Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

Deep Neural Network-based Voting Assistant

Bachelor's Thesis

Matthias Alexander Kleiner

makleine@ethz.ch

Distributed Computing Group
Computer Engineering and Networks Laboratory
ETH Zürich

Supervisors:
Zhao Meng
Prof. Dr. Roger Wattenhofer

February 26, 2022

Acknowledgements

First I would like to thank my advisor Zhao Meng for always taking time out of his days for regularly scheduled meetings, the always great discussions we had and most importantly for being a very supportive supervisor.

Further, I would like to thank Roger Wattenhofer and the DISCO Group for the opportunity to write my thesis with them.

The only other person that was regularly part of the process of writing this thesis is my good friend David Zollikofer, with whom I spent many days working side by side on our respective theses.

Lastly, a big thank you to my family for supporting me throughout my studies and the writing of my thesis in particular.

Abstract

Voting Advice Applications (VAA) have become an integral part of today's elections and voting procedures. They seemingly have not evolved a lot since their rise to popularity. Smartvote, a popular VAA in Switzerland, provided us with a large amount of data. With this data, from the Swiss national council elections of 2019, we push for innovation in the field of VAAs. Despite the large amounts of data available, machine learning has not yet established itself in the area of voting advice. In this thesis, we explore the possibilities given such data.

In order to achieve our goals, we apply natural language processing techniques to solve both existing and new tasks. A common task is to recommend to voters the party with which they have the most overlapping political opinions. This is our starting task to ensure our understanding and the functionality of NLP methods for this data set.

Most interestingly, we tackle tasks that could push democracy as well as VAAs forward, given the arising opportunities. We are able to finetune a Transformer based model, called Longformer[1], to predict answers for questions with very high testing accuracy, if the question is seen during training. The significantly more difficult task of predicting answers to questions unknown at the time of training also shows very promising results. We find the latter task to be very dependent on the chosen input context as well as the questions to be predicted. Therefore, the task outlines limitations of the data we use. Thus, encouraging the application of a model like ours for design choices when planning a questionnaire for voting advice. Our findings pave the way for new exiting possibilities in the domain of politics and advice systems, as we present a model that is able to learn political contexts and profiles, allowing it to predict future opinions.

Contents

Chapter 1

Introduction

In recent years, Voting Advice Applications have established themselves as an important source of information for voters, especially during election periods. Such tools are particularly attractive in countries where the voters have a large amount of parties and candidates to choose from. In general, it makes voting easier and less time-consuming for those that lack or are unwilling to spend the time to inform themselves more on their own. Hence, having access to such a VAA may be a deciding factor that moves people to vote due to the reduced effort necessary.

Looking at Smartvote, we see that the backbone of their system for the national council elections are the candidates that fill out a questionnaire. Based on a voter's answers to the same questionnaire, the candidates or parties that align most with the answers given are recommended to the voter. Because the answers are public information, it is difficult for candidates, that get elected, to change their opinions. Therefore, it is a good idea to base the recommendations on this overlap, as it makes the candidates' answers mostly reliable and representative information. Such VAAs also bring massive amounts of collected data with themselves, as all the answers voters give are stored and anonymized for later use. This data could for example lead to significant insights into an entire country's distribution of opinions and political orientations.

Of course, there are some discussions to be had in terms of political consequences. An example for this is that VAAs could influence opinions and voting outcomes depending on the implementation of the recommendation systems.

Due to the VAAs there exists a lot of data previously not available. This makes machine learning techniques a very effective and powerful tool to leverage advice systems to new possibilities. Given a questionnaire and its answers, we have many ideas, that we start to explore in this thesis, as to how this could shape the future of voting and possibly even democracy in its entirety. Imagine for example the following: At the beginning of each year, every citizen of a country can fill out a detailed questionnaire that will be used throughout the year to predict the opinions each person will have on a future topic that comes

up during said year. As a result, the reccuring voting would become obsolete. Additionally, it gives everyone the possibility to vote easily without having to regularly spend time informing themselves about new topics. Naturally, whether this is a desirable future is up for debate but given it is a possibility why should one not at least look into such a revolutionary direction.

The goal of this thesis is to discover the capabilities of the Smartvote data set when modern deep learning based techniques are applied on it. We start by exploring the comparatively easy task of party classification, meaning that given the context of the questionnaire's questions and answers, we want to predict which party this person is most likely affiliated with. This is very much in-line with what VAAs do nowadays. However, our goals are far beyond that task. Let's say we are given a set of answers to the questionnaire, where some sets of answers are incomplete. We now want to train a model that understands the context of all answers to the questions that are present and based on the political profile representation this builds it should predict the most likely answer that is missing. This described task is still on the easier side of things. Now, we assume that we are again given the same data but we want to predict answers to questions that are not present in the data set, meaning we want to train a model that builds a deep enough understanding of the political opinions, such that it is able to transfer this information to new questions. Thus, it would be able to predict what a persons' opinion on a future questions most likely will be. We will explore these settings in the course of this thesis.

If we achieve the above, we can say with certainty the future of democracy and voting holds a lot of possibilities and opportunities. In Switzerland for example, citizens are privileged to be able to vote on specific topics directly and thus have direct influence without having to be dependent on a representative for everything. Now let us consider another country where it may simply not be feasible to do something of the likes. There may arise new opportunities that incorporate the opinions of all the people without having to let them vote directly for everything, but instead by making them all answer a questionnaire a representative political profile distribution of the country could be created and thus the decisions taken could always be in accordance with this representation.

1.1 Background

We were presented with the opportunity to work with the large Smartvote[1] data set for the duration of this thesis. The data is based on a questionnaire that got filled out by the 2019 candidates of the Swiss national council elections and all the voters that sought voting advice. The idea behind the recommendation system used by Smartvote is that they compare a voter's answers to the ones of all the candidates and based on a way of ranking the similarity provide a list of most compatible candidates.

1.2 Literature Review

State of the art of VAAs currently are not machine learning based. Smartvote for example uses Euclidean distance in a multidimensional space as a measure of similarity between the candidates and the voter consulting the application. Some of the most popular and original VAAs are Stemwijzer[2] and Wahl-O-Mat[3]. The lack of research and application in such a data heavy environment calls for experiments applying modern machine learning techniques as a recommendation system or simply to learn new correlations from the available data.

During our literature review on related topics, we almost exclusively found literature concerned with the implications of VAAs or other social sciences focused topics. The most relevant publications are based on work from two previous projects at the DISCO group at ETH, namely "A Machine Learning Analysis of the Swiss Political Spectrum and Candidate Recommendation Process" by Bensland et al. (2020), as well as "Voting Smartly! Towards Assisting Voting Advice Application with BERT" by Zhong et al. (2021) and the available description of the recommendation calculation used by Smartvote.

The previous student applied a BERT language model to try to solve some tasks we will also address throughout this work. However, as those experiments are relatively basic with the exact preprocessing of data unknown, we explore the same tasks more thoroughly and expand them in multiple ways to find opportunities and also explore the limitations.

[1] https://Smartvote.ch by https://politools.net

1.3 Task Description

The goal of this thesis is to explore Deep Learning possibilities in the area of VAAs for elections and voting. In particular, we focus on adapting Natural Language Processing techniques in order to fit them to the tasks described in this section. While the data set contains more information that is not limited to only the questions and answers, such as level of education, political interest, age, self assessment on a left-to-right axis and many more. Our goal however, is not to use e.g., age or education related bias as a way of prediction but rather just the political profile representations the model will learn independent of who a voter or candidate is. While such additional information could certainly increase performance, it would most likely also lead to issues where the recommendations and predictions end up very biased and as such reinforce certain existing prejudice in society. Therefore, We concentrate our efforts to use only the questions and answers on the following two main tasks.

