**3 Methodology**

In this section, the methodology of this paper is outlined. First, I give an overview, then, I explain the methods of prompt engineering and fine-tuning. I further introduce the two datasets.

**3.1 Overview**

For the main research question of detecting and classifying political stance in the legislative proposals, ChatGPT3.5 is fine-tuned on 1,393 proposals and their voting outcome. The fine-tuned model’s accuracy in classification is compared to the a prompt-engineered baseline version of ChatGPT3.5, see figure xy. The fine-tuned model is then further fine-tuned on a small set of rejected amendments with the task to generate counterfactual political stance, exploring RQ2. The answers of this second fine-tune are qualitatively compared to a version of ChatGPT3.5 that was only fine-tuned on rejected amendments. This serves to inquire if fine-tuning on stance detection can also improve the generation of text with counterfactual political stance.

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Figure 2: Overview of the methodology in this thesis

**3.2 Prompt Engineering ChatGPT3.5 as a Base Model**

As touched upon above, while general LLMs perform NLP tasks with remarkable performance, fine-tuning can improve LLMs’ performances further. To obtain a baseline performance, first ChatGPT3.5 is prompted to classify the political stance in the given legislation. To find the best possible answers, prompt engineering is used to refine the model’s performance[[1]](#footnote-1). Prompt engineering is the process of giving an LLM instructions “by customizing it and/or enhancing or refining its capabilities“ (White et al., 2023). By using so-called “hard prompts”, prompt engineering is different from prompt tuning, which works with “soft prompts” - adjusting additional parameters as part of the input sequence (see Wang et al.’s influential paper (2023) and Liu et al. (2021) for an overview). The following methods of researched prompt engineering are applied in combination:

|  |  |
| --- | --- |
| Prompt Engineering Method | Implementation Example |
| Role prompting:  Assigning a role to the model which to imitate (G. Li et al., 2023; Shanahan et al., 2023) | “Imagine you are a member of the European Parliament and based on your years of experience, you are an expert in predicting how the different party groups will vote on a given law.” |
| Instruction prompting:  When giving complex instructions, add details to the task and specific steps to complete the task (Efrat & Levy, 2020; Mishra et al., 2022) | “Assess the political direction, wording, framing, and topic relevance of the law. Reply with eight percentages rounded to two decimal places:  xx.xx% European Conservatives and Reformists (ECR), xx.xx% …”” |
| Asking the model for an improved version of the prompt: LLM-chosen or LLM-improved prompts have been shown to outperform human-written prompts (White et al., 2023; Zhou et al., 2023) | “Suggest an improved version of this prompt for better and more accurate results.” or “Ask four additional questions that would help you produce a better version of my prompt.” |

Table 1: Prompt engineering methods and examples and how they were implemented

Other best practices used consist of adding delimiters to separate parts of the input, specifying the expected output length, using reference texts, and telling the model to take time before answering (OpenAI, n.d.-a). As recommended (OpenAI, n.d.-b), the prompt that results in the best answers is also used to prompt the fine-tuned model in a shortened version, see annex xy for the full prompts.

In existing research, the performance gain of prompt engineering has been found to be limited. For example, with large datasets or complex tasks, giving reference texts becomes difficult or infeasible (P. Liu et al., 2021). Additionally, prompt engineering alone can be error-prone and insufficient for multi-class classification tasks (Han et al., 2021). Interestingly, when prompting the model to suggest a better version of the prompt, the model asks for reference data about historical voting pattern of each party group on similar legislation. This supports the legitimacy and effectiveness of fine-tuning.

**3.3 Optimising ChatGPT3.5 via Fine-tuning**

Fine-tuning is just one of several methods to optimise LLMs for specific tasks. Apart from fine-tuning, retrieval augmented generation (RAG) as another effective method to feed an LLM new data should be mentioned. The main application of RAG is to let the LLM generate more specific and up-to-date answers using an external dataset (see Lewis et al. 2020 for an overview). For this research, the focus is on enhancing the model’s contextual understanding and domain-specific knowledge to detect political stance, as can be done with fine-tuning. This goes further than purely improving a task of question-answering via external data with RAG. Interesting new approaches are being explored by combining RAG and fine-tuning or by applying RAG for voting advisory applications(Lin et al., 2024; Schiele et al., 2024). While showing potential for further research, this research concentrates on only fine-tuning.

