Comparable Corpora in Wikipedia Text for Machine Translation

Jia Xu[+], Casey Kennington[*], Česlav Przywara[*], Lilian Wanzare[*]

[+]Institute for Interdisciplinary Information Sciences,
Tsinghua University, Beijing,100084, P.R.China
[*]Department of Computational Linguistics
Saarland University, D-66123, Saarbrücken, Germany
{xujjia,bakuzen,ceslav.przywara,liliwanzie}@gmail.com

December 13, 2011

Abstract

Machine translation systems need training data in order to function. Those tranining data are known as parallel corpora, which is a set of sentences in two languages where the sentences are translations of each other. At this point in time, there are several useful and sizable parallel corpora available, but there is still a need to gain more. We use a known method to extract parallel sentences from a comparable corpora, and use the extracted sentences as parallel corpus to help in MT accuracy. We describe the methodology for doing this, and how we used the Moses MT Experiment Management System to run our experiments.

1 Comparable Corpora

1.1 Overview

Statistical Machine Translation (MT) systems are designed to find patterns of how a source language is translated into a target language. In order to do this, large amounts of what are called parallel corpora are required for training the systems. These parallel corpora are sentences in the two languages, where each line in each corpus corresponds to its translation in the other language. There are several well-known corpora like these available, such as Europarl[1], JRC[2], and DGT[3]. These are very useful corpora for the languages that are supported. In general, the more training data like these that can be used, the better and more accurate the translations tend to be.

Though there are useful parallel corpora available, there are some limitations. First, not all languages are supported. Second, these training corpora only cover a small domain of possible translations. Natural human languages are dynamic and speakers of those languages create novel sentences continually, but if we train an MT system on corpora within a certain domain and context, it might be difficult to later discern how something should properly be translated.

For example, Europarl contains proceedings of European Parliament and both JRC and DGT are translations of law into the European languages. Now, how should the MT system translate, for example, a text about sports or technology? Thus we need to increase the reach of domains that an MT system has. To combat these two problems, we need parallel corpora for each language and each domain. However, coming up with these translations gets expensive and is very time-consuming. There is, however, an easier way.

The Internet has a vast amount of text in many languages. There are websites such as news sites and wiki sites that might contain similar information on a subject in different languages. Even though a news article in two different languages may not be translations of each other, they will most likely contain very similar information. In the same way, wiki sites such as Wikipedia have articles and pages about millions of topics, many of which are in a number of different languages. Again, they may not necessarily be translations of each other, but they might contain similar information. Now imagine obtaining many thousands of these pages into a corpora in two languages. These corpora, though not quite ready to be used as MT training data, are called comparable corpora, and with some processing could be made to be useful to an MT system.

1.2 Previous Work

As stated earlier, this idea is not new. In face, several approaches have surfaced which have all had some success to some degree.

- Approach 1: WER, TER, TERp - Sadaf Abdul-Rauf and Holger Schwenk (2009)
- Approach 2: Maximum Entropy - Munteanu and Marcu (2004)
- Approach 3: Phrase Extraction - Munteanu and Marcu (2006)
- Approach 4: Information Networks - Heng Ji (2009)

1.3 Our Work

We take the approach given by the 2004 paper by Munteanu and Marcu. First, we spent time establishing baseline scores for a number of language pairs. In order to accomplish this, we used the freely available Moses SMT system, particularly the Experiment Management System (EMS) to encapsulate and organize our experiments. We used the previously mentioned corpora (Europarl, DGT, JRC) to train and a News-Commentary corpora to test. Our baseline scores are the well-known MT metric BLEU[4]. We made use of the Juropa[5] cluster, which included installing and connecting together all of the necessary programs such as GIZA++[6], Moses[7], SRILM[8], and organizing all of the corpora. We then used a previously processed set of Wikipedia data in English and German to use as a comparable corpora, which we processed and compared to our baseline score for the German-English language pair.

2 Obtaining Baseline Scores

2.1 Moses EMS

Moses is a freely-available statistical machine translation system, and it comes with a lot of programs to help you train, tune, and evaluate your MT system. It comes with an experiment harness, known as the Experiment Management System that is a series of perl script files which read and parse a configuration file, then do all of the training, tuning, and evaluation for you. The said configuration file needs to have access to a working directory, and know where the training, tuning, and evaluation data files are. It also needs to know where all the executables are for the supporting programs that are to be run, such as SRILM, but for the most part everything it needs is relative to the moses installation directory.