1.3.1 Party Classification

One of the most common, and also one of the original, goals of a VAA is to find the party that aligns best with the interests and opinions of the person filling out the questionnaire. Since we first need to explore the data set, it is a good starting point to try to understand the data in terms of correlation between answers and parties. Therefore, we decided on this being the first problem to tackle.

Our approach is to use the questions in combination with the answers as inputs. The data will be split into a train and test set. We train the model in a supervised manner on the train set, where the prediction is a class label corresponding to a party. This should then build an internal representation of the data we can use to classify a set of answers, from previously unseen test set data points, into a class representing a party.

1.3.2 Answer Prediction

This task takes everything a step further into a new direction for VAAs. Similar to party prediction, we use questions and respective answers as inputs together with an additional question that we want to predict an answer for. Thus, we split the questions into three subsets. The context set, the profile building set and the unseen set. The context questions are the questions which in combination with the answers of a data point build the context of an input. Profile building questions are used to train the model in a supervised way by predicting the answers for just these questions. Lastly, the unseen questions are the ones that we do not show the model during training but instead they are used to evaluate

the transferability or generalization of the trained model representations to new questions. This task can be interpreted as two different ideas. One of them being an incomplete answer set, for which the model should fill out the last missing answers depending on the given answers. The second and much more interesting interpretation is for the unseen questions. Assume we have a voter's answers to a questionnaire, we now want to predict what just that voter's opinion on a future topic will most likely be. As we already motivated in the introduction, such models could be very strong tools for future democratic systems and ease of voting.

If we achieve the goal of predicting answers for such new questions, it means that our model gains a meaningful understanding of correlations in within political topics. This is a big leap forward in the area of politics, since many systems today work with relatively simple multidimensional distance relationships. With a successful language model on the other hand, new opportunities arise to be explored for such correlations, based on language as well as the attention representation of transformer based model.

Chapter 2

Related Work

Our goal is to explore the applicability of NLP ideas to the Smartvote data set. Thus, the next section on Natural Language Processing models provides an introduction to the workings of the Longformer model[1] as well as its most important predecessors. This enables us to point out the differences with other models and why we choose this particular one as our starting point.

2.1 Natural Language Processing Models

2.1.1 Transformer

The following section is a summary of "Attention Is All You Need" [4] by Vaswani et al. (2017).

Vaswani et al. introduce the Transformer, which is a purely self-attention based alternative to recurrent neural networks. Simply put, self-attention computes a representation of an input sequence by weighted association of different input positions.

Like many transduction models, it relies on an encoder-decoder architecture as visualized in figure 2.1. The encoder computes a representation given an input sequence, and the decoder then generates the output element by element in an autoregressive manner. By comparing the predicted next token output to the original token at the respective position, the model is trained in a self-supervised way.

Both the encoder and the decoder consist of a stack of identical layers. In the encoder part, every layer has 2 sub-layers, firstly a multi-head self-attention and secondly a fully connected feed forward network based on the input positions. In addition, they use a residual connection around each sub-layer combined with a normalization layer, as visualized in figure 2.1. Thus, for both of the sub-layers, they get $LayerNorm(x + Sublayer(x))$ as the output. The dimension used is

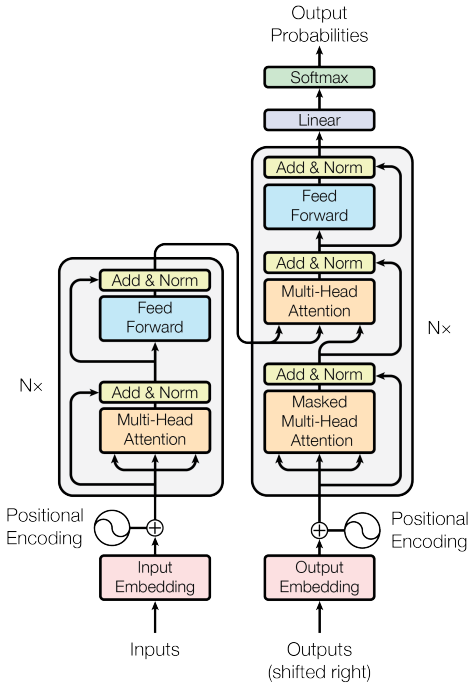
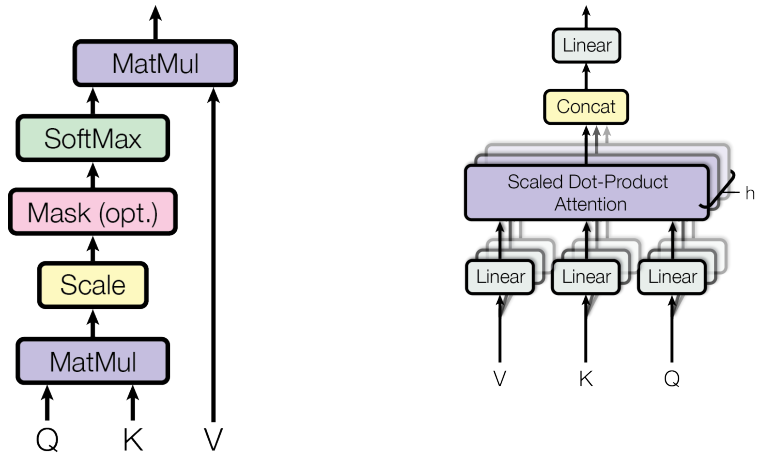


Figure 2.1: Transformer architecture. Illustration from[4].

consistently $d_{model} = 512$ to allow for the residual connections. The decoder employs the same sub-layers with the difference that the self-attention sub-layer is slightly adapted to ensure a position cannot attend to subsequent positions. Together with the fact that the output-embedding used as the decoder input has an offset of 1, this implies that predictions for any position only depend on the preceding positions. Obviously this is important, because otherwise the model could access the information it is supposed to predict.

Attention is in essence a function that maps a query and a given set of key-value pairs to an output. This output is a weighted sum of the values, where the weights are computed per value depending on the correlation of the query and key for the respective value. For the Transformer, they use scaled dot-product attention, as illustrated in figure 2.2a, with queries and keys of dimension d_k, while values have dimension d_v. Attention is usually computed in large batches, which is fast when using optimized matrix multiplications for query (Q), key (K), and value (V) matrices. In their paper, they introduce a scaling factor of $\frac{1}{\sqrt{d_k}}$ to counteract a drop-off in performance for larger values of d_k when compared to additive attention.

$$Attention(Q, K, V) = softmax(\frac{QK^>}{\sqrt{d_k}})V \tag{2.1}$$



(a) Scaled Dot Product (b) Multi-Head

Figure 2.2: Transformer Attention Implementation. Illustrations from[4].

They further find the attention to perform better if, instead of using a single attention function, the queries, keys, and values are first projected h times with different learned projections before applying attention to all projections in parallel. The outputs of that are then concatenated and linearly projected again, as demonstrated in figure 2.2b. This approach is named multi-head attention. They claim this idea enables the model to attend to different representation subspaces.

Like many sequence transduction models, they implement learned embeddings used to convert tokens to vectors of the dimension used by the model. Since, in this purely attention based mechanism, there exists no inherent positional information, they include it by adding a positional encoding to the input embeddings.

2.1.2 BERT

BERT is one of the most well known Transformer based models and the stepping stone for many models that came after it. Therefore, we include a brief introduction to the findings in "BERT: Pretraining of Deep Bidirectional Transformers for Language Understanding"[5] by Devlin et al. (2019).