Fine-tuning describes the process of using a pre-trained task-agnostic model and specialising it at a certain task or multiple tasks, sometimes also called transfer learning (Brown et al., 2020). Various methods exist under the umbrella of fine-tuning methods, including gradient-based fine-tuning and in-context few-shot learning.

Gradient-based fine-tuning, also called conventional or full fine-tuning, updates all parameters of a model to perform a single task, resulting in expensive computation (Alex et al., 2022). The influential paper “Language Models are Few-Shot Learners” of Brown et al. showed that state-of-the-art performance can be achieved on various tasks with comparably small numbers of prompt examples, in this case only 32 examples per task (Brown et al)[[2]](#footnote-2). The method of in-context few-shot learning provides an LLM a task, e.g. a classification task, with some labelled examples and asks the model to predict the label of the last given example (Alex et al., 2022). It uses the pre-trained existing network and builds on it by changing or adding only the very last layer of the model (Jurafsky & Martin, 2008). Following the example of Brown et al. and Alex et al., in this research the sub-method of few-shot learning is used due to its immediate advantage over gradient-based fine-tuning explained above.

ChatGPT3.5 has been chosen as the pre-trained model because it can be fine-tuned via the publicly available OpenAI API for a fee and because it is one of the largest current LLMs (OpenAI, 2024). In theory, ChatGPT4 as an even larger model is preferable because larger models tend to show less bias when being fine-tuned (Brown et al., 2020) and higher baseline performance at zero-shot tasks (Radford et al., 2018). However, ChatGPT4 is not openly fine-tuneable, and this research’s findings must be limited to the analysis of ChatGPT3.5.

ChatGPT3.5 still represent a powerful model with the high number of 1.75 billion parameters that ensure an outstanding performance. Shi and Lipani (2023) show that continued pre-training does not necessarily help to improve the results of fine-tuning, which raises the research question by how much a fine-tuned ChatGPT4 would outperform ChatGPT3.5. It is noteworthy that fine-tuned versions of ChatGPT3.5 can outperform a baseline ChatGPT4 at tightly defined tasks (Peng et al., 2023). This supports that reasonable insights can be drawn from the results of fine-tuning ChatGPT3.5 even when it is not the strongest pre-trained model available.

**3.3.1 Prompting**

The prompt messages for fine-tuning consist of the combination of the prompt task, one legislative proposal, and the voting pattern as the expected answer, imitating a user-assistant conversation. The prompt messages were compiled in a jsonl-file and uploaded to the API, see an exmaple in annex xy.

Studies on zero-shot learning have shown that even without giving any explanatory prompt, but just by providing reference texts, an LLM can grasp what the task is (Logan IV et al., 2021; Radford et al., 2018). In this thesis, the prompt is repeated with every fine-tuning message as part of the system setting to enhance the performance of the fine-tuning even further and to rule out low performance caused by incomplete prompting. The task is further placed at the beginning of the fine-tuning messages as studies have shown that LLMs can give answers best when important information is placed at the beginning or end of the prompt (N. F. Liu et al., 2023). This has the added advantage that even for long legislative proposals which might exceed the context window, the system’s role is still prompted with the task.

**3.3.2 Hyperparameter Setting**

When fine-tuning, the OpenAI API allows to set the hyperparameters batch size, number of epochs and learning rate multiplier (lrm). The scaling factor for the learning rate can be used to control the risk of overfitting, the number of epochs defines the full iterations made through the training data. For varied sizes of training datasets, the recommended number of epochs can be calculated from the total number of tokens and the maximum number of tokens per example. With batch size sets the number of examples processed in each batch and can be set and allows to control how frequent and with how much variance the model’s parameters are updated.

**3.4 Datasets**

Two datasets were compiled to approach the two research questions, a main one for RQ1, and a smaller one to explore RQ2, an application of the first research question. The collected data is aggregated data of low ethical risk and has been approved to be in adherence with the Research Ethics Policy of the London School of Economics.

**3.4.1 Main Dataset for Detecting Political Stance**

The main dataset consists of roll call votes on the 1,393 legislative procedures that have reached the final stage of a proposed law between June 2009 and June 2022 in the European Parliament (EP)[[3]](#footnote-3). These 1,393 laws were filtered out of 30,672 individual votes on parts of laws or amendments that were voted on in the EP following the Ordinary Legislative Procedure (COD) in the same period.

