Applied Language Technology: Assignment 1

Stian Steinbakken (11962992) & Kristian Korrel (1031937)

September 29, 2017

1 Program execution

We have provided two Python 3 scripts for this assignment. *data_reader.py* is used to read in sentences and alignments from either the internet or local files. It defaults to reading in the files from a *data* directory which should reside in the same directory as the script. Alternatively, you could change the parameter *read_locally* to *False* to read in the data from the internet. Note that when reading in the word alignment file, we switch the order from GE-EN to EN-GE.

phrase_extraction.py is the main file for this assignment. It can be run without any arguments, by typing the following command:

```
$ python3 phrase_extraction.py
```

The script will first read in all English and German sentences and word alignments. After all lines are read, it will extract the consistent phrases using an algorithm of which the pseudocode is provided in Section 5.2.3 of [1]. All phrases with more than 5 words are ignored. During this process the script will also obtain and store useful statistics, which will be used later to calculate translation probabilities. These statistics include the number of times individual words appear (in both languages), the count for each word alignment (in both directions and the number of times phrases appear and how often they appear together. Since the corpus is limited in this assignment, we chose to store all this information in memory instead of on disk. After all phrases are extracted and the relevant statistics are obtained, we calculate the phrase probability and lexical weighting (KMO approach) for each extracted phrase pair and write this to a file in the same directory as the Python script. The file contains all the phrase pairs the program has extracted, in addition to the probabilities of each pair, their lexical values and their frequencies. Each line is of the form

$$f \;|||\; e \;|||\; p(f|e) \;\; p(e|f) \; l(f|e) \; l(e|f) \;|||\; freq(f) \; freq(e) \; freq(f,e) \qquad (1)$$

2 Discussion

The idea behind statistical language models is that all components of a language (such as vocabulary, grammar, etc.) can be learned from data. This, however, requires training data which resembles the true probability distribution of the underlying language (translation). Because both the amount of data that can be obtained for training and training time is limited, certain data sparsity problems are always likely to occur.

Section 5.3.2 of [1] discusses the problem that an uncommon foreign phrase f can be mistakenly mapped to a common English phrase e with a high probability. We can see examples of this in the second row of table 1. For instance, we can see that the word 'that' occurs 4002 times as a phrase, while the paired sentence only occurs once, thus yielding a translation probability $p(e|f)$ of 1, which seems quite counterintuitive if we look at the contents of the phrase pair. To combat this phenomena we could use bidirectional translation probabilities; that is using the translation probability in both directions as feature functions in a log-linear model.

Another problem occurs when both phrases do not occur often in the training corpus. Examples are given in rows 4 and 5 of Table 1. The example in row 4 shows phrases which appear only once. This has the effect of both $p(f|e)$ and $p(e|f)$ becoming 1 which is probably a major overestimate. The solution we discussed before does not help us here since the problem occurs in both directions. We view the phrases as atomic units. Since these units are not well enough represented in our data, a solution to this problem is to break down the phrases into smaller parts, namely words. One implementation of this idea was provided by lexical weighting [2]. We have also implemented this in our code.

Another interesting thing to note is that in the example in the first row of Table 1, *durch* only has a word alignment to *in*. All other English words in this phrase pair are unaligned and can therefore be included in a consistent phrase. The fact that this phrase pair only appears once in the corpus however, probably indicates that although consistent, this phrase pair is not a good translation. Pairs like these should probably be removed by some pruning method which would also reduce the size of the transition model and therefore translation time.

The problems mentioned above are partly due to the size of the phrases. Generally, the longer the phrases, the less the chance of them appearing in the corpus, thus yielding a higher chance of making phrase pair with questionable probabilities as the ones we have discussed above. In our experiments we limited our phrases to length 5.

References

[1] Philipp Koehn. *Statistical machine translation*. Cambridge University Press, 2009.

[2] Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based

Foreign phrase	**English phrase**	$p(f\|e)$	$p(e\|f)$	$l(f\|e)$	$l(e\|f)$	$freq(f)$	$freq(e)$	$freq(f,e)$
durch	point in having rights without	0.5	0.0012	0.0032	1.74e-14	831	2	1
, vorangetrieben werden	that	0.00024	1.0	6.59e-08	0.16254	1	4002	1
.	distribution sector .	1.0	0.00016	0.9568	1.236e-08	6161	1	1
innere politische lage in einem	internal political situation of one	1.0	1.0	2.699e-06	0.00026	1	1	1
haben wir besonders	we have taken particular	0.5	1.0	0.01705	0.0001	1	2	1

Table 1: Examples of phrase pairs

translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics, 2003.