Itay Hassid, 209127596
Aviv Shisman, 206558157

Ex.1 - Writeup - NLP

1. How did we handle unknown words in hmm1?
Words that we haven't seen before, we replaced with signatures.
For example, words which start with capital letter and not in our dictionary we replaced with signature called “^A” and the unknown word is very likely to have the same tag as other words with capital letter.
We added also signature of suffixes, prefixes, numbers, dash containing, etc. in words, so many unknown words would fit one of the patterns.
Words that haven’t fit any pattern will be signatured as unknown - “^UNK”.

2. What’s our pruning strategy in the viterbi hmm?
In the original algorithm we iterate for every r,t and then for every t_tag so the complexity for a word is O(|TagSet Size|^3).
We changed 2 things, first, for every word we check only its possible tags, usually words in the dictionary have about 5 tags so it’s pointless to check all existing 43 tags. We do the same thing for signatures and save their possible tags too.
Computing the possible tags for every word is done before opening the test input.
Second, **t_tag** and **t** are the previous tags of the current tag **r**, so we save every pair (t_tag, t) in the i-th iteration and there’s no need to check every possible combination because we know what combination exist.

3. test scores:
hmm-greedy, hmm-viterbi, maxent-greedy, memm-viterbi
<u>Ass1-tagger:</u>
    Hmm:
        Greedy: Test set: Accuracy: 36928/40117 (92%)

Viterbi: Test set: Accuracy: 38204/40117 (95%)

MEMM:

Greedy: Test set: Accuracy: 36960/40117 (92%)

Viterbi: Test set: Accuracy: 37325/40117 (93%)

<u>NER:</u>

MEMM:

Greeady: Test set: Accuracy: 44855/51578 (87%)

Viterbi: Test set: Accuracy: 47478/51578 (92%)

The rest is in the ner.txt that in the folder “ner”.

4. Is there a difference in behavior between the hmm and maxent taggers?
Yes, maxent taggers are more flexible, you can add/remove features very easily.
On the other hand, Hmm tagger more limited than maxent because once the score function defined, the context is closed.
In addition, adding feature in maxent cause bigger change in the final result comparing to HMM.

5. Is there a difference in behavior between the datasets?
The NER dataset has less data (less tag examples) and that’s caused the per-token accuracy be higher.

6. What will you change in the hmm tagger to improve accuracy on the named entities data?
We would change the score function and find the optimal parameters that would improve the final result.
Also we can add additional features like next words.

7. What will you change in the memm tagger to improve accuracy on the named entities data, on top of what you already did?

We haven't check for rare word, so a possible thing to do is to give every word a feature as not rare and rare word, then choose the most suitable tag considering every time one of those features.

8. Why are span scores lower than accuracy scores?

Spans are tokens sequences. So one tag that's not fit ruin the whole span, while the per-token accuracy stay pretty much the same.
More intuitive explanation is that there are more tags than spans, so the tag accuracy is higher the the spans'.