Previous models, like the Transformer, are limited by only allowing a token to attend preceding tokens. While this may work fine for autoregressive tasks, it is clearly harmful when context from both sides of a token is necessary for a deeper understanding. BERT, which stands for Bidirectional Encoder Representations from Transformers, introduced a masked language model (MLM) pretraining objective to alleviate the unidirectionality constraint. In MLM, some tokens of the input sequence are replaced by special [MASK] tokens and the goal for the model

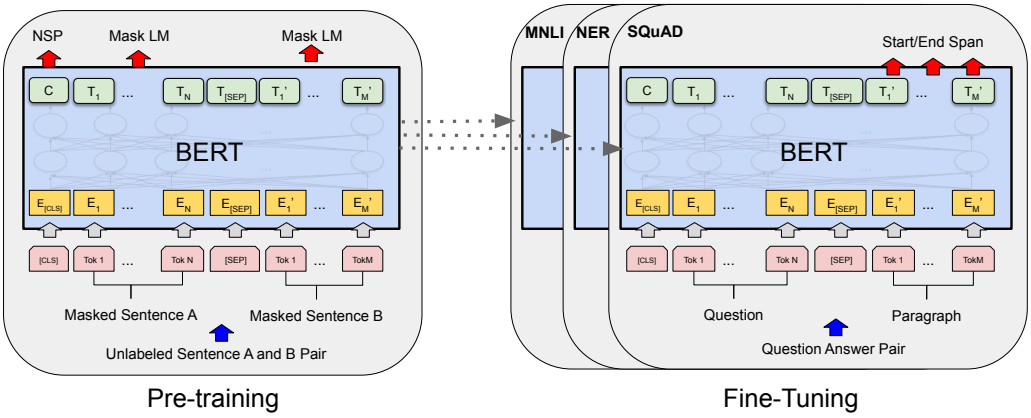


Figure 2.3: BERT Split Training. Illustration from[5]

is to predict which tokens were originally at the respective locations. In addition, next sentence prediction (NSP) is simultaneously learned as a secondary objective. For the NSP objective, the model's input always consists of two sequences concatenated with a [SEP] token. The goal is to predict whether the second sequence immediately follows the first sequence in the original text.

A main idea of BERT is to employ a two-step training process: pre-training and fine-tuning, as visualized in figure 2.3. They first pretrain the model using a large corpus of text in a self-supervised fashion by leveraging their MLM and NSP ideas. The training is now continued from this checkpoint by finetuning the model for a specific task. Such a split procedure allows for the model not needing to learn language understanding from scratch every time. Hence, comparably few parameters need to be learned during finetuning which enables data sets that are relatively small to be usable with language models.

The architecture they use for BERT is almost identical to the original Transformer encoder[4]. The BERT base model employs the following: number of Transformer blocks $L = 12$, hidden size $H = 768$, number self-attention heads $A = 12$, and a maximum input sequence length of 512. An input sequence always has a special classification token [CLS] at the first position. It is used to summarize the representation of an input sequence for classification tasks, such as NSP. If an input consists of two sequences, it is separated by the [SEP] token. They further add a learned embedding to each token, which indicates if it belongs to the first or second sequence. The input is therefore computed by acquiring the tokens embeddings, adding the segment embeddings and the position embedding to each token, as visualized in figure 2.4.

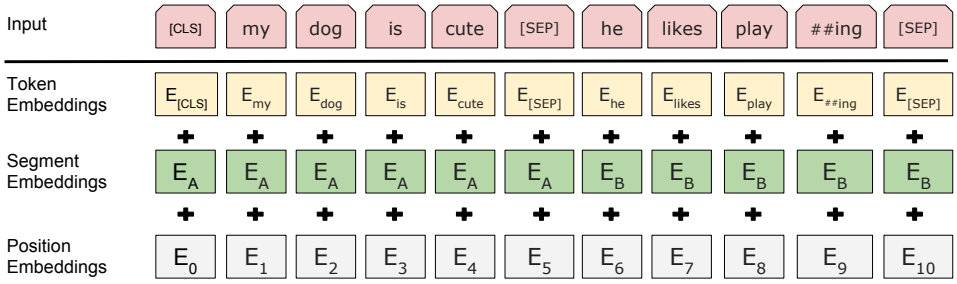


Figure 2.4: BERT Input Embeddings. Illustration from[5]

The extension of the Transformer BERT brings to the table is an important one. It achieved state of the art performance on multiple tasks, while utilizing a pretrained model which is only fine-tuned on the respective task. This implies that large text corpora of unlabeled data can be used for pre-training and thus a lot of redundant computation can be avoided by fine-tuning from the same checkpoint.

2.1.3 RoBERTa

RoBERTa is an important intermediate step for us because Longformer uses the pretrained RoBERTa checkpoint, to continue training from, in order to avoid the costly pretraining. The following section is based on "RoBERTa: A Robustly Optimized BERT Pretraining Approach"[6] by Liu et al. (2019).

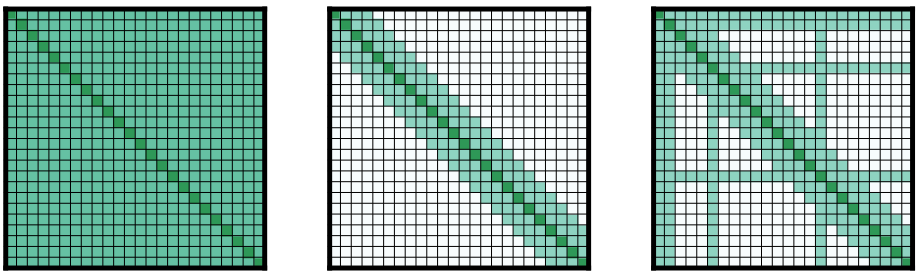
In the paper Liu et al. claim that BERT is significantly undertrained and as a consequence propose how to improve. To achieve this, they keep the architecture the same as for BERT.

BERT preprocesses the data corpus used for pretraining once, meaning the mask used in MLM is static. To avoid this, they propose to compute the mask dynamically every time a sequence is passed through the model. According to them, it should be an especially big advantage when the number of epochs is large. Further, they question how useful the NSP is. Through a set of experiments with different structures of choosing the input sequences in combination with NSP or without NSP, they conclude the following: Using single sentences performs bad, since the model very likely does not learn long-range connections. Full input sequences even across documents performs slightly worse than using full sequences, where they add an exception if it would be across multiple documents. Furthermore, removing the NSP loss slightly improved performance on downstream tasks. They sugggest that it will be advantageous to use Byte-Pair Encoding (BPE), which is a mixture of character and word-level representation. BPE is built on statistical analysis of training corpora to find relevant subword

units instead of entire words. Keeping track of large vocabularies is consequently enabled by using a new BPE approach of building on bytes instead of unicode characters as the base subword units. All of the above, combined with a few more small additions, builds the configuration called RoBERTa.

2.1.4 Longformer

This subsection is dedicated to introduce and summarize the for us most important model, since their pretrained model is our starting checkpoint for finetuning. We present in the following a summary of relevant parts from the "Longformer: The Long-Document Transformer"[1] paper, by Beltagy et al. (2020).



(a) Full (b) Sliding Window (c) Window and Global

Figure 2.5: Attention Strategies. Illustrations from[1].

