LINGI2263 : COMPUTATIONAL LINGUISTICS

Group 2 : Project 2

Authors:
Crochelet Martin (2236-10-00)
Baugnies Benjamin (6020-10-00)

Professor:
Pierre Dupont
Cédric Fairon

2013 - 2014

1 Tags and words statistics

The first part of the assignment consisted of creating a Lexicon based on the training corpus, and extracting some general information about the words and tags it contains. Our lexicon is limited to the 5000 most common words. Other words are replaced by the `<UNK>` token. Among these words, we extracted the 10 most common words. They are summarized along with their frequency in table 1.

Rank	Word	# occurences
1	THE	59466
2	,	49639
3	.	41859
4	OF	31067
5	AND	24709
6	TO	22190
7	A	19718
8	IN	18230
9	THAT	8974
10	IS	8623

Table 1: Most common words

While parsing the texts, we found 187 different tags. The 10 most and least common of these can be found below 2.

	Most Common			Least Common	
Rank	Tag	# occurences	Rank	Tag	# occurences
1	NN	143023	-1	JJR+CS	1
2	IN	104385	-2	JJ$	1
3	AT	84266	-3	NN+HVD	1
4	JJ	58365	-4	RBR+CS	1
5	.	51997	-5	IN+IN	1
6	,	49641	-6	NR+MD	1
7	NNS	49412	-7	IN+NP	1
8	NP	32992	-8	WRB+BER	1
9	CC	32496	-9	WRB+MD	1
10	RB	31147	-10	NP+MD	1

Table 2: Most and least common tags

We can see that the most rarely seen tags are almost all compound tags. Moreover, the list of least used tags is actually not very precise since a lot more than 10 tags are used only once. The presented list here is just a chosen subset of those tags but we could have chosen other tags.

2 Tags and words statistics

3 HMM Tagger

This last part has been more difficult than foreseen: indeed, we have been able to compute the emission and transition matrix from the training set however, computing the whole set of possible states for a given segment has been quite a challenge. In the joined code, you can see that the whole HMM POS-tagger has been implemented less this challenging part. We expect the tagger to be a lot more slower than the baseline one since it has to access the matrix for each possible state which is a lot more complex than the normal, baseline, version. Moreover, our choice of smoothing is a little bit unorthodox since we simple use a default epsilon value; a better implementation would follow the formula given at slide 18 of the fifth lecture.

About the evaluation of the tagger, we have obviously not been able to execute it however, the implementation is given in the python file compare.py.