the sentence since the MT system has not yet encountered them.

4.3.2.1 Support sentences for difficult words — a difficulty measure

To support difficult words, we first must detect them: we need a difficulty measure.

Our difficulty measure is based on *t* (translation probability) and *n* (fertility) scores of a translated word. For each possible alignment of a source word, *t* measures how likely that translation is. An easy word, with only few possible alignments, should have a relatively high maximum *t* score, maximized over all possible translations for that word. Conversely, a difficult word, with many possible translations, should have a low maximum *t*.

For every possible number of target words that a source word may be translated into, the fertility score *n*, for that word and that number, measures the probability of a translation into that number of words. Just as with *t*, an easy source word should translate into only a few possible patterns, and thus only a few possible numbers of target words, resulting in a high maximum *n* score. Conversely, a difficult source word will tend to have a low maximum *n*.

We normalized both t and n by measuring them in standard deviations from the mean. Summing these normalized *t* and *n* measures, we arrive at a total difficulty score, $d = t + n$, for each word: low *d* indicates high difficulty; high *d* indicates an easy word to translate.

4.3.2.2 Support sentences for difficult words — the algorithm

With an initial training set of 2000 sentences chosen by the high average word frequency algorithm, we added additional training sentences in variable-size sets, as follows:

1) Choose a training sentence by the high average word frequency algorithm. Call this sentence *S*.
2) Weight all the words in *S*:
   - if there is no occurrence of the word in previous training sentences, assign to this word a weight of 6;

- if there is one occurrence, assign weight of 5;
- if there are two occurrences, assign weight of 4; and so on until weight of 1.

This detects the new and less frequent words in *S*, weighting most highly those least represented in the training data so far.

3) Score all other potential training sentences with their weighted density of these less frequent words in *S*.

So far, this is identical to the previous algorithm, for finding support sentences for new and less frequent words. The next step differs since we use the difficulties of the less frequent words in *S* to determine how many support sentences to add:

4) *d* is the difficulty of less frequent word in the sentence *S*, and *D* is the sum of all the *d*'s. *D* is measured in standard deviations above or below the mean.
5) Choose the *h* highest scoring of the potential support sentences. To add more support for the sentences containing more difficult words, *h* is determined by *D*. At this point, there are two conditions to estimate: (a) the number of less frequent words, and (b) the difficulty of the less frequent words. Support sentences are selected based on these conditions.
6) Choose the *h* highest scoring of these sentences, and add them to the training data, along with *S*.

Using this algorithm, a single word-frequency-chosen sentence will receive a variable number of support sentences, depending on how many less frequent words it contains.

4.3.3 Results for all support sentences algorithms

Fixed support sentence sets do better than random sentences as we have seen in the previous results according to METEOR. By using variable-size sets of support sentences for difficult and less frequent words, we further see an improvement on the translations. The MT evaluation scores for variable-size of support sentences can be seen in Table 3.1 and in the corresponding statistics in Table 3.2.

5 Conclusion and Future Work

Sentences chosen by maximizing discounted average word frequency produced better SMT results than sentences chosen randomly. Selecting sentences by this algorithm is computationally quick and simple.

From the iterative algorithms results, adding support sentences for word-frequency-chosen sentences produces better results than random sentences. Preliminary results indicate that by choosing support sentences based on "new and less frequent words", there is an improvement to the SMT training data set.

This model can be very easily duplicated with any vocabulary of any language over any domain. What has been particularly interesting in our research has been to figure out the number of occurrences a "new" word or newly acquired word needed to be learned or trained by the system. We found indications that seeing a word 3 times in different contexts was most cost-efficient for system learning. And varying the number of times a word is seen according to its difficulty was slightly more cost-efficient than holding it fixed.

We plan to apply these techniques to a new language and on a domain other than the European parliament.

References

1. Peter E. Brown Vincent J. Della Pietra. The Mathematics of Statistical Machine Translation: Parameter Estimation. Computational Linguistics 19(2), 263-311, 1993.

2. Franz Josef Och, Hermann Ney. "Improved Statistical Alignment Models". *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, pp. 440-447, Hongkong, China,* October 2000.

3. Franz Josef Och: "An Efficient Method for Determining Bilingual Word Classes"; *pp. 71-76, Ninth Conf. of the European Chapter of the Association for Computational Linguistics; EACL'99, Bergen, Norway,* June 1999.

4. Ulrich Germann, Jahr, M., Knight, K., Marcu, D., and Yamada, K. Fast Decoding and Optimal Decoding for Machine Translation. *Proceedings of ACL-01.* Toulouse, France 2001.

5. Ulrich Germann. Greedy Decoding for Statistical Machine Translation in Almost Linear Time. Proceedings of HLT-NAACL, Edmonton, AB, Canada, 2003.

6. R. Clarkson and R. Rosenfeld. Statistical Language Modeling Using the CMU-Cambridge Toolkit. *From Proceedings ESCA Eurospeech* 1997.

7. Kishore A. Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. *Technical Report RCC22176 (W0109-022), IBM Research Division, Thomas J. Watson*

Research Center, YorkTown Heights, NY, September, 2003.

8. Katharina Probst, Ralf Brown, Jaime Carbonell, Alon Lavie, Lori Levin, and Erik Peterson. Design and implementation of controlled elicitation for machine translation of low-density languages. *Workshop MT2010 at Machine Translation Summit VIII,* 2001.

9. Lavie A., K. Sagae, and S. Jayaraman. The Significance of Recall in Automatic Metrics for MT Evaluation. *Proceedings of the 6th Conference of the Association for Machine Translation in the Americas* (AMTA-2004), Washington, DC, 2003. http://www2.cs.cmu.edu/~alavie/METEOR/

10. W. Oard, and Franz Josef Och. Rapid-Response Machine Translation for Unexpected Languages. *Proceedings of the MT Summit IX*, 2003.

11. Ulrich Germann. Building a statistical machine translation system from scratch: How much bang for the bucks can we expect? *In ACL 2001 Workshop on Data-Driven Machine Translation*, Toulouse, France, 2003.