Beltagy et al. follow the same approach of pretraining once and subsequently finetuning for many tasks while consistently outperforming RoBERTa. Despite the Transformer's success being partially thanks to the self-attention mechanism, it brings inherent problems of computational requirements with itself. Most notably, the quadratic growth in memory needed w.r.t. the sequence length. Those requirements essentially render such models infeasible for long sequences. As a result, the typical maximum input sequence length is 512 tokens. If such a model is supposed to be used on a task that requires longer inputs, the input has to be split up, and the outputs need to be recombined. This leads once again to the problem of highly task specific model architectures that the whole pretraining and finetuning paradigm intends to avoid. Of course, it most importantly reduced the model's ability to capture long contexts across the splits.

In the paper, they introduce their idea to sparsify the full self-attention matrix by explicitly specifying input location pairs attending to each other. First, given the importance of local context, they use a fixed-size window attention around every token, called sliding window attention. When layering many such attention layers, it results in the last layer being able to reach all input locations and thus compute representations that include context spanning even a long input sequence. For a fixed window size, w each token attends to $\frac{1}{2}w$ tokens on

both sides. This idea results in a complexity of $O(n \quad w)$ scaling linearly with the input sequence length n. The receptive field, as they call the possible reach at a certain layer, at the last layer is $l \quad w$, for transformer with l layers.

Furthermore, the dilated sliding window is introduced, which works analogously to dilation in CNNs, and thus further increases the receptive field using the same amount of computation. This dilated idea is not present in the model we use, as it requires a custom CUDA kernel to be efficient. The third attention concept they make use of is global attention, which can be set for specific token locations. It can attend all locations, while all other locations also attend to it. This idea can be included while modelling a task input, as it may be helpful to have global attention on the [CLS] token used for sequence classification or on an entire question for question answering, given a long context sequence. Since the amount of global attentions used is supposed to be a fixed small amount, this changes nothing about the linear complexity w.r.t input sequence length. The addition of global attention requires a slight change of the linear projections used in regular attention. This is solved by simply using two sets of projections, one for global attention and one for the sliding window attention.

One of the main motivations behind their work is to create a pretrained model that is able to handle long documents and can be used to finetune on many tasks. Their implementation can take input sequences with lengths of up to 4096 tokens, which is 8 times more than BERT. They actually used the released RoBERTa checkpoint to continue training on, in order to save cost and time resulting from MLM training. Since RoBERTa uses learned position embeddings with the maximum being 512, they decided to repeat them 8 times instead of initializing the rest randomly, as this significantly reduced the time necessary until it converges. Everything combined, the resulting model consistently outperforms RoBERTa on downstream tasks. For more implementation or methodology details please refer to the original paper[1].

The inputs we intend to use to finetune a pretrained model checkpoint are far beyond 512. Further, we assume that context spanning across the entire input sequence will be important to compute meaningful representations. The Longformer allows us to make use of those features, while the computational requirements do not skyrocket. However, it nevertheless needs to be mentioned that despite Longformer's complexity scaling linearly w.r.t. the input sequence length, the fixed window size they use is 512. This implies that a sliding window of size equivalent to the input sequence length of BERT is used in addition to some more global attention locations. Thus, the actual requirements of the model will still be a higher than those of a BERT model.

2.2 Smartvote Data

The amount of literature and related work in the technical area concerned with VAAs or the Smartvote data is very limited. The most relevant is a previous student, who worked with the same supervisors on a similar topic, which however has very limited content. There was also a group project conducted in the DISCO group that explores the Smartvote data set but is otherwise not related[1].

2.2.1 Official Smartvote Recommendation Method

The official Smartvote recommendation method is publicly accessible[2]. In the following, we provide a short summary.

Smartvote compares answers given by users of its platform to all the candidates who filled out the questionnaire in order to create a ranking, that represents the overlap of political profiles, which is then presented to the user. The voter can choose answers from a discrete set of answers depending on the type of question. All the answers represent a value between 0 and 100. Further, the voter has the option to assign a weight to a question, representing how important this question or topic is to them. There are the options "+", "=", and "-" which behind the scenes represent a scaling factor of 2, 1, and 0.5 respectively.

The calculation uses the Euclidean distance, where in a first step the distance, of all answers provided, between a candidate (c) and a voter (v) is calculated.

$$Dist(v,c) = \sqrt{\sum_{i=1}^{n}(w_i \quad (v_i \quad c_i))^2} \tag{2.2}$$

Where $Dist(v,c)$ is the distance between the voter and the candidate over the i questions, v_i is the voter's answer, w_i the voter's weight and c_i the candidate's answer to question i. Additionally, the maximal possible distance between candidate c and voter v over all questions answered by the voter is computed:

$$MaxDist = \sqrt{\sum_{i=1}^{n}(100 \quad w_i)^2} \tag{2.3}$$

Where MaxDist is the maximal Distance between the voter and the candidate over n questions and w_i being the weight for question i chosen by the voter Finally, the found measure of distance is transformed to a percentage between 0 and 100, by applying normalization

$$Matching(v,c) = 100 \quad \left(1 \quad \left(\frac{Dist(v,c)}{MaxDist}\right)\right) \tag{2.4}$$

[1] https://pub.tik.ee.ethz.ch/students/2020-FS/GA-2020-01.pdf
[2] https://Smartvote.ch/de/wiki/methodology-recommendation

They assess their recommendation methodology as follows: The recommendation is a purely mathematical and thus a politically neutral measure. It is a comparatively simple method of calculation that can be easily understood, also by users without a special mathematical or statistical background. Using the publicly available answers of the candidates and the user's own answers, one can easily recompute the recommendation to verify it.

Chapter 3

Methodology

First off, in this chapter we give a detailed overview of the Smartvote data set, to then continue explaining all the important findings regarding it. The second part consists of the Longformer finetuning implementation details.

3.1 The Data

The data was collected in 2019 during the time of the national council elections in Switzerland, as part of the Smartvote VAA that provides suggestions regarding which candidates have the closest political profiles when compared to one's answers.

3.1.1 Structure

The data is part of a questionnaire consisting of 75 questions. Smartvote encouraged the political candidates to fill out the questionnaire, which results in the first part of the data set, the candidate set. The second part of the data consists of the voter's answers from using the recommendation tool.

The questionnaire contains three types of questions: "Standard 4", "Slider 7", and "Budget 5". Out of all questions, 60 are Standard 4, which have the answer options 0 = no, 25 = rather no, 75 = rather yes, and 100 = yes. Further, 7 questions are Slider 7, meaning that the answer options are 0 = completely disagree, 100 = fully agree, with 17, 33, 50, 67, and 83 being intermediate positions. The remaining 8 questions are Budget 5, asking how the amount of money spent on something should be changed, with 0 = significantly less, 25 = less, 50 = same amount, 75 = more, 100 = significantly more. It is also possible to only fill out the rapid version of the questionnaire that consists of merely 31 questions. We will however not be making use of data points that have a lot of answers missing.

In addition to the answers, there are information fields different for each respective part of the data set, or were anonymized for the voter part. Such fields include the party affiliation, occupation, education level among others. For the purpose of this thesis, we will only use the answers and the party affiliation. It is important to mention that the party affiliation is a required and reliable label for the candidate set, whereas for the voter set it is voluntary information that can be entered by choosing from a drop-down menu that was anonymized before we received access to the data. Thus, for the party classification task using candidate data we have access to accurate labels in contrast to the voter data. Despite this issue, we train a model using the voter data. Further, the candidates had the option to include a written comment for each question they answered, and the voters could give each question a weight representing how important a topic is to them.

