Data Selection and Smoothing in an Open-Source System for the 2008 NIST Machine Translation Evaluation

Holger Schwenk, Yannick Estève

Laboratoire d’Informatique du Maine (LIUM), University of Le Mans
72085 Le Mans cedex 9, FRANCE

{schenk,esteve}@lium.univ-lemans.fr

Abstract

This paper gives a detailed description of a statistical machine translation system developed for the 2008 NIST open MT evaluation. The system is based on the open source toolkit Moses with extensions for language model rescoring in a second pass. Significant improvements were obtained with data selection methods for the language and translation model. An improvement of more than 1 point BLEU on the test set was achieved by a continuous space language model which performs the probability estimation with a neural network. The described system has achieved a very good ranking in the 2008 NIST open MT evaluation.

Index Terms: statistical machine translation, continuous space language model, open source, NIST evaluation

1. Introduction

Statistical machine translation (SMT) is today considered as a serious alternative to rule-based approaches. These systems are usually very well ranked in international evaluation campaigns and some commercial products are based on this technology. The availability of the open source SMT toolkit Moses [1] was a major factor to advance the field, enabling researchers and companies to easily build and use a SMT system. Although all the necessary software is provided in the Moses toolkit it is however not straightforward to build a large state-of-the-art SMT system. In this paper we give a detailed description of such a system that was ranked very well in the 2008 NIST open MT evaluation.

The system was designed with the main characteristic that it can be trained and used on a single off-the-self machine. The maximum memory usage is about 27GB during n-best list generation, the filtered phrase-table and the language model being loaded into memory. This could be substantially reduced by keeping the translation model on disk [2] or using quantized language models [3]. Although this usage exceeds the standard memory size, servers of this type can easily be bought for a few thousand dollars. During the last years, distributed language models are getting more and more popular, see for instance [4], but we decided to do all processing on a single machine since we believe that this is more realistic for many stand-alone applications. There is also increasing interest in system combination of several SMT systems, but this was voluntary not used either since it necessarily leads to longer processing times.

Special attention was given to a careful and efficient use of the available data. We try to give some insight in the compromise of quality versus quantity, i.e. do we rather need a limited amount of good data or is it more important to have large amounts of any data? It was also important to us to build compact representations of the statistical models. The last feature of the developed system is a continuous space language model that performs the probability estimation in a continuous space.

2. Task Description

NIST is administering open international evaluations of automatic machine translation technology since 2001. These evaluations have played a key role to advance the field by providing a common test bed and infrastructure to compare the most promising approaches. Starting with the translation of Mandarin to English, four different translation directions are of interest today. Two types of evaluations are distinguished: constrained, i.e. the allowable resources are limited to a list of LDC corpora, and unconstrained, i.e. any publicly available data may be used. In this paper we report results with our constrained Arabic/English system. The official results of all participants have been published by NIST.1

The allowable resources in the constrained track include about 175M words of parallel data (bitexes) and several billions words of English texts to train the target language model. NIST also provides development and test data from the previous evaluation campaigns. It was known that the test set of the 2008 evaluation would contain about 20.000 words of news wire text and 20.000 words of WEB or newsgroup data. Therefore we decided to develop and optimise our system on the 2006 test set, the only one that also contains WEB data. The older development and test sets were added to the training material. The official metric is the BLEU score calculated with four human references. A human comprehension test is also underway [5], but the complete results are not yet available.

3. System Architecture

The goal of statistical machine translation is to produce a target sentence e from a source sentence f. It is today common practice to use phrases as translation units [6, 7] and a log linear framework in order to introduce several models explaining the

1http://www.nist.gov/speech/tests/mt/2008/
translation process:

\[ e^* = \arg \max_e p(e|f) = \arg \max_e p(f, e)P(e) = \arg \max_e \{\exp(\sum_i \lambda_i h_i(e, f))\} \quad (1) \]

The feature functions \( h_i \) are the system models and the \( \lambda_i \) weights are typically optimised to maximize a scoring function on a development set [8]. In our system fourteen features functions were used, namely phrase and lexical translation probabilities in both directions, seven features for the lexicalized distortion model, a word and a phrase penalty and a target language model (LM).