Votes after June 2022 were not used in order to match the training data of ChatGPT3.5, which was trained on data no later than 2021. Due to the longer time horizon in legislative procedures, we can assume that laws which were voted on in the first half of 2022 still target topics that ChatGPT3.5 was trained on. Votes on budget decisions were further excluded from the dataset. Even though they make up a significant part of the EP’s exercise of power, they follow a format which deviates from the format of legislative texts and could decrease the accuracy of the fine-tuned model’s prediction. Given the previous discussion on legislative procedures, only votes following the COD are included.

Using the roll call votes of each individual MEP, for each of the 1393 laws the approval rates per party were calculated as the percentage of members voting in favour out of all members of the party. Party group names that changed over the three legislative periods covered (2009-2014, 2014-2019, 2019-2024) were aligned for the purpose of coding (see annex xy)[[4]](#footnote-4). Alex et al. highlight that using class labels like 0 and 1 lead to imprecise evaluation results with few-shot learners. It was hence chosen to label the party groups using their names rather than numerical values which resembles the deployment of ChatGPT and allow the model to draw from its existing knowledge (Alex et al., 2022).

Four types of majorities were defined with a threshold of 66% approval rates, a general majority, a left-leaning and a right-leaning majority, and consensus, see the definition of each type in table xy. As mentioned above, a left-right-spectrum has been found to be the most defining spectrum for policy decisions (Hix & Noury, 2009).

|  |  |
| --- | --- |
| Type of Majority | Votes in Favour |
| General Majority | EPP & S&D each >= 0.66 |
| Left-leaning Majority | The Left & Greens/EFA & S&D each >= 0.66  and ECR & EFD/IDG each < 0.66 |
| Right-leaning Majority | ECR & EFD/IDG & EPP each >= 0.66  and Greens/EFA & The Left each < 0.66 |
| Consensus | All present MEPs |

Table 2: Definitions of types of majorities

A two-third majority was chosen over a simple majority because of the consensual nature of the European Parliament, where most decisions are supported by a grand majority. Stricter definitions of majorities with higher thresholds were evaluated but led to less variance in classes (see annex xy). The definition of a two-third majority threshold maximizes the variance in classes in all legislative periods (see xy).

Ein Bild, das Diagramm, Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Figure 3: Number of legislative proposals classified into four types of majorities per legislative period

To the roll call votes, summaries of each legislation were added which were scraped from the official Parliament websites. Summaries were chosen over the original legal texts because they are written in less technical language that resembles the training data of ChatGPT3.5 better, as it is a general conversational model and not a model specialised on legal language. The full legal text further includes standardized paragraphs, e.g. addressing the involved committees and referencing previous legislation, which do not display political direction but only increase the number of tokens that needs to be processed during the fine-tuning and with it the costs.

For the overall analysis, the dataset was split into a training set of 80% per legislative period (976 bills in total) and a validation and test set of each 10% (140 bills each). Since there are only few hyperparameters to tune and the training task is complex, this was preferred over a split of 70:15:15. Each legislative period was split individually, to ensure representation of different topics, party coalitions, and possible unobserved factors during the three periods.

**3.4.2 Small Dataset for Altering Political Stance**

A second and smaller dataset was compiled to answer RQ2: “To what extent can a fine-tuned model generate amendments to a legislation that make it more consensual or of counterfactual majority support?” Here, the generative nature of ChatGPT should be exploited, which sets it apart from the previous methods in detecting political stance.

For this explorational study, rejected amendments on the topic of migration were filtered out of all amendments[[5]](#footnote-5). The search resulted in 31 rejected amendments that were proposed to 9 different proposed bills, adding to 30 pages of text (see git repository for full texts). The rejected amendments were presented by either the group of The Left (Left) or the group(s) of Identity and Democracy and Europe of Freedom and Democracy (EFD/IDG). As they represent the parties’ positions, they ideally constitute parts of counterfactual legislative proposals. The text of each amendment was scraped from the Parliament’s websites.

The data is split into 20 training examples, 5 validation examples, and 5 test examples. The five test examples are fed into the final model with three iterations, to allow for a range of generated amendments and to account for variance in GPT’s answers.