3.1.2 Statistics

The candidate set consists of 4'663 entries, out of which 3'926 are complete, meaning the candidate answered all questions, and out of those, 3'913 are confirmed by Smartvote. For the 3'926 complete data points, the average number of comments is 585 per question. However, the comments are mostly in German, making it impossible to use for us, since we rely on a model pretrained for English. As expected, the number of candidates per party is highly varied, since the candidates are representing a total of 69 parties. In the following, we will refer to the n parties that are represented by the most candidates as the top n parties. About 94% of the candidates are affiliated with the top 25 parties. Figure 3.1 shows the number of candidates listed for the top 25 parties. The "All" label stands for the distribution using the entire candidate set, the "Complete" labels uses the reduced set of all candidates that answered the entire questionnaire and "Combined Complete" means that only the complete data points are included in addition to the youth parties being merged into the respective main parties.

In comparison, there are 427'572 entries in the voter set, out of which 93'481 are complete and 198'111 data points have between 70 and 75 questions answered. The voluntary information regarding party affiliation is a little strange since the labels are: "1", "2", "3", "4", "5", "6", "7", "20", "23", "24", "25", "27", "28", "8888", and "9999". If one compares these with the options provided when filling out the questionnaire, one could potentially guess the actual parties. As previously mentioned, voters have the possibility to enter a weight to every question. Only 3'462 voters entered 50 or more weights, 29'352 entered 25 or more and 95'968 entered 10 or more weights. When using only complete answers it is even less, which implies that relatively few voters seem to have carefully filled out the questionnaire including the weights.

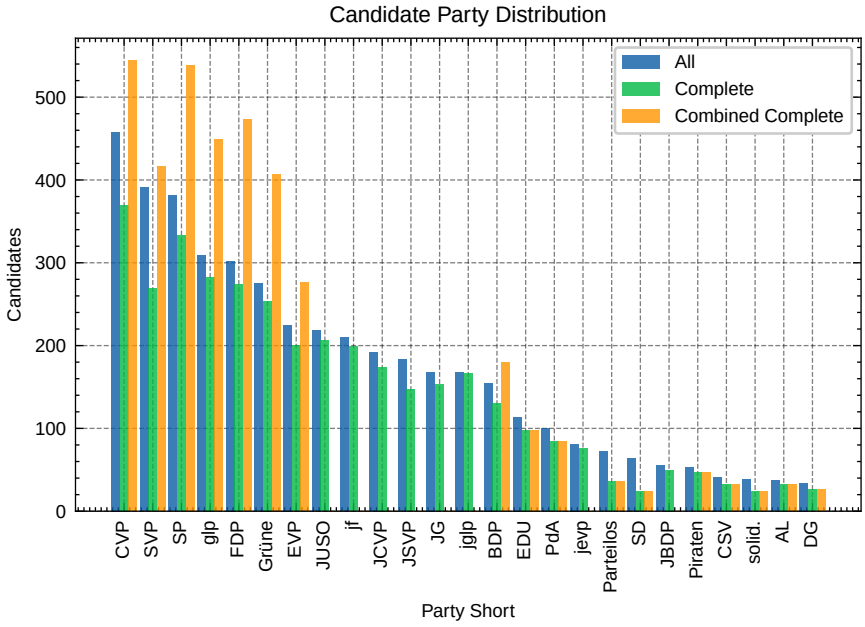


Figure 3.1: 3 versions of party distribution

Above, we introduced the three question types and the respective answer options. Since only 15 questions have a different type than Standard 4, the answer options of those questions, namely 17, 33, 50, 67, and 83 will be severely underrepresented. Later, we will take this into account when constructing the inputs.

3.1.3 Preprocessing

First, many questions contain "What is your position the following statement:" [sic] as a prefix, which does not contribute to the meaning of a question, so we strip those prefixes.

For both parts of the data set, we remove the data points that do not contain answers to all questions or have at most m N/A answers, with m being a variable that can be set for a specific execution. We also use an option to disregard all data points, that are not in the top n parties. The choice of the value n is especially important for party classification, as we will see later. For the voter set, choosing the top parties also includes removing the data points with labels "8888", "9999", or "N/A". This is because the drop-down selection allows for three ways to not choose a party, 1. not touching the option at all, 2. choosing "none", 3. choosing "other".

A very important step is to apply an answer simplification or reduction map to the answers. Firstly, because we may have to replace N/A answers, which we usually do by setting the answer to 50. Secondly, we noticed that, mostly for the answer prediction task, having too many possible answers makes the task too complex for either the information contained in the questionnaire and it's answers or for the model. Therefore, we used a couple different answer reduction maps. We define them here using names for later use.

- Identity = f0: 0, 17: 17, 25: 25, 33: 33, 50: 50, 67: 67, 75: 75, 83: 83, 100: 100, N/A: 50g
- Quarter step = f0: 0, 17: 25, 25: 25, 33: 50, 50: 50, 67: 50, 75: 75, 83: 75, 100: 100, N/A: 50g
- Ternary = f0: 0, 17: 0, 25: 0, 33: 50, 50: 50, 67: 50, 75: 100, 83: 100, 100: 100, N/A: 50g
- Binary = f0: 0, 17: 0, 25: 0, 33: 0, 50: 0, 67: 100, 75: 100, 83: 100, 100: 100, N/A: 0g.

The identity mapping is what we use for the party classification task, while the rest are for answer prediction.

For the answer prediction task, we split the questions into three disjoint subsets: 1. the context questions, 2. the profile building questions (PBQ), also called train questions, and 3. the unseen questions (UQ), also called test questions. The context questions are always part of the input, combined with a data point's answers. This constructs the context the model is supposed to infer from. The profile building questions are, one at a time, appended to the described context during training. We use those questions together with a train subset of the data points to learn representations of political profiles. Next, the model's ability to predict answers of the unseen questions, given a data point's context, is evaluated.

During our experiments, we found that the model performance is heavily dependent on these question sets. Therefore, instead of simply choosing the subsets at random, we had the following idea: First, we compute Pearson's correlation coefficient for each pair of questions, meaning the correlation of the answers to the respective questions, resulting in the correlation matrix which is provided as figure 3.2. We now compute an absolute average along each question, which we use as a measure of how correlated a question is with others, see figure 3.3. Of course, this is just a linear correlation and thus far from perfect. However, this gives us a way of making a more representative choice rather than a random one,

