Multiword Expression Extraction using data from parallel corpus

Anna Vechkaeva

Introduction

Multiword expressions (MWE) identification is an important task in computational linguistics. For example, it can be used in machine translation, as in general, literal translation of MWE from one language to another does not convey the original meaning.

So, in order to perform a good translation it is necessary to identify multiword expressions in source language and their translations into target language. For machine translation task, multiword expressions extraction can be performed for each language pair separately, as for some language pairs literal translation of some expressions may be the correct translation.

There are lists of multiword expressions for different languages. So, why not to use them in machine translation? Firstly, this lists can be not full and contain not all the expressions. Secondly, languages evolve, over time, new multiword expressions can be created. This process is not easy to document.

In order to have valid list of multiword expressions, it is necessary to have tool which can extract them from texts.

The aim of this work is to extract multiword expressions for the English language using parallel aligned data.

In order to extract multiword expression I used Europarl parallel corpus [Koehn, Philipp 2005]. This corpus is extracted from the proceedings of the European Parliament. It contains parallel data for 21 languages. As german-english corpus is the biggest, I decided to use it in my project. The sentences in the corpus are aligned, however it does not contain word alignment. In order to get aligned data I used GIZA++ [Franz Josef Och, Hermann Ney 2003].

Steps for MWE extraction

Data Normalisation

There were cases in the corpus when to a sentence in one language an empty string in another language was aligned (about 8000 empty strings in english text and about 2000 empty strings in English). Some of these misalignments could be fixed.

The reason for the most common misalignment was the fact that in German the official addresses are usually marked as separate sentences, however in English they are marked only with coma. So, it appeared that empty lines in English were aligned with addresses in German (e.g. Frau Präsidentin!). And addresses in English were attached to the next sentence.

Such misalignments were easily fixed. For it and other similar cases, I wrote the script, which attaches the “lost” part of sentence to its main sentence. It decreased the number of empty lines to about 3 000 for the german text and to about 1500 for the english one. However, there were cases when it was not possible to automatically determine to which sentence (previous or next) this “lost” part should be attached. In order to avoid word alignment mistakes based on these sentence misalignments, in these cases, I removed the empty line, the sentence aligned to it and next and previous sentences in both corpuses. I had to remove previous and next sentences because these misaligned parts should belong to one of them, and its absence could decrease the accuracy of word alignment with GIZA++. All the removed sentences some up to about 15 000. As the English-German Europarl corpus has almost 2 000 000 sentences, it did not have much of impact to the size of the corpus.

The script which deals with empty lines (normalisation/emptyln.py) gets two files as an input. The files should contain text in two languages. Each sentence should be a separate line and sentences in the two texts should be aligned. It outputs two texts, in which sentences are aligned and there are no empty lines.

The sentences in Europarl parallel corpus are aligned, however in order to use GIZA++ further processing was needed. First of all it was necessary to tokenize all wards, lowercase them and remove punctuation.

The script for text normalisation (Normalisation/normalize.py) takes output of emptyln.py (http://emptyln.py) and tokenizes, lowercases them and removes punctuation.

Runing GIZA++

Two normalized on previous step files were given to GIZA++ as input. German text was given as source language and the english one as target language. It makes next step (MWE extraction) easier. GIZA++ ran five iterations of model1 followed by five iterations of model3 and five iterations of model4. Output files generated by GIZA++ are in the folder (name of the folder) As an output I got a file giza.VA3.final with alignments.

MWE extraction

After I got the aligned data, I could start to extract multiword expressions. At the initial step, all english words, which are clustered together and are translated into German as one word were considered candidates for being multiword expression.

The next step was to filter all the candidates. To do so I decided to check how many times the MWE candidate was encountered in corpus, than find percentage of the times it was aligned with one word. If this percentage was more then 50% this words were considered multiword expressions. To get more accurate data, on this step only expressions which were encountered in the text more then 10 time were considered.

In order to filter out such expressions as “of the”, which only consist of determiners and auxiliaries, I decided to use a list of english stop words [Stopword Lists]. If the extracted expression consisted only of the words from this list, it was not considered multiword expression. Also, if the MWE candidate had only two words and one of them was in a stop words list the words were not considered MWE. It helped to deal with expressions of the form DET + NOUN being placed in the final list of multiword expressions. The stop words list is stored in a separate file ‘/CL_project/MWE_extraction/stop.wrd’.

Problems

After running all the steps listed above on the english-german corpus of 20 000 sentences I realized that the German language was not the best choice for the task of multiword expressions extraction based on parallel data. The problem is that German has a lot of compounds which are usually translated into English as several words. So, as a result of MWE extraction I got a list of MWEs the majority of which was a translation of german compounds (e.g. “community law” aligned with “gemeinschaftsrechts” or “full employment” aligned with “vollbeschäftigung”). The list of MWE extracted based on english-german corpus is in the file ‘CL_project/results/MWElist_en_de_20k.txt’.

In order to avoid the problem with german compounds, I decided to run the MWE extraction on a different language pair. English-greek Europarl corpus was randomly chosen.

Results

English-german corpus

Although there was a problem with german compounds when running MWE extraction on english-german data, some actual multiword expressions were extracted (e.g. “long term”, “taken into account”, “taken place”, “give rise”, “point of order”, “at long last”). Some MWE were extracted only partly (“one hand”). Only about 14% of extracted expressions were actual multiword expressions.

English-greek corpus

I ran MWE extraction script on english-greek corpus of 50 000 sentences. To normalize it, I used normaisation script which I wrote for english-german corpus. It dealt with empty lines in english-greek corpus worse then it would for english-german corpus. If it found empty line, it just deleted the line of its number, of the previous one and of the next one from both files.

For english-greek corpus the multiword extraction script worked better then for english-german corpus. The number of MWEs it extracted based on greek data was smaller then for german data (for corpus of 20 000 sentences, on english-german data it found 223 MWEs, and on english-greek data only 45 MWEs).

For corpus of 50 000 sentences 143 MWEs were found. The list of MWEs extracted based on english-greek data can be found in the file '/CL_project/results/MWElist_en_gr_50k.txt'. According to my judgement, only 30% of them are actual multiword expressions, which is much better then in case of using english-german language pair.

How to run the scripts

In order to run normalisation step, it's enough to put two texts in different languages with aligned sentences (the text from which MWEs should be extracted should be called "corpus.MWE" and the second text should be named "corpus.src") at directory '/CL_project/normalisation/' and run following commands in the Command line:

```
$ python3 CL_project/Normalisation/emptyln.py
$ python3 CL_project/Normalisation/normalize.py
```

The next step is running GIZA++. Source corpus should be the corpus based on which the MWEs will be extracted and target corpus should be the corpus with the text on the language for which the MWEs will be extracted.

After running GIZA++ the output file with alignment (*.A3.final) should be placed to the directory '/CL_project/MWEextraction'. To extract multiword expressions the following comands should be run in the Command line:

```
$ puthon3 CL_project/MWEextraction/MWEextraction.py
```

The output file which contains a list of extracted multiword expressions ("MWElist.txt") can be found in the directory '/CL_project/MWE extraction/'.

References

Koehn, Philipp. “Europarl: A parallel corpus for statistical machine translation.” MT summit. Vol. 5. 2005.

Stopword Lists. Default English stopwords list: http://www.ranks.nl/stopwords (http://www.ranks.nl/stopwords)

Franz Josef Och, Hermann Ney. “A Systematic Comparison of Various Statistical Alignment Models”, Computational Linguistics, volume 29, number 1, pp. 19-51 March 2003