TDT4310 – LAB 2

Exercise 1

a) The most frequent tag in the Brown Corpus is NN, which appears 152 470 times.
b) 7756 words are ambiguous.
c) 15,57% of the words are ambiguous.
d) The words a, that, to, in, :, it, as, well, out and i are the most ambiguous. The sentences are printed in the source code.

Exercise 2

	Accuracy
BT	0.83
	0.88
	0.71
	0.85
UT	0.83
	0.87
	0.70
	0.83
RT	0.20
	0.19
	0.13
	0.13
DT	0.13
	0.13
	0.09
	0.09

Brown 90/10
Brown 50/50
Chat 90/10
Chat 50/50

BT = Bigram Tagger

UT = Unigram Tagger

RT = Regex Tagger

DT = Default Tagger (with NN)

a) It seems that DT performs better on a 50/50 split, maybe it has a bigger chance to correctly guess an NN tag with more test data? It does not benefit from more training data since it does not learn from it. It also seems like DT performs worse on the NPS Chat Corpus than on the Brown Corpus. I think that is because the Brown Corpus is more standardized (in terms of language and syntax) than the NPS Chat Corpus, which means it probably has more nouns so there is a bigger chance to guess correctly.
b) You can see the data in the table above, and in the source code.

Exercise 3

Brown 90/10
Brown 50/50
Chat 90/10
Chat 50/50

BT = Bigram Tagger

UT = Unigram Tagger

RT = Regex Tagger

LT = Lookup Tagger

	100%	75%	50%	25%
BT	0.84	0.81	0.84	0.78
	0.88	0.89	0.87	0.86
	0.72	0.68	0.67	0.68
	0.86	0.84	0.83	0.80
UT	0.83	0.81	0.84	0.78
	0.87	0.87	0.86	0.85
	0.70	0.66	0.65	0.68
	0.83	0.82	0.81	0.79
RT	0.20	0.20	0.20	0.19
	0.19	0.20	0.20	0.19
	0.13	0.13	0.13	0.13
	0.13	0.13	0.13	0.13
LT	0.53	0.52	0.52	0.51
	0.52	0.52	0.51	0.52
	0.57	0.52	0.57	0.59
	0.57	0.58	0.58	0.59

a) When I switched to using LT instead of DT, I got the same results for RT and above (RT+UT, RT+UT+BT), when using all of the text (100%). I think this is because RT rarely does backoff to LT. I checked this by removing the backoff attribute in LT with no change to the accuracy of RT.
LT performs considerably better than DT as the default tagger, as can be seen from the provided table.

b) BT and UT have worse performance when data size is reduced, as can be seen from the graph. I think this is because less training/test data leads to less accurate results in machine learning as a general rule. LT and RT are not affected by this, as they do not learn from training.

Other things that may be of interest:

- LT performs a little better for the NPS Chat Corpus than the Brown Corpus, and I think that is because the vocabulary of the Brown Corpus is more diverse than that of the NPS Chat Corpus, which makes it harder to predict tags based on the most common words. That is only speculation from my side, the actual differences are minor (see table).
- RT performs a little better for the Brown Corpus than the NPS Chat Corpus because the language in the chat corpus is less standardized, i.e., more difficult to guess based on a pattern because of abbreviations, slang words and emojis.
- I still do not really understand how a 50/50 split gave better accuracy than 90/10, but I have understood that accuracy does not necessarily mean how good an algorithm is, in absolute terms. I also do not deny the possibility of making some mistake in the source code, leading to these numbers, but I cannot think of anything specific. Please let me know if my numbers are off because it is really bugging me!

Exercise 4

a) The probability of NN being “we” is 0.
The probability of VB being “like” is 0.001821446452927263.
The probability of PP being followed by VB is 0.2515910650776814.

b) The probability of PP VB PP NN being the tags for “I conduct my conduct” is 1.2659661944739403e-16, which seems really, really, really small to me (based on my limited number knowledge). I am not sure if I did this one correctly, and I do not have any “control data” to see actual probabilities for more common sentences. I did try using the example sentence “I like my house”, but the probability I got was not much better, so maybe an actual low probability would be abysmally small.

c) The best tags for “You should invest in the stock marked” are PP MD VB IN AT NN VB. The probability for this is 8.508902099084633e-24, which also seems awfully small.