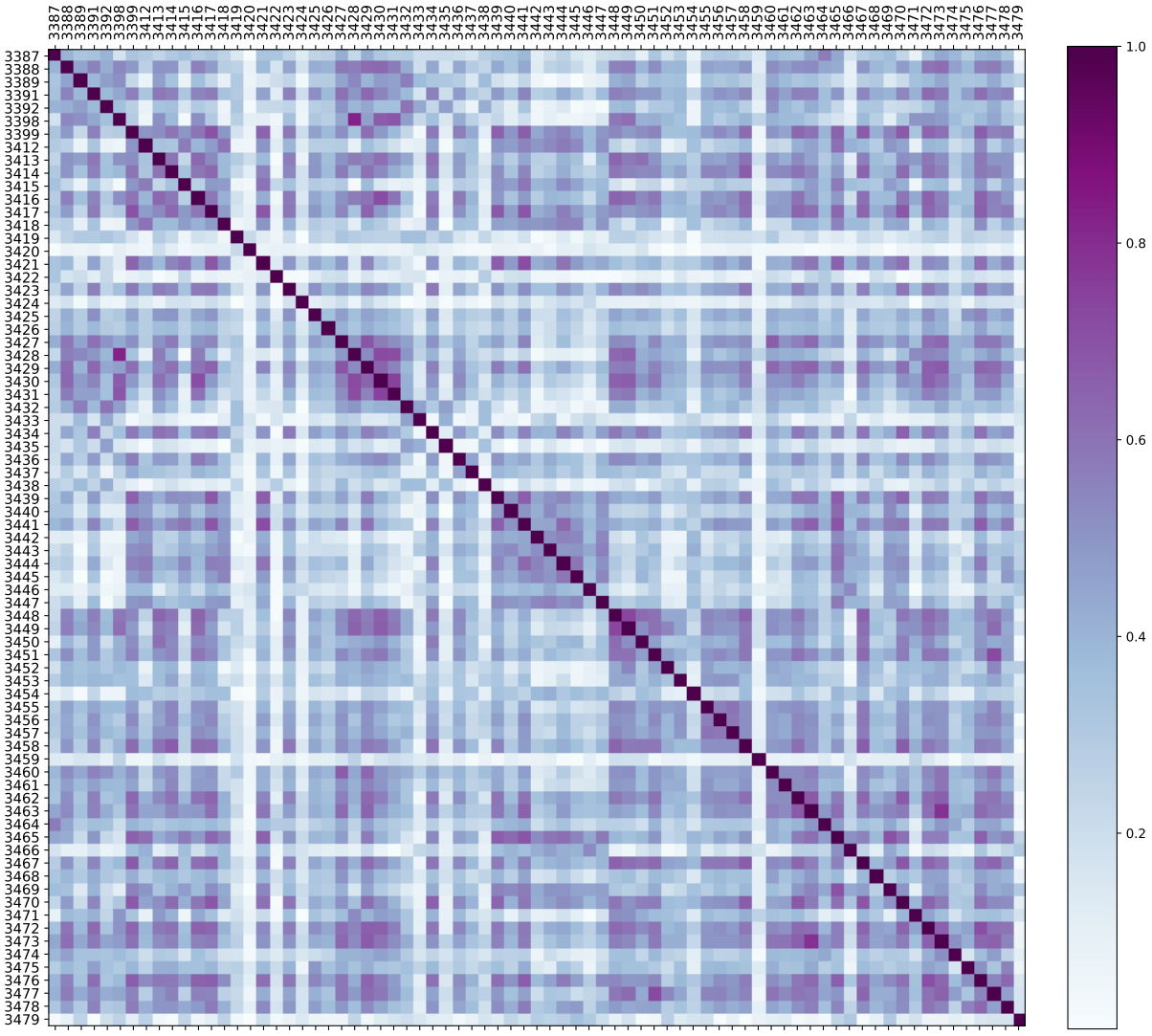


Figure 3.2: Pairwise Question Correlation Matrix, questions included in appendix B

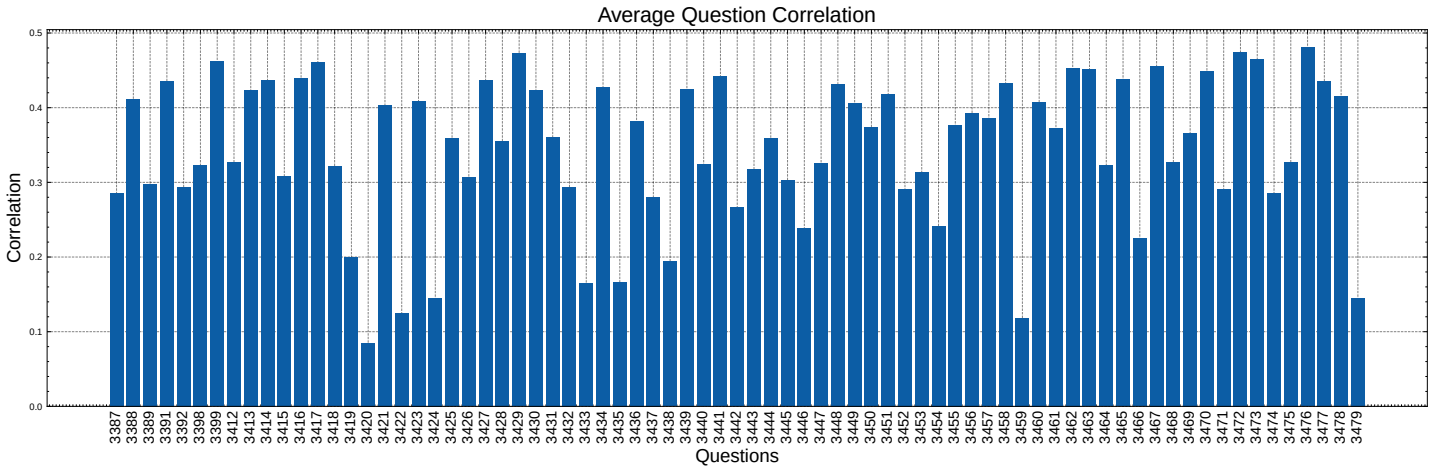


Figure 3.3: Averaged absolute correlation of question i with the rest.

while we now also have the ability to appoint a measure of entailed task difficulty to a subset choice Hence, we can now choose the questions with maximum correlation to achieve an easier, average correlation for a medium, and minimum correlation for a harder task difficulty setting.

Because of the way we train the model for the answer prediction task, we have a train data set, a test data set, the profile building questions and the unseen questions. This means we model two different test settings. The first one uses the test data to predict on the profile building questions and the second one uses the test data to predict on the unseen questions. Since the task to predict answers for unseen questions is a lot harder in comparison, we came up with an intermediate task. The idea is that we use the "tuner007/pegasus_paraphrase"[1] model, which is based on Pegasus[7] and trained for paraphrasing, to rephrase the profile building questions. Hence, we can additionally evaluate our model using the test data on the rephrased profile building questions (RPBQ). For example "Do you support an increase in the retirement age (e.g. to 67)?" is rephrased to "Do you think the retirement age should be increased?" and "An initiative calls for the introduction of paid paternity leave for four weeks. Do you support this proposal?" to "Do you support the idea of paid leave for fathers?".

3.1.4 Input Preparation

To model our inputs in a more structured way, that is immediately understandable for the model, we introduce <QUESTION> and <ANSWER> as new special tokens to the vocabulary. The other special tokens are: <s> the [CLS]

[1] https://huggingface.co/tuner007/pegasus_paraphrase

token, </s> the end of sequence token and <pad> the padding token. The idea of using the <QUESTION> and <ANSWER> tokens is to clearly separate the questions and answers with these tokens. We achieve that by putting a <QUESTION> token before every question and an <ANSWER> token before every answer.

For party classification, the input has the following structure: <s><QUESTION> Question 1 <ANSWER> Answer 1 ... <QUESTION> Question 75 <ANSWER> Answer 75 </s>. The input for answer prediction is almost the same. The only difference is that we use only the context subset of the 75 questions in a row in combination with 1 question to be predicted, without the answer at the end. Due to this difference, the amount of inputs is multiplied by the number of profile building questions or unseen questions respectively. Therefore, the time it takes to train the model for answer prediction is also multiplied by the same factor.

We have two ways of assigning global attention to tokens. For the first one we set global attention for all <Answer> tokens and the <s> token. The second way is for specific execution of answer prediction only. We set the same global attention as for the first one but additionally set it for all tokens of the question to be answered.

3.2 The Model

3.2.1 Longformer Finetuning

The model we use throughout our experiments is built on the Transformers library[2] by Huggingface[8] and PyTorch[3][9]. In particular, the pretrained model we decided to finetune is called "allenai/longformer-base-4096" and is provided by the authors of the Longformer paper[1].

As we already explained, there are a few main advantages Longformer has over other pretrained models. Longformer is a well-performing pretrained model that takes inputs of up to 4096 tokens. Thus, it can compute representation with large context spaces while the memory requirement does not scale quadratically with the length of the input sequence.

We model both tasks described as sequence classification tasks. This means that given a single input sequence, the model is supposed to assign it a class.

