Report for ANLP Wintersemester 2025/2026

The goal of the module ANLP was to automate the simplification of text samples to a B1 and A2 standard using two methods. We were tasks with using two methods; method one was finetuning of pre-trained small- to medium-sized Language Models (T5 Models) taken from Huggingface, while method two was to use an LLM/generative AI Model.

Task 01

For the first task we took the already provided notebook, which was used for tutorial purposes and worked based of that. We each duplicated a working version of the notebook -a direct upload of the lecture notebook had resulted in the issue of return statements being disregarded when inside the cell, so that had to be fixed before we could work with what we considered a baseline to start off from- in the molab service and worked independently on finetuning. While Kristan would later take the LLM task, Tim devoted himself to the T5 Models and Janina tried out how a different dataset would influence the results. After doing some research work on different datasets and trying to implement them, we were discouraged to use them further trying out different datasets. Since Tims had successfully implemented a new model, while Kristian had run into numerous errors and issues, we decided that Tim would take the lead for this task. While Janina and Kristian supported him anyway they could.

Tim tried out the t5-small-wikilarge-text-simplification by bogdancazan, here the problem was much obvious and presented very early in implementation since the model would repetitively repeat single words, for example ‘the the the the the’.
The next model we implemented was t5-small-finetuned-text-simplification by mrm8488 which led to more promising results with high precision but still low weighted_f of 0.2892. After some investigation by the team, it became clear that instead of simplifying the sentences often the Model would cut off sentences where previously had been commas or even in the middle of a sentence.

Finally, based on the recommendation of the professor, we tried out two more models flan-t5-large and flan-t5-small. In addition to the consultation of the professor, we also researched on how to improve the models. We found out that the length penalty could be the key to improve the simplification. Further extensive testing with the trial data and adjusting the length penalty (0.0, 0.4, 0.7, 1.0) led to the unsurprising result that using the flan-t5-large model resulted in a better weighted_f of 0,3487, when using a length penalty of 0.0.

Based on the trial results we used the flan-t5-large model on the test data, again with a length penalty of 0.0, and came to an end result of 0.4026 for our simplification task using T5-Models.

Task 02

For the second task we started off with the lecture notebooks as well. The goal was to use prompt engineering to use an LLM to simplify text samples. For this we used credits from the Google Cloud Services through which we could use the Vertex AI Platform to access Gemini Models. Kristan led the writing/re-writing of the code for this task, while Janina and Tim supported him as instructed by him.

We used the Gemini 2.5-flash Model. The system prompt sets up an LLM as a specialist for simplifying text to specific CEFR reading levels. It defined what each level requires, established optimization goals, and gave guidelines on how the model should write at each level. The purpose of this was to ensure controlled, consistent text simplification that is both accurate and stylistically aligned with the provided examples. The system prompt contained redundancies, but in the hope that this would reinforce the adherence to the instructions. We tested our approach with zero, one and two shot learning in the user prompt, using the first two text samples of the test data. We also prompted the LLM to always give us simplifications for both reading levels in the hope that by creating both answers simultaneously it would be better at differentiating the levels.

For the test data the model results followed our assumptions, zero shot learning performed the worst with a weighted_f1 score of 0.59 whereas with one shot learning it shot up to 0,72 and a small improvement for the two shot where the weighted_f1 was at 0,74. We expected similar results when running the validation data. This turned out to be a wrong assumption as when we ran it with the validation data the zero shot now was performing the second best, in the 0,7 range as well. We are unsure why this is.

Finally, we uploaded everything into our repository and Janina took over the main part in writing this report. During the whole process we kept in contact and support each other, so that while most of the cleanup of the notebook and report writing was done by Janina, the implementation of the seq2seq Models by Tim and the majority of the LLM work by Kristan we each had a hand in each step of the project. From small changes, to helping with specific larger problems/errors the project can be described as a team effort.