UNIVERSITY OF EDINBURGH

Machine Translation
Assignment 2

s1533593 & s1549106

February 10, 2016

QUESTION 1

Table 1 shows the speed and scores for different parameters of the model. It is apparent that with 50 hypotheses per stack and 50 translations per phrase the model is already reaching a ceiling performance. In other words, it is futile to allow for extra hypotheses in the stacks, given that a large subset of them are extremely unlikely and only slow down the process. It is also clear that to take advantage of larger stacks one needs to consider more translations. However, the steepest improvement in performance is seen when raising k to 5, even without increasing the stack size. As for the translations themselves, while some portions of texts are more grammatical when likelihood is increased, many others appear equally distorted. In other words, the improvement is not explicitly manifest in the output. Some errors in particular are always repeated; for example the translation of *elle* to *it*, even when the semantic role of agent would suggest an animate referent (and therefore *she*).

QUESTION 2

Allowing for adjacent words to swap once increases the size of our search space. More precisely, we obtain a set X of $2(n-2)+1$ possible paths (e.g. figure 1) for each phrase combination covering the whole foreign sentence f, where n is the number of phrases (if we also allow for punctuation to be swapped). Therefore, when we compute $h(j,e)$, we do not assume $h(i,e)$ to be its only possible prefix; we must also consider a parallel $h(i,e)$ in which a swap occurred. Our new search space is thus:

$$e^* = arg\max_e \max_{x \in X} log\Big(\sum_a p_{TM}(f,a|e)p_{LM}(e|x)\Big) \tag{1}$$

We implement this search space by gradually building alternative hypotheses at every stack i. To do so, we add a label to each hypothesis that tells us whether it was swapped. If it was not swapped, its receiving stack will produce another prefix hypothesis with the swap. However, because at every stack we only consider the s most probable hypotheses, we are not actually considering all paths for each set of phrases; many will be formulated and discarded right away. Moreover, because we perform a breadth-first search in the resulting graph, we never calculate the same sub-path twice.

QUESTION 3

The computational complexity of our algorithm is $O(skI)$, where I is the length of f and s and k are the parameters discussed above.

QUESTION 4

A hypothesis that spans the words f_i to f_j is placed in stack j. This is made possible by the adjacency assumption we made, otherwise stack j would have to contain hypothesis of length j (i.e. covering j words, adjacently or not).

QUESTION 5

See code.

QUESTION 6

Table 2 shows the time of decoding, along with the log-likelihood of the data given different parameter for our new model. As expected, this model performs better despite being slightly slower. The pattern of increased likelihood and time is along the same lines as in table 1 for the first few runs. Interestingly, the likelihood starts decreasing for high values of s and k. The quality of the translations upon human inspection does not seem to vary considerably. The swaps also occasionally cause mistakes. For example, *word by word* is now *word, to the words.*
We also explored the possibility of only allowing non-adjacent swaps, meaning that if h1 and h2 have been swapped, h3 and h4 may not. This allowed us to check that the decrease in performance is due to the models, which cannot cope with too large a search space. In fact, in table 3 this decrease is much slower. For this reason, we explore more options for reducing the search space in question 7.

QUESTION 7

First, we implemented a more memory efficient version of monotonic decoding, reducing usage from 98258944 bytes to 97923072. Secondly, from Wuebker et al. (2012), phrase based SMT can be improved using look ahead LM computation. This allows to avoid calculating the p_{LM} for all phrases. We can do so by calculating a weighted score for a candidate phrase $\tilde{e}$ and pruning these. We include two more parameters. One is a log-probability threshold w for the innermost search loop (Wuebker et al., 2012: 2.2). This LM look-ahead score (pre-calculated and stored in a lookup table) is added to the overall hypothesis score and if this final sum is less than provided threshold, then the current hypothesis is discarded without computing the full LM score. Phrase candidate pre-sorting is implemented as in Wuebker et al., 2012: 2.1, who pre-sort the phrase list (taking the top m candidates) based on a weighted LM score of all the words of this phrase, assuming the phrase to be a complete sentence. An example of a working parameter setting is:

```
mydecoder.py -s 4 -w -100 -m 5 -k 10
```

FIGURES

Table 1: Performance for different parameter settings

k	s	time	log-likelihood
1	1	0.941	-1439.873990
1	5	0.963	-1436.360138
5	1	0.972	-1375.793152
50	5	1.540	-1355.897952
5	50	1.386	-1359.829402
50	50	5.567	-1353.247828
100	100	11.878	-1353.247828
10	100	2.745	-1354.642177
100	10	2.413	-1353.247828
500	500	31.649	-1353.247828
50	500	19.001	-1353.247828
500	50	8.138	-1353.247828

Figure 1: Graph for a set of four phrases

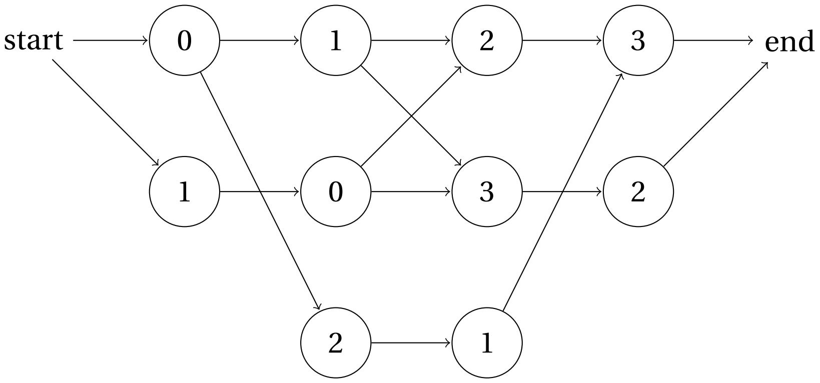

Table 2: Performance for different parameter settings with swapping.

k	s	time	log-likelihood
1	1	0.979	-1409.498800
1	5	1.010	-1389.626999
5	1	1.017	-1366.643021
50	5	2.232	-1353.530533
50	50	11.479	-1387.260049
100	100	27.054	-1399.221462
500	500	112.352	-1416.329285

Table 3: Performance for different parameter settings with non-adjacent swapping.

k	s	time	log-likelihood
1	1	0.979	-1408.951130
1	5	0.971	-1389.876218
5	1	0.964	-1371.733392
50	5	5.217	-1351.981240
50	50	11.479	-1369.820616
100	100	25.808	-1373.724584
500	500	112.352	-1378.246032

REFERENCES

Wuebker, J., Ney, H., & Zens, R. (2012). Fast and scalable decoding with language model look-ahead for phrase-based statistical machine translation. In *Proceedings of the 50th annual meeting of the association for computational linguistics* (pp. 28–32). Jeju, Republic of Korea. Retrieved from `http://www.aclweb.org/anthology-new/P/P12/P12-2006.pdf`