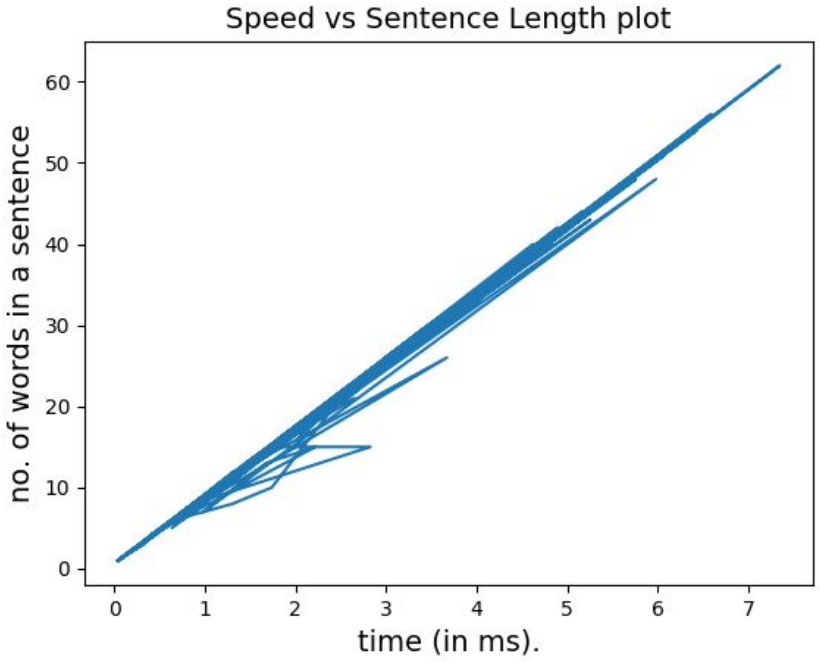
Using the crude-smoothing approach, the tagger achieved the following results on the test set.
Note: These results can be emulated by running:
python eval.py de-utb/de-eval.tt outputs/de-tagged.tt on the command-line from the project folder.

	Precision	recall	F1 score:
DET	0.8232	0.9755	0.8929
NOUN	0.9296	0.9141	0.9218
VERB	0.9202	0.9211	0.9206
ADP	0.9348	0.9775	0.9557
.	0.9608	1.0000	0.9800
CONJ	0.9498	0.8974	0.9228
PRON	0.8671	0.8364	0.8515
ADV	0.9043	0.8058	0.8523
ADJ	0.8099	0.7222	0.7635
NUM	0.9905	0.7704	0.8667
PRT	0.8712	0.9251	0.8973
X	0.2222	0.0909	0.1290

Accuracy: **0.9095**

From the speed vs sentence-length plot (shown below), it can be seen that the Viterbi algorithm is linear in the length of the sentence.



A note regarding implementation:

Since *de-eval.tt* and *de-test.t* are the same files except that the POS-column is stripped in the latter, I have used *de-eval.tt* to predict the POS-tags for *de-test.t.*

This was an implementation-based choice, since I am already using the ConllCorpusReader from nltk library to read the files in CONLL format. So to save time, instead of using another corpus reader only for *de-test.t*, I am reading only the words/sentences from *de-eval.tt*.