The system is based on the Moses SMT toolkit [1] and constructed as follows. First, Giza++ is used to perform word alignments in both directions. Second, phrases and lexical reorderings are extracted. Both steps use the default settings of the Moses SMT toolkit. A 4-gram back-off target LM is then constructed as detailed in the next section. The translation itself is performed in two passes: first, Moses is run and a 1000-best list is generated for each sentence. The parameters of this first pass are tuned on Dev06 using the cmert tool. These 1000-best lists are then rescored with a continuous space 5-gram LM and the weights of the feature functions are again optimised using the open source numerical optimisation toolkit Condor [9].

The details of this optimisation procedure are as follows:

1. The \( n \)-best lists are reranked using the current set of weights. A hypothesis is extracted and scored against the reference translations of the development data.
2. The obtained BLEU score is passed to Condor, which either computes a new set of weights (the algorithm then proceeds to step 1) or detects that a local maximum has been reached and the algorithm stops iterating.

It is stressed that Moses and the CSLM are only run once and that the whole second pass tuning operates on \( n \)-best lists. This usually takes a couple of hours, most of the time being used by the NIST scoring tool. This basic architecture of the system, summarized in Figure 1, is identical to the one previously used in a medium-sized task [11].

### 3.1. Target language model

The first source of LM training data is the English part of the bitexts. However, these texts come from many different sources, some of them being rather small but nevertheless important. Building one LM on the pooled data bears the risk that large amounts of general data, e.g. the UN texts, dominate over important but limited in-domain data. Therefore, six source specific LMs were built and interpolated, namely translations obtained from the Gale program (1.1M words), various newspaper wire data (8.1M words), the ISI automatically extracted parallel texts (35M words), the development sets from the previous years (0.9M words) and the UN data (130M words). For all these texts specific LMs were built using modifit Kneser-Ney smoothing and a cut-off of 2 for the 4-grams. For the larger UN data a cut-off of 2 for the 3-gram and of 3 for the 4-grams was used in order to keep the size of the LM reasonable. We found no evidence that keeping more n-grams improves the quality of the LM, at least as measured by the perplexity. It can be seen from Table 1, first three lines, that the interpolated LM has a much lower perplexity than the pooled LM, in particular for the

\[ \text{WEB part. Note that the pooled LM is larger since a cutoff of 2 on the 4-grams was applied for all data, including the UN texts. It is also striking that adding the UN data actually worsens the results of the pooled LM. LDC also provides a large collection of newspaper texts, known as the Gigaword corpus. Adding 3.4G words from this collection reduces the perplexity by about 21%, equally for the news wire and WEB parts of Dev06. Again, seven separate LMs were built and interpolated using the same cut-offs than for the UN data.}

The last resource used for language modeling consists in a collection of 1 billion 5-grams, obtained from texts collected by Google on the Internet. It has been reported that significant improvements of the performance of an SMT system can be obtained using all this data [4], but this requires substantial hardware equipment that exceeded by far our possibilities. Therefore, we applied various steps of filtering with the goal to keep only the most important \( n \)-grams. First of all, we extracted all unique 4-grams, ignoring 4-grams that contain either

<table>
<thead>
<tr>
<th>Corpus</th>
<th>train</th>
<th>LM</th>
<th>Perplexity Dev06</th>
</tr>
</thead>
<tbody>
<tr>
<td>bitexts pooled</td>
<td>175M</td>
<td>666M</td>
<td>189.3</td>
</tr>
<tr>
<td>idem w/o UN</td>
<td>45M</td>
<td>278M</td>
<td>183.0</td>
</tr>
<tr>
<td>bitexts ipol.</td>
<td>175M</td>
<td>309M</td>
<td>161.7</td>
</tr>
<tr>
<td>+ Gigaword</td>
<td>3.4G</td>
<td>3.7G</td>
<td>128.1</td>
</tr>
<tr>
<td>+ Google</td>
<td>169M</td>
<td>5.5G</td>
<td>114.5</td>
</tr>
<tr>
<td>+ CSLM</td>
<td>3.4G</td>
<td>≈ 1G</td>
<td>98.3</td>
</tr>
</tbody>
</table>

Table 1: Characteristics of various LMs. The perplexity is given for the total Dev06 corpus, as well as separately for the news wire and WEB part. LM size corresponds to the file size on disk in binary SRILM format.
unknown words or special characters like @ or #. We then applied a cut-off of 500. Finally, we only kept 4-grams that contain bigrams which are also observed in the WEB part of our development corpus. The remaining 139M 4-grams were used to build our “Google LM”. Adding this LM reduced the perplexity by about 9% in average, but most of the improvement is obtained on the WEB part of Dev06, which is actually not surprising given the source of this data (see Table 1 fifth line). The final LM that was used with the Moses decoder was obtained by interpolating all the specific LMs (5 LMs trained on the bitexts, 7 Gigaword LMs and 1 Google LM). It is composed of roughly 62M bigrams, 122M trigrams and 264M 4-grams, and its file size is about 5.5G using the binary representation of the SRI LM toolkit [12].