Migration was chosen as a subset for exploration because it represents a topic that has first, been voted on throughout all legislative procedures, second, is a crucial topic of recent politics and typically evokes polarized positions (Ruedin & Morales, 2019)*.* Third, immigration is a topic for which amendments from both left-wing and right-wing party groups were put forward and were rejected in a decent quantity in the observed time frame. Even though no generalizable claims can be made from the small sample, this way, the topic’s amendments show representative features. Lastly, immigration is a topic for which three subtopics could be found that both the left-wing and the right-wing party groups made amendments on. Peterson and Spirling (2018) highlight that political stances can only be compared if two opposing parties talk about the same topic differently, rather than talking about different topics.

The subtopics are framing of migrants (see Entman (1993) for the theory of frames)[[6]](#footnote-6), stance towards European integration in the context of migration, and stance towards economic migration and labour rights of migrants.

**3.5 Qualitative Analysis of Fine-tuning to alter Political Stance**

For the analysis of the generation task, a qualitative approach was chosen since it can deliver insights from a small and content-heavy dataset. Out of the many qualitative content analysis methods, the method of structuring content analysis by Kuckartz is applied, as it is specifically appropriate for explorative studies (Kuckartz & Rädiker, 2022; Mayring, 2000). Kuckartz proposes a seven-step process which includes defining deductive categories, “coding” the test following those categories and inductively adding sub-categories after deep analysis of the texts in an iterative process (2022). The three subtopics found in the migration dataset are set as deductive categories, see annex xy for definitions and examples.

The underlying hypothesis for this research question is that the model which was fine-tuned on over 1,000 legislative proposals during the classification task might be able to deliver more accurate answers to a prompt that refers to the content of the fine-tuning than a non-fine-tuned model; an hypothesis supported by Wu et al. (2023).

To examine this hypothesis, the fine-tuned model of the classification task is further fine-tuned on the small training set of rejected amendments. The prompt used for the fine-tuning asks the model to generate an amendment which a left-leaning or a right-leaning majority would likely make to the proposal, see annex xy. In the prompt, the model is given the correct supporting majority and asked for changes by the opposite type of majority. The prompt was refined using the prompt engineering methods described.

With this prompt, amendments are generated with a single fine-tuned GPT3.5 and with the double fine-tuned classifier GPT3.5. The generated amendments and the original ones are coded with the three deductively set categories. Due to the scope of this thesis, the inter-coder-reliability cannot be tested. If the fine-tuning of the classification task allows the double fine-tuned model to draw from this knowledge for generating counterfactual amendments, a similar number of subtopics about immigration by left and right are expected with similar frames used, indicating that the model can distinguish and apply both positions correctly. The results can be found in chapter 4.3, analysis.

**4 Analysis**

This section reports on the performance of the fine-tuned model in comparison to the performance of ChatGPT3.5 as a baseline model. It further assesses the LLMs’ ability to generate counterfactual amendments of legislative proposals.

**4.1 Prompt engineered ChatGPT3.5 as a base model**

For the analysis of the fine-tuned model, the variable of interest is accuracy. It is reasonable to evaluate the fine-tuning process based on the number of correct guesses over the number of total guesses to get an objective, comparable variable. Various papers have used accuracy as the way to evaluate the fine-tuning of models (Brown et al., 2020; Lan et al., 2024; Latif & Zhai, 2023; H. Liu et al., n.d.).

To assess the standard ChatGPT3.5 model, the accuracy of random guesses is computed. For the correct majority, the accuracy of random guesses of course ranges between 0.49 and 0.51, as it is a binary random guess. Random guesses for the correct party approval rates only achieve an accuracy of around 0.09 for all three legislative periods. This value results when applying a tolerance level of 5%, meaning a random guess of 0.78 instead of 0.82 for a certain party is considered a correct prediction.

The baseline accuracy of ChatGPT3.5 was calculated using 5% of the overall training set. It was aimed to keep the legislation used for calculating the baseline as small as possible to avoid that the model might already learn from the baseline prompts before fine-tuning. The standard ChatGPT3.5 recognizes the broad direction of a proposal with a good accuracy for the predicted type of majority of 0.7375. It however fails dramatically to predict party percentages correctly, with an accuracy of 0.0094, ten times less accurate than random guesses, with the same tolerance level of 5%. Even when increasing the tolerance level to 10%, the accuracy of 0.0312 does not exceed the one of random guesses. When examining its predictions, it is notable that the model mainly sticks to round numbers of 5%, 15% or 100% rather than giving precise floating values, even when specifically encouraged to do so[[7]](#footnote-7). It further includes logic errors, e.g. predicting over 80% support of S&D and EPP but no general majority, although it was given the definition of the types of majorities. Overall, ChatGPT3.5 is able to recognize the political trend and give party percentages that theoretically match the trend but fails to give accurate and plausible party percentages.

