

Reflection paper - TDP030

Summary of labs

Text segmentation

Here we made a simple tokenizer for an inputted text.

This lab is probably the one that took the longest, seeing as we had to find a good regex, which took a while.

Text classification

Here we made a text classifier for predicting whether a speech was right wing or left wing, and evaluated it using accuracy, precision and recall.

This was a very good introduction to the course, I feel. Very basic, but still interesting and fun to do.

Language modelling

Here we played around with n-grams in order to generate sentences based on Sherlock Holmes novels.

This was a fun lab due to the randomness of the sentences that it put out. When you got something coherent it usually ended up funny.

Part of speech tagging

In this lab we used POS-taggers in order to parse and tag text (the Stockholm Umeå corpus).

This lab was interesting in that it automagically tagged sentences, which was interesting. It was kind of difficult to answer some of the reflection questions, since it felt like they required knowledge of what the tags meant, which wasn't always specified. You could probably have answered the questions without this knowledge, but it felt necessary in order to give a complete answer.

Syntactic analysis

In this lab we used a syntactic analysis tool, and played around with the data it produced.

This lab was probably the most forgettable one, since it didn't have much implementation (at least in my opinion). It was still interesting, but it didn't feel that different from just sitting through the lecture.

Semantic analysis

In this lab we used a semantic analysis tool and played around with the data it produced.

This lab, like the last, felt like it lacked in the hands-on experience department. While interesting, it felt like it didn't help in learning much more than if you had just gone to the lecture. Repetition, not deeper understanding, so to speak.

Which connections do you see between the labs and the other parts of the course?

The labs very much help in reflecting on what the lectures teach. It helps you understand what is being taught from a more hands-on perspective, which always sticks with you better (at least for us programmers).

The labs also made it easy to choose what project to do, since we could determine what would seem fun to work with while trying the different methods out in the labs.

Which connections do you see between the labs and other courses on the programme?

For us IP students there aren't many similarities, at least as far as I can remember. I did notice some surprisingly similar elements between the implementations of natural language processing and computer language processing (especially with parse trees working exactly the same as BNF grammar).

Which of the knowledge and skills that you have developed or trained may be most relevant for you in the future?

If one would be so lucky to end up working for Google (or any other company that would greatly benefit from NLP), it would help a lot.

That being said, for programming in general I feel like the labs made you think and implement things in ways at least I don't usually operate, so the course might indirectly help in a lot of future endeavors.

Which parts of the labs did you find the most interesting? Which were less interesting?

Language modelling and part-of-speech tagging was very interesting. From an outside perspective they seem like very difficult topics, but when you actually got to implement the parts we did it was surprisingly simple, which is probably what made it so fun (besides them being interesting subjects in the first place).

How did you and your lab partner complement each other?

I would say that since we both come from the same program (IP), we both have very similar skillsets when it comes to the labs. However, since we in the IP program often practice pair-programming, we did the same here, and this is the first course I think I really felt it helped a lot. We both had to think about the problem, but since only one of us were actively coding the other could take a different perspective, which often helped (one of us was thinking of the logical side of things, and one of the implementation side).