One nice feature of the EMS system is that it keeps track of each step of the process, so it is easy to find and fix problems that might arise. Better still, when you find and fix a problem, you can invoke the EMS system again and it will pick up where it left off, not needing to redo everything from the start. It only redoes necessary steps. It can also detect changes in the configuration file, so if you change it and add, for example, another set of training corpora, it will know to re-train and any subsequent step that might need to be redone.

The EMS system further gives visual output of the process. Before you invoke the script to execute, you can run it in a test mode where it checks to make sure the files that your configuration file points to actually exist, and that your settings should work. It will then display an image file which is a graph, or one might call it a flowchart, of the process. Parts that need to be completed are green. Parts that are completed will turn blue. Parts that have errors and cause the process to stop will show in red, thus making it easy for you to know where to look in order to fix the problem. During execution, it will also display the image and if you leave it open, you will see it updating the color scheme as it goes along the process. If you close the image, the script will still run, and at any time you can open the image to see what step the script is on. Each step of the EMS is kept, and log files are well organized. Every training file, or output of any supporting program are kept. This is quite useful if, for example, you want to use your trained files or language model file for another experiment. It can potentially take up quite a bit of space, but that is something that MT generally requires.

2.2 Data Preprocessing

Because of the number of language pairs we dealt with, we had to use several different corpora for training and baselines evaluation. Usually each of them comes in its own format (both charset encoding and internal structure), so a bit of preprocessing was required. For each part of the EMS pipeline Moses requires the input data to be provided in two files: one containing source language sentences and the other target language sentences. These files must be in a plain text, sentence per line format with parallel sentences placed on corresponding lines.

We should mention that Moses EMS can utilize multiple corpora for given language pair in a single experiment. We took advantage of that whenever it

was possible.

As the last step, all the data has been converted to UTF-8 encoding. Sometimes this required also replacement of XML character entities by corresponding UTF-8 characters - for example ü has been replaced by Ü etc.

2.3 Baseline Scores

When evaluating the baseline scores we followed an approach taken by Koehn at al. as described in *462 Machine Translation Systems for Europe*[9] and we set up Moses system with the same settings: maximum sentence length 80 words, bi-directional msd reordering model and 5-gram language model. On the other hand we didn't experiment with exactly the same language pairs, thus we employed also several different datasets (corpora) in addition to JRC.

Following datasets have been provided by other parties and used without modification:

- *acquis* - dataset extracted from part of JRC corpus (as described in forementioned paper by Koehn et al.), provided as a test set by Euromatrix project[10]. Domain: EU legislation.
- *eparl* - dataset based on a recent release (version 5) of the Europarl corpus, provided as a training data for WMT 2010 Translation Task[11]. Domain: proceedings of European Parliament.
- *nc* - dataset based on News Commentary corpus - provided as a training data for WMT 2010 Translation Task.
- *racai* - dataset of 3000 German-Romanian sentence pairs covering three domains: legal, medical and software related (1000 sentence pairs from each domain).
- *tilde* - dataset provided by the Tilde company to evaluate the translation quality
- *wmt-dev* - dataset provided as a part of development data for WMT 2010 Translation Task (2008 test set).
- *wmt-test* - dataset provided as a part of development data for WMT 2010 Translation Task (2009 test set).

Following datasets have been extracted from freely available parallel corpora for purpose of the evaluation task:

- *dgt* - dataset extracted from entire DGT corpus. Domain: EU legislation.
- *dgt-f* - dgt dataset without the sentences found in acquis dataset (a match of sentence in any language resulted in whole sentence pair being taken off the dataset).
- *ijs-elan* - dataset extracted from IJS-ELAN corpus[12]. IJS-ELAN corpus contains 1 million words from 15 parallel Slovene-English / English-Slovene texts. Domain: not specified.

- *jrc* - dataset extracted from entire JRC corpus. Document pairs with the same number of sentences have been implicitly treated as 1:1 aligned. Other document pairs have been processed using sentence alignments provided with the corpus and only sentence pairs with 1:1 alignment have been put into dataset. Domain: EU legislation.
- *jrc-f* - dataset extracted from JRC corpus in the similar manner as in case of jrc dataset, but with two additional restrictions: (1) Sentence pairs from the documents used for composition of acquis dataset have been omitted. (2) Any other sentences found in acquis dataset have been also omitted.
- *setimes* - dataset based on parallel corpus of the Balkan languages, generated from Setimes news articles[13].