**4.2 Fine-tuned ChatGPT3.5 for Classification**

Before calculating the accuracy of the generated classifications, the training and validation loss were analysed to find the best hyperparameters for fine-tuning the model. Combining both the losses and the accuracy is recommended in the documentation of OpenAI (OpenAI Platform | Analyzing Your Fine-Tuned Model, 2024).

Since fine-tuning the full dataset is expensive, different combinations of hyperparameters were first tested on a sample of the dataset, with the precaution that not all insights can be transferred to the full dataset. Lower numbers of epochs than the recommended number calculated via the number of tokens (in this case 4) did not notably decrease the validation accuracy, higher numbers of epochs than the recommended number however led to significant overfitting, see figure xy (figure extracted from all hyperparameter plots in annex xy. Hence, only the recommended number of epochs is tested for the full dataset. On the sample dataset, lrm of <1 were found to decrease performance dramatically, lrm of >10 led to stagnant loss and accuracy. Differences in lrm between 2 and 10 were not significant. Due to the difference in dataset size, no inference can be made from the batch size of the sample dataset.

Overall, with the sample dataset the loss remained high, and the accuracy did not reach a satisfactory level. As expected, the models are not able to reliably classify given new legislation with such few training cases. In fact, the responses vary from some correctly formatted cases to utter hallucination including illegible words and made-up party group names[[8]](#footnote-8).

Ein Bild, das Text, Screenshot, Diagramm, Schrift enthält.

Automatisch generierte Beschreibung

Figure 4: Example of overfitting with higher number of epochs, loss, sample dataset

Ein Bild, das Text, Screenshot, Schrift, Diagramm enthält.

Automatisch generierte Beschreibung

Figure 5: Example of overfitting with higher number of epochs, accuracy, sample dataset

Based on the observations with the sample dataset, the full dataset was tested on the recommended number of epochs, testing learning rate multipliers of 2 and 6 and batch sizes of 5, 10, and 15. Overall, the size of the dataset decreased the importance of hyperparameter setting and all six tested hyperparameter combinations resulted in similar outcomes for loss and accuracy, see figure xy and xy.

Ein Bild, das Text, Screenshot, Reihe, Diagramm enthält.

Automatisch generierte Beschreibung

Figure 6 Accuracy of multiple fine-tuned models, trained with different hyperparameters

Ein Bild, das Text, Diagramm, Reihe, Screenshot enthält.

Automatisch generierte Beschreibung

Figure 7: Losses of multiple fine-tuned models, trained with different hyperparameters

Although slowing the increase of accuracy in the first steps, smaller batch sizes proofed to advance the model’s performance to a small degree in the last steps of the three epochs as they naturally prolonged the training process. While allowing each model to update its parameters more frequently, higher batch sizes also resulted in greater upward and downward spikes in accuracy throughout the training process. A bigger learning rate multiplier let the training and validation loss not decrease further but at a quicker pace.

All checkpoints were compared, checkpoints being certain steps of the training process that are saved as individual models possible to use. Ultimately, the validation loss and validation mean token accuracy only showed variation of less than 0.0325 and 0.0099, respectively, between the different checkpoints of finished models, see table xy.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Job ID | Training Loss | Validation Loss | Training Accuracy | Validation Mean Token Acc. |
| ftjob-GXjMS97heBGSRcXSj0a9eFvU | 0.27403 | 0.26033 | 0.925 | 0.92693 |
| ftjob-ORbrKicdOcpRWKUqcFkhCwKG | 0.26207 | 0.27851 | 0.93056 | 0.92768 |
| ftjob-ORbrKicdOcpRWKUqcFkhCwKG | 0.26384 | 0.28382 | 0.92222 | 0.92634 |
| ftjob-MWW3uqYEdflxMO1kCb40D8K4 | 0.28921 | 0.27074 | 0.92171 | 0.93118 |
| ftjob-MWW3uqYEdflxMO1kCb40D8K4 | 0.27664 | 0.28335 | 0.92708 | 0.92596 |
| ftjob-MWW3uqYEdflxMO1kCb40D8K4 | 0.288 | 0.27439 | 0.925 | 0.92805 |
| ftjob-MWW3uqYEdflxMO1kCb40D8K4 | 0.27747 | 0.28564 | 0.92909 | 0.92292 |
| ftjob-ORbrKicdOcpRWKUqcFkhCwKG | 0.27163 | 0.29183 | 0.93041 | 0.92136 |
| ftjob-ORbrKicdOcpRWKUqcFkhCwKG | 0.272 | 0.29282 | 0.92618 | 0.92206 |
| ftjob-ORbrKicdOcpRWKUqcFkhCwKG | 0.28344 | 0.28182 | 0.92768 | 0.92703 |