[2] https://github.com/huggingface/transformers
[3] https://pytorch.org/

For one task we intend to classify data points into parties. The second task is to predict an answer to a question from a discrete set of possible answers. The approach of modelling the tasks as classifications therefore seems fairly obvious. We can build on the LongformerForSequenceClassification from the transformers' library. It is essentially the same as the original Longformer with an additional linear layer, on top of the last pooling layer, which has $c = \#classes$ outputs. Hence, all the weights can be loaded from the pretrained model except for the last linear layer, which is initialized at random.

3.2.2 Training

In general we use 20% of the data for testing and the remaining 80% for training. When training for party classification we use a simple cross-entropy loss, the AdamW optimizer, $lr = 3 \ \ 10^{\ 5}$ with a cosine scheduler with linear warm up of 2 epochs and a cycle of 0.5, batch size of usually 32, mixed precision and gradient checkpointing. Training for answer prediction, we use cross-entropy loss with weights to compensate for uneven label distribution, the AdamW optimizer, mostly with $lr = 3 \ \ 10^{\ 5}$ and in some special cases $lr = 5 \ \ 10^{\ 6}$ using a cosine scheduler with linear warm up of 2 epochs and a cycle of 0.5, mixed precision, gradient checkpointing and a batch size of usually 64 which is only possible with either DistributedDataParallel or gradient accumulation. Most models were trainend using 1 or 2 Geforce RTX 3090 GPUs. The models for party classification can be trained in just a few hours, while for answer prediction our training takes between 24 and 48 hours, depending on whether we use just a single GPU or multiple, and how much data is used in a specific training run.

The goal was not necessarily to find the best model for a task, but rather to explore the data set and it's opportunities given that we use a long document transformer. Thus, we did not really do hyperparameter tuning and more so explored how a collection of setups performs.

Chapter 4

Results

4.1 Party Classification

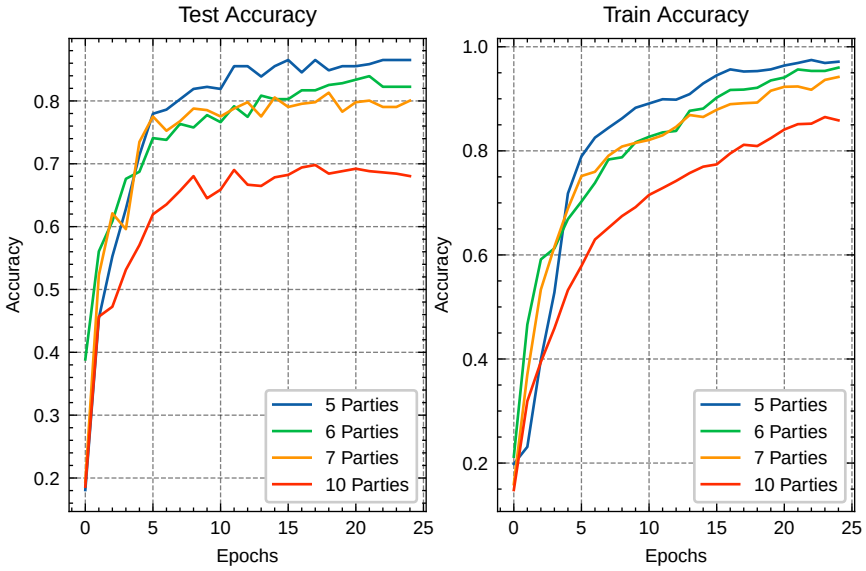


Figure 4.1: Candidate set accuracies

For Party classification on the candidate set, we trained our model using only complete answers and the top 5, 6, 7, and 10 parties. In figure 4.1 the development of the accuracies over 25 epochs are compared. As expected, when using more parties to predict on, the task becomes harder and hence the prediction accuracy lower. In table 4.1 we include a comparison to the previous student, who trained a BERT model with the top 6 parties, that achieved a test accuracy of 77.36%[1] while our Longformer based model reaches a test accuracy of up to 83.94%.

[1] Data cleaning and preparation may vary slightly

	CVP	SVP	SP	glp	FDP	Grüne	Total
BERT	68.42	92.83	87.10	75.00	70.00	68.42	77.36
Longformer	87.69	87.93	84.29	85.96	78.57	77.55	83.94

Table 4.1: Comparison of previous results and ours.

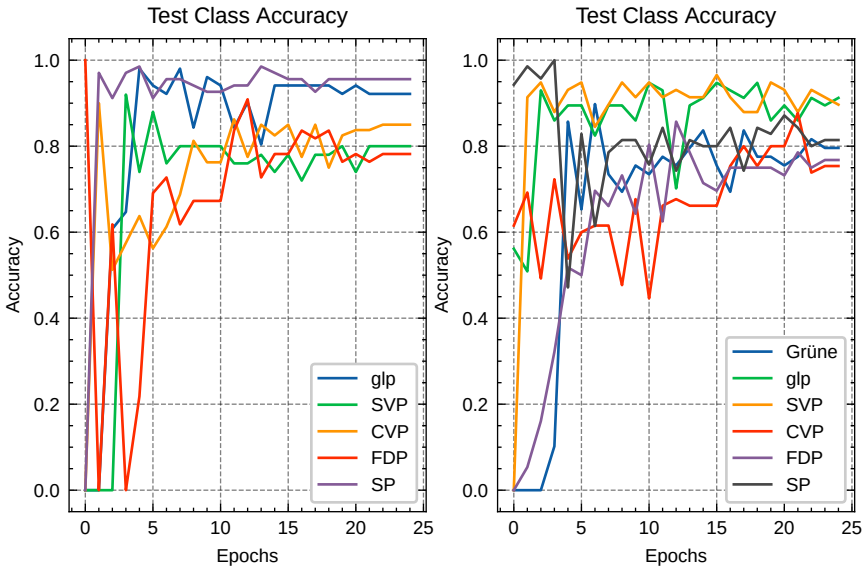


Figure 4.2: Candidate set class accuracies, 5 and 6 parties resp.

In figures 4.2 and 4.3 one can observe interesting changes in class accuracies when adding more parties considered. As an example, with 5 parties the accuracy for SP is above 95%, while when considering 6 parties it suddenly drops to only 80%. With the only new factor being that Grüne is added as a possible answer. When looking at the included 5 parties, we notice that SP is the only real representative from the left section of the political spectrum. Hence, making it relatively easy to distinguish SP from the rest. But when Grüne is included, the two parties overlapping interests become hard to distinguish with the information given.

This effect becomes even clearer when looking at the class accuracies for the results with the top 10 parties. There, some youth parties are included, namely JUSO, jf and JCVP. As one would expect, the youth parties have many shared opinions with the main party, making it very hard to distinguish the two. This issue is clearly visible in figure 4.3.