Table 1 presents baseline scores for all the language pairs evaluated in our task.

Source language	Target language	Training dataset	Tuning dataset	Testing dataset	BLEU score
Croatian	English	setimes	-	setimes[600]	26.0
English	Croatian	setimes	-	setimes[600]	30.2
English	Estonian	dgt, jrc	tilde[1000]	tilde[520]	10.1
English	Greek	setimes	-	setimes[600]	24.8
English	Latvian	dgt, jrc	tilde[1000]	tilde[520]	10.2
English	Lithuanian	dgt, jrc	tilde[1000]	tilde[520]	11.5
English	Romanian	setimes	-	setimes[600]	40.8
English	Slovenian	dgt, jrc	ijs-elan[1500]	ijs-elan[1000]	11.4
German	English	eparl, nc	wmt-dev	wmt-test	19.4
German	Romanian	dgt, jrc	racai[1500]	racai[1500]	18.1
Greek	Romanian	setimes	-	setimes[600]	35.4
Latvian	Lithuanian	dgt, jrc	tilde[1000]	tilde[520]	8.7
Lithuanian	Romanian	dgt-f, jrc-f	-	acquis	35.7
Romanian	English	setimes	-	setimes[600]	31.2
Romanian	German	dgt, jrc	-	racai[1500]	14.9
Romanian	Greek	setimes	-	setimes[600]	22.1
Slovenian	English	dgt, jrc	ijs-elan[2000]	ijs-elan[1000]	12.4

Table 1: Baseline scores for all evaluated language pairs.

3 Maximum Entropy

3.1 Munteanu and Marcu 2004

In the Munteanu and Marcu 2004 paper, they trained a maximum entropy classifier with some small parallel corpora and non-parallel corpora. Their training data included some in-domain traning data (news text), but mostly out-of-domain data (UN Proceedings). They also used a specific amount of non-parallel training data to add to the maximum entropy classifier. They did this so the classifier would be able to learn about truly parallel sentences and non-parallel

sentences and be able to distinguish what it takes for two sentences to indeed be parallel.

In order to train a maximum entropy classifier, they needed to determine what features could be used to best distinguish if two sentences are parallel or not. Of course, one very important thing is to know if an individual word in a sentence translates into a word in the target sentence, so a dictionary will be necessary. They also made use of the IBM Model 1 which gave a word-level alignment which became a very strong indicator if two sentences are translations of each other. If two sentences have words that don't correspond via a dictionary, they obviously won't be parallel. Further, they took the sentence lengths, longest continuous span of translated words, longest unconnected string, and used those features for training with the before mentioned training data.

With a trained model, they then took large amounts of news text that they previously obtained. Now, which sentences should we compare? If we take a single sentence in the source language and then test it against every sentence in the target language, the processing could take a very, very long time. So, they took news articles that were within 5-10 days of each other, taking the assumption that news stories which discuss the same event will appear on websites in a similar tempora timeframe. This greatly reduced the number of comparisons, but will left it open enough to find sentences that would indeed be translations of each other. They kept the sentences that had a 0.7 probability of being translations of each other and appended those to a list for each language, in the end making a parallel corpus, in their case, in Arabic and English.

3.2 Our Project

We did something similar to the 2004 Munteanu and Marcu paper. However, we didn't use the IBM Model 1. Instead, we found a custom dictionary and did our own alignment search. We used a freely available maximum entropy classifier[14] written in C++. It was quite easy to adapt to our needs. We used features such as number of translated words, longest contiguous span, longest unconnected substring, sentence lengths, and the difference between the source and target sentences.

Also unlike the paper, we used Wikipedia text. The set of all articles in German and English were previously processed for matching titles and sentence segmenting. We further processed the resulting text to make sentences into smaller segments by splitting on specific punctuation such as commas and parenthesis. This works because the sentences aren't aligned parallel, but are rather compared by taking a page in English and German where the title is the same and checking all sentences with all other sentences, where a sentence is on its own line in the file. Shorter segments are easier for an MT system to process and learn from, so if possible we keep the sentences short.