Table 3: Checkpoints of fine-tuned models with the ten lowest sums of training and validation loss

The best model was identified by choosing the model checkpoint with highest accuracy out of the ten model checkpoints with the smallest sum of training loss and validation loss, see highlighted the model in table xy. This model had been tuned with the hyperparameters lrm of 6, batch size of 10, and 3 epochs.

Finally, this best performing model was presented 134 legislative proposals of the hold-out testset. Just as with the standard GPT3.5 model, the accuracy of the responses was calculated for only the party percentages, the majority guesses, and an overall number. The fine-tuned model showed near to perfect answers regarding the format, a substantial improvement from the improperly formatted or hallucinated answers which the model fine-tuned on the sample dataset gave. Only the correctly formatted answers were included in the calculation of the accuracy, resulting in a tendency to underestimate the impact of the fine-tuning. An exceptional accuracy of 0.91 for the recognition of the correct majority and a strong increase compared to the standard model to an accuracy of 0.23 for the party percentages were achieved. When the tolerance level was relaxed to 0.1, meaning a generated guess of 0.98 instead of 0.9 is still considered correct, the accuracy for the party percentages increased to 0.43. See figure xy and xy for a comparison of both models’ performances and a random guess baseline performance.

Ein Bild, das Text, Screenshot, Diagramm, Reihe enthält.

Automatisch generierte Beschreibung

Figure 8: Accuracy of the fine-tuned model compared to the standard model and a random guess, tolerance level for predictions 0.05

Ein Bild, das Text, Screenshot, Diagramm, Reihe enthält.

Automatisch generierte Beschreibung

Figure 9: Accuracy of the fine-tuned model compared to the standard model and a random guess, tolerance level for predictions 0.1

**4.3 Exploring a Generation Task: Generating Counterfactual Amendments**

Having analysed how fine-tuning can improve ChatGPT3.5’s ability to detect political direction in text, one application of the model is explored. This analysis can only be exemplary in the scope of this thesis. Systematically analysing counterfactual versions of bills in a qualitative or quantitative way is thinkable using text analysis methods, using Spirling and Peterson’s method of accuracy as a sign for oppositional political positions, or following Bakker’s example of a Reward model, all approaches offering interesting possibilities for further research. This paper aims to illustrate one pilot example with the method explained above.

First, the standard ChatGPT3.5 model was fine-tuned on the rejected amendments, testing 9 hyperparameter combinations. Again, higher numbers of epochs than the recommended number calculated via the number of tokens were found to not notably increase the validation accuracy or decrease the loss and lrm of <1 or >10 decreased performance. Since the number of epochs are driving the costs of fine-tuning most significantly, for the full dataset, only the recommended number of epochs is tested with different lrm and batch sizes.

The hyperparameters lrm of 5, batch size of 1, and 4 epochs were found to deliver the best combination of training and validation loss at the checkpoint of step 50, see annex xy figure xy and xy for the full analysis of hyperparameters. Accuracy is not considered a very informative measure here, since the training set is small, and the evaluation is better done qualitatively.

The hyperparameters that worked best for the standard GPT3.5 model were adopted to fine-tune the classification model. This second fine-tuning process led to a satisfactory training and validation loss of 0.44 and 1.44, respectively.