We also used a so-called continuous space language model (CSLM). The basic idea of this approach is to project the word indices onto a continuous space and to use a probability estimator operating on this space [13, 14]. Since the resulting probability functions are smooth functions of the word representation, better generalization to unknown n-grams can be expected. A neural network can be used to simultaneously learn the projection of the words onto the continuous space and to estimate the n-gram probabilities. The inputs to the neural network are the indices of the n−1 previous words in the vocabulary $h_i = w_j, w_{j+1}, \ldots, w_{j-n+1}$ and the outputs are the posterior probabilities of all words: $P(w_j=i|h_i)\forall i \in \{1, N\}$ where $N$ is the size of the vocabulary. Three layers are used: a projection layer, a sigmoidal hidden layer and a softmax output layer. The value of the i-th output neuron corresponds directly to the probability $P(w_j=i|h_i)$. Training is performed with the standard back-propagation algorithm minimizing the error function:

$$E = \sum \limits_{i=1}^{N} t_i \log P(w_j=i|h_i) + \beta \left( \sum \limits_{ij} w_{ij}^2 \right) \quad (2)$$

where $t_i$ denotes the desired output and $w_{ij}$ the network weights. The first part of this equation is the cross-entropy between the output and the target probability distributions, and the second part is weight decay term.

The CSLM is still a n-gram approach, but the LM probabilities are “interpolated” for any possible context $h_i$ instead of back-off to shorter contexts. This approach was already successfully used in small or medium-size phrase-based SMT systems [10, 11]. Here, we use it the first time on large amounts of data, almost 3.5G words, using a resampling algorithm [15]. The CSLM achieved an additional perplexity reduction of almost 14% on top of the large LM which includes the filtered Google data. The storage requirements of the CSLM are also lower since it uses a distributed representation of the knowledge.

3.2. Translation model

The findings of the previous section have shown that the different sources of the bitexts have not the same importance and that they should be weighted accordingly. Unfortunately, this is not straightforward to perform for the translation model. Limited experiments with multiple phrase tables were not concluding. This approach has the major disadvantage that the number of feature functions increases rapidly, five feature functions being used by Moses for each phrase table. Running GIZA on small bitexts may also lead to suboptimal alignments. Therefore the alignment and phrase extraction was performed with the whole bitexts of 165M words. The total processing took about 84h and resulted in a phrase table with 228M entries and a compressed file size of 6.2G. An analysis of the importance of each source was performed. Table 2 compares the BLEU scores in function of the size of the bitexts, using an LM trained on the English side of the bitexts and the Gigaword corpus. Using only the GALE\(^1\) and news wire\(^4\) bitexts, that is only 9M words in total, and remarkable BLEU score of 43.02 was obtained.

![Figure 2: BLEU scores on Dev04 and Test06 when using the GALE bitexts and all sentences of the ISI parallel texts with a confidence score superior to the threshold.](image)

Another important source are the automatically extracted parallel text prepared by ISI [16] and distributed by LDC. This corpus contains 1.1M sentence pairs (about 35M words) which were automatically extracted and aligned from the monolingual Arabic and English Gigaword corpora, a confidence score being provided for each sentence pair. Figure 2 plots the BLEU score as a function of this confidence score: only sentences with a confidence score superior to the threshold are kept. These experiments were performed in a preliminary stage of system development using the GALE and filtered ISI bitexts only. The coefficients of the feature functions were tuned for each run.

<table>
<thead>
<tr>
<th>Bitext</th>
<th>#words</th>
<th>Dev06</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gale+nw</td>
<td>9M</td>
<td>43.02</td>
</tr>
<tr>
<td>Gale+nw+ISI</td>
<td>35M</td>
<td>45.09</td>
</tr>
<tr>
<td>Gale+nw+ISI+dev</td>
<td>36M</td>
<td>45.38</td>
</tr>
<tr>
<td>Gale+nw+ISI+dev+un</td>
<td>165M</td>
<td>45.98</td>
</tr>
</tbody>
</table>

Table 2: BLEU scores in function of size of the bitexts.