We train on 10,000 sentences from Europarl and another 10,000 sentences from the news-commentary corpora for the parallel training. For the non-parallel training, we use the same corpora, only we shift the source language by 1 sentence, so a sentence is compared to the sentence that is 1 line below its corresponding sentence. These sentences are non-parallel and useful to help the maximum entropy classifier distinguish between translations and non-translations. One thing to note here, and a similar issue arose in the 2004 Munteanu and Marcu paper, was the amount of training sentences for parallel

vs. non-parallel. To be accurate, most sentences in existence in two languages are not translations of each other, but if we trained, for example, using 10,000 sentences of parallel text, and then trained each sentence on the 9,999 other non-parallel sentences, we would end up with a model that has seen so many non-parallel sentences that it would just assume all sentences are non-parallel and we wouldn't find any possible translations at all. Allowing the number of parallel training sentences and non-parallel traning sentences to be approximately the same allows the features to do most of the distinguishing.

As explained before, we took our trained model and used it to distinguish sentences in the two languages given Wikipedia articles on the same topic. We checked every sentence in the article in the source language against every sentence in the target language. Our threshold of probability for sentences to be translations of each other was set to 0.99, because during our tests it was considering some sentences to be translations that obviously weren't when it was set to 0.7 as in the paper.

The number of resulting sentences after processing was 56,000. On inspection, a look at some of the sentence translations looked reasonable. German and English had their own respective sentence files that were added to the Eurparl/News-Commentary German/English baseline, which was 19.38, and retrained. The improvement was only slight, the new BLEU score increased to 19.40.

4 Conclusion

In our case, there wasn't much improvement but there are many improvements we can make to our system, including adding the IBM-1 model information as a feature. However, there are many languages and language pairs that need improvement in their machine translation ability, and making use of comparable corpora is a practical approach to getting the training data needed to improve machine translation for many language pairs.

5 Acknowledgements

This work was partly supported by the FZ Jülich with an access to the IBM Blue Gene/P computer JUGENE through project HB07.

References

[1] Various, *European Parliament Proceedings*, statmt.org.

`http://www.statmt.org/europarl/`

[2] Various, *JRC-Acquis Multilingual Parallel Corpus*, langtech.

`http://langtech.jrc.it/JRC-Acquis.html`

[3] Various, *The DGT Multilingual Translation Memory of the Acquis Communautaire*, langtech.

http://langtech.jrc.it/DGT-TM.html

[4] K. Papineni, S. Roukos, T. Ward and W. J. Zhu, *BLEU: a method for automatic evaluation of machine translation*, ACL 2002, pp. 311-138.

[5] Various, *JuRoPA - Jülich Research on Petaflop Architectures*, fz-juelich.de.

http://www.fz-juelich.de/jsc/juropa/

[6] F. J. Och and H. Ney, *A Systematic Comparison of Various Statistical Alignment Models*, Computational Linguistics, volume 29, number 1, pp. 19-51 March 2003.

http://www.fjoch.com/GIZA++.html

[7] Various, *Moses - statistical machine translation system*, statmt.org.

http://www.statmt.org/moses/

[8] A. Stolcke, *SRILM - An Extensible Language Modeling Toolkit*, Proceedings of International Conference on Spoken Language Processing, Denver, Colorado, September 2002.

http://www.speech.sri.com/projects/srilm/

[9] P. Koehn, A. Birch and R. Steinberger, *462 Machine Translation Systems for Europe*, Proceedings of International Conference on Spoken Language Processing, Denver, Colorado, September 2002.

[10] Various, *Euromatrix Test Sets List*, statmt.org.

http://matrix.statmt.org/test_sets/list

[11] Various, *Shared Task: Machine Translation for European Languages*, ACL 2010 Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR, statmt.org.

http://www.statmt.org/wmt10/translation-task.html

[12] Erjavec T., *The ELAN Slovene-English Aligned Corpus*, Proceedings of the Machine Translation Summit VII, pp. 349-357. Singapore, September 1999.

http://nl.ijs.si/elan/

[13] Tyers F. M. and Alperen M. S. *South-East European Times: A parallel corpus of the Balkan languages*, Proceedings of the Workshop on Exploitation of multilingual resources and tools for Central and (South) Eastern European Languages, LREC 2010.

http://www.setimes.com

[14] Various, *A simple C++ library for maximum entropy classification*, University of Tokyo, Tsuji Labratory.

http://www-tsujii.is.s.u-tokyo.ac.jp/~tsuruoka/maxent/