The single fine-tuned model and the classifier fine-tuned model were prompted to generate counterfactual amendments with three iterations. The two batches of generated amendments were compared to the original amendments based on the method of structuring content analysis and on the number of categories they feature. Table xy displays the number of codes found in each group of amendments.

|  |  |  |  |
| --- | --- | --- | --- |
| Category | Number of coded segments | | |
|  | In original amendments | In generated amendments with fine-tuned GPT3.5 | Generated amendments with double fine-tuned GPT3.5 |
| Framing of migrants | Total: 21  Left: 7  Right: 14 | Total: 2  Left: 0  Right: 2 | Total: 4  Left: 0  Right: 4 |
| Stance towards European integration in the context of migration | Total: 19  Left: 3  Right: 16 | Total: 1  Left: 0  Right: 1 | Total: 4  Left: 1  Right: 3 |
| Stance towards economic migration and labour rights of migrants | Total: 9  Left: 3  Right: 6 | Total: 1  Left: 0  Right: 1 | Total: 0  Left: 0  Right: 0 |

Table 4: Number of coded segments per category with qualitative content analysis

After fine-tuning, the models both mainly reproduce the amendments’ format, as most answers began with “A right majority would propose”. Out of the 18 generated answers however, 4 were incorrectly formatted in a way that made the amendments uninformative, two others were too short to be interpreted in a meaningful way without any context.

The double fine-tuned model generated less than a fourth of amendments of each category compared to the original amendments, see table xy. The few answers that could be assigned to a subcategory of immigration were short, sometimes incomplete, and of little practical use. E.g. the answer “"Delete the part 'persons in distress at sea and” might reflect the stance of a right-wing party but leaves many questions about context and actual impact. Other answers which displayed some subcategory evidenced improbable party positions, e.g. a right-wing majority proposing to add to a proposal on a European network for labour mobility: “focus on filling transparent and non-discriminatory vacancies”. In the original amendments, the opposite political position becomes apparent.

Clearly the dataset’s size was insufficient to let either of the models reliably reproduce the desired format. Due to the quality of the results, no reasonable qualitative comparison can be made between the single fine-tuned and the double fine-tuned model. Neither did the double fine-tuned model draw from its previous fine-tuning of classifying legislative proposals to generate more informed answers in a notable way. With this exploratory analysis, the above-mentioned hypothesis could not be supported but room for further research is identified.

Hier man nur ein **Auszug aus dem Annex**, weil ich die Hyperparameteranalyse so oft erwähne:

**Hyperparameter Testing for Classification Task, Loss, Sample Dataset**Ein Bild, das Text, Reihe, Screenshot, Diagramm enthält.

Automatisch generierte Beschreibung

Figure 11: Loss for different hyperparameters on a sample dataset for the classification task of detecting political stance

**Hyperparameter Testing for Classification Task, Accuracy, Sample Dataset**

Ein Bild, das Text, Reihe, Diagramm, Screenshot enthält.

Automatisch generierte Beschreibung

Figure 12: Accuracy for different hyperparameters on a sample dataset for the classification task of detecting political stance

1. The OpenAI website on fine-tuning advises developers to first exploit all possibilities of prompt engineering to assess whether fine-tuning is necessary (OpenAI, n.d.-b). [↑](#footnote-ref-1)
2. See definition of Alex et al.: “the capacity to complete a task given a small number of demonstrations“ (Alex et al., 2022, p. 1) [↑](#footnote-ref-2)
3. The dataset was kindly made available by Dr Simon Hix, who collected the data with Doru P. Frantescu during the project VoteWatch, see https://www.votewatch.eu/ [↑](#footnote-ref-3)
4. Official antecedent party group names were coded with the most recent party group name. Party groups that not officially succeeded a previous group but follow the same political direction were coded with a combined name of both groups, e.g. EFD/IDG. [↑](#footnote-ref-4)
5. Keywords used for the search were “migration”, “asylum”, “visa”, “border”, “worker”, and “freedom of movement”. [↑](#footnote-ref-5)
6. The most commonly received definition of framing is to “select some aspects of a perceived reality and make them more salient” (Entman, 1993). [↑](#footnote-ref-6)
7. Sometimes, ChatGPT3.5 answered in a cautious way, highlighting that a thorough analysis of the policy was needed to make a prediction. Here one can see the limits of the standard model which was pre-trained to not be too certain about certain questions. [↑](#footnote-ref-7)
8. Example of illegible generated answer after fine-tuning with sample dataset: "EPS can/will/will not RESOLUTION (EC) No 416/15 IN THE ANNEX of the Agriculture about the over-the-counter (OTC) trade of derivatives, central counterparties and trade repositories. SPECIAL EPASE A, supporting A3 10/0078.nocing reporter LA ATNINTERNENT AGUSURTURE suppektor UPSPECIMRE off (1470 Yok./data” [↑](#footnote-ref-8)