It turned out to be critical to not use all the automatically aligned sentences. The optimal threshold was determined at 0.94 (25.7M words). This is almost two points BLEU better than keeping more sentences (threshold of 0.9, 27.0M words). Restricting furthermore the threshold, on the other hand, worsens the results since the number of words decreases considerably (7.9M words for 0.99). The optimal value of the threshold may of course change once all the bitexts are included, but running these experiments with more than 100M words of bitexts is computationally expensive.

A very competitive system with a BLEU score of 45.38 on Dev06 can be built by using only 36M words of bitexts: the

---

\(^1\)LDC2005E83, 2006E24, E34, E85 and E92

\(^4\)LDC2005T07, 2004E72, T17, T18, E46 and 2006E25
GALE, news wire, filtered ISI bitexts and the previous development data (third line of Table 2). Adding finally the UN bitext, that is 130M words, has a relatively small impact on the performance of the SMT system: the BLEU score “only” increases by 0.6. This may be seen as evidence that small amounts of high quality in-domain data seems to be as important as large amounts of general bitexts. Note that we use less bitexts than LM data because of the filtering of the ISI data (165M versus 175M). This baseline system was further improved by optimising the pruning settings of the Moses decoder and adding the Google and the CSLM respectively (see Table 3).

<table>
<thead>
<tr>
<th>System</th>
<th>Dev06 NW</th>
<th>Eval08 All</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>43.99</td>
<td>41.69</td>
</tr>
<tr>
<td>beam tuning</td>
<td>44.40</td>
<td>42.13</td>
</tr>
<tr>
<td>+ Google LM</td>
<td>44.70</td>
<td>41.90</td>
</tr>
<tr>
<td>+ CSLM</td>
<td>45.96</td>
<td>42.98</td>
</tr>
</tbody>
</table>

Table 3: Performance of the final system.

The optimisation of the decoding parameters achieved an improvement of 0.4 BLEU, uniformly for all conditions and test sets. The Google LM brought 0.3 BLEU on the Dev data, all the improvements being concentrated to the WEB part. Unfortunately, it turned out to be not useful on the Eval08 test data. An unexpected and very satisfactory result is the good performance of the continuous space LM. It achieved an improvement of 1.1 BLEU on the test data, on top of this heavily optimised system. This final system achieved a very good ranking in the 2008 NIST MT evaluation. The official results of all participants have been published by NIST.1

4. Conclusion

This paper provided a detailed description of an Arabic/English statistical machine translation system that was very well ranked in the 2008 NIST Open MT evaluation. This system is based on the open source toolkit Moses, completed by software to perform rescoring with a continuous space language model in a second pass. Currently there seems to be a tendency to use extremely large language models operated on distributed LM servers, and to combine several independent machine translation systems. Both techniques are reported to improve translation quality but they usually increase the computational complexity or ask for dedicated equipment (cluster of several machines). The system described in this paper achieves state-of-the-art performance on a well established task without using these techniques.

Another important finding of this papers is the fact that the careful selection and weighting of the available training data may have an important impact on the quality of the translations. This can be done in an efficient way for the target language model by creating source specific models, estimating automatically the interpolation coefficients and merging all models into one. Unfortunately, a similar procedure is not yet known for the translation model.

Another promising technique is to increase the amount of parallel texts by extracting automatically (almost) parallel sentences from large monolingual collections. Such a corpus was provided by ISI/LDC [16]. However, it turned out to be crucial to carefully select a subset of all provided sentences based on a confidence score. We obtained larger improvements in the BLEU score by adding this data than using the UN corpus. The latter one contains four times more words, but it is less related to the task. We are currently working on extensions of the algorithm proposed by Munteanu et al. [16] to automatically extract additional parallel texts.

Finally, this work confirmed the good performance of the continuous space language model. This approach was initially developed to improve the estimation of unseen n-grams when only a very limited amount of in-domain training data is available. Here, more than 3.5 billion words of text were used. Nevertheless, a significant improvement of more than one point BLEU on the test data was achieved, with respect to an already heavily optimised system. This work has been partially funded by the French Government under the project INSTAR (ANR JCJC06_143038).

5. References