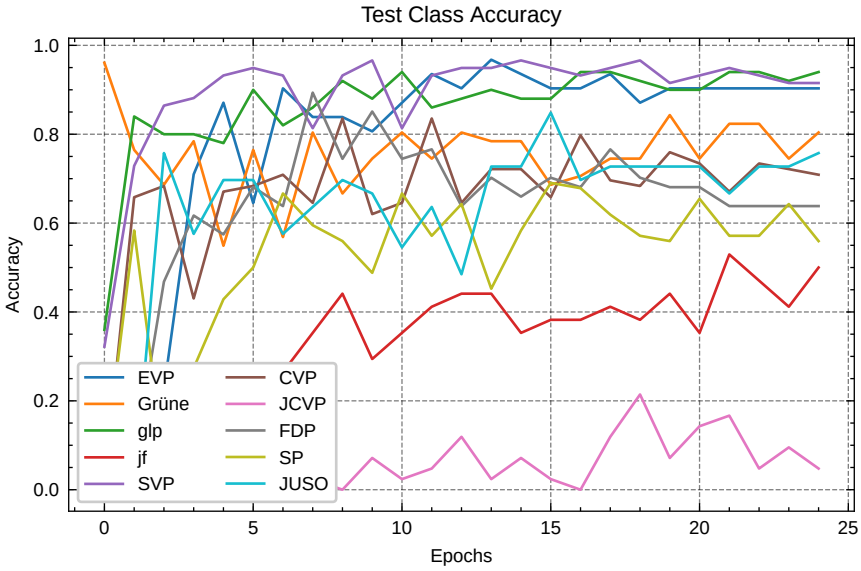


Figure 4.3: Candidate set class accuracies, 10 parties

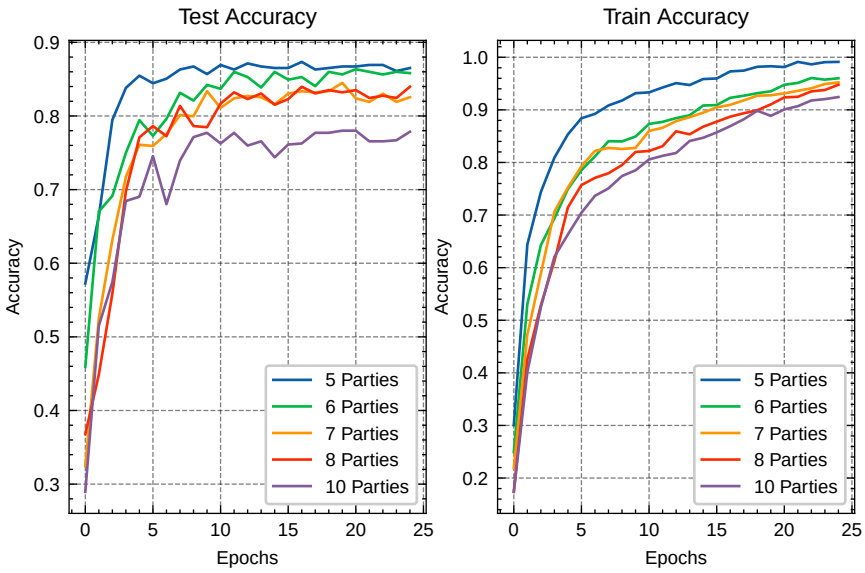


Figure 4.4: Candidate set accuracies (youth merged)

With this new gained knowledge in mind, we merge the youth and main parties that are present in the candidate set and retrain the model. When looking at figure 4.4 it is pretty surprising to see that even with up to 10 parties, the test accuracy is still about 78%. When comparing the top 6 party executions with and without the merged youth parties, one can see that with the youth merged, the accuracy achieved is up to 86% compared to up to 84% without.

Figure 4.5: Candidate set class accuracies, 6 parties (youth merged)

Figure 4.5 illustrates that the variance of class accuracies is smaller relative to the previous runs. Presented with the information contained in the questionnaire data, the model is still able to capture the differences of most parties relatively well, now that the youth parties are not listed independently anymore. However, we see that the limitations of the representation space starts to show, as the model struggles with learning to classify PdA and BDP correctly, as demonstrated in figure 4.6. Naturally, this may also imply that some parties do not have enough standout opinions differentiating them from other parties, or that those exact opinions are not covered by the questionnaire.

We did not experiment with the voter data to the same extent we did with the candidate data, because it was soon obvious that the voter set is very noisy in comparison. We train a model on the voter data with the top 5 parties. The results, shown in figures 4.7 and 4.8, are clearly worse than the ones achieved on the candidate set. There are probably many reasons for this, but a few come to mind immediately. A voter filling out this questionnaire does so, likely relatively careless in comparison to a candidate. Because for a candidate, the answers chosen could decide between being elected or not. Whereas a voter may just fill it out for fun or to try out different answers out of interest. Further, many voters do not align their political positions entirely with some parties, but rather pick and choose. Things like these lead to a more unpredictable and noisy data set, thus producing worse training results. Surely the results we present for this are far from optimal, but at this point our focus shifted to the answer prediction

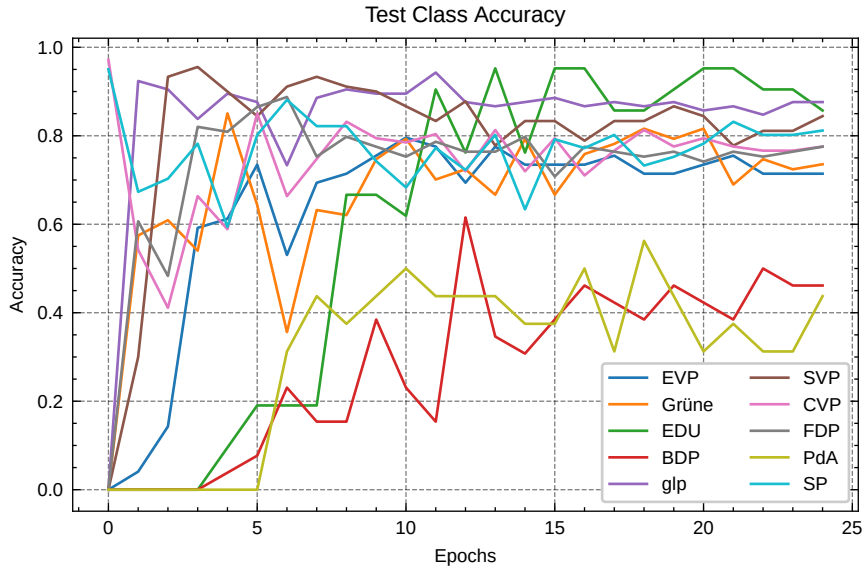


Figure 4.6: Candidate set class accuracies, 10 parties (youth merged)

task.

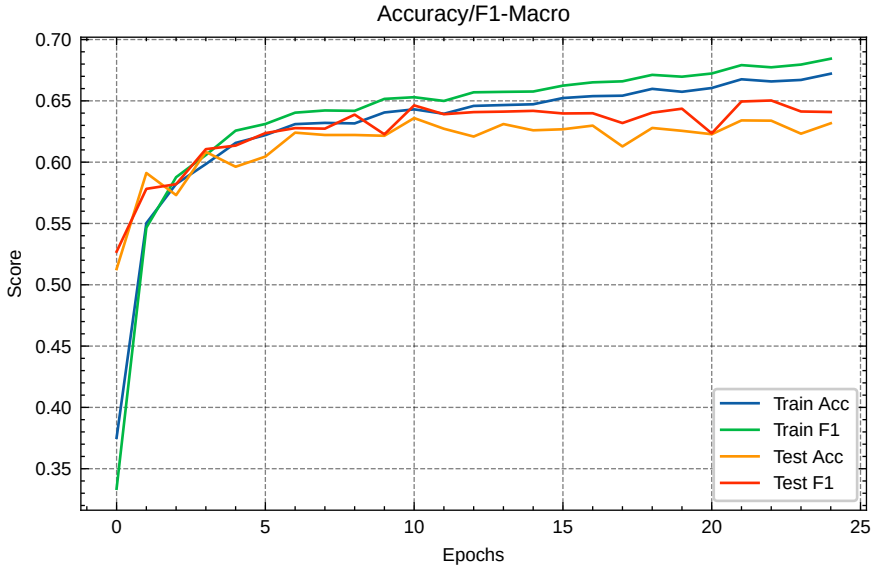


Figure 4.7: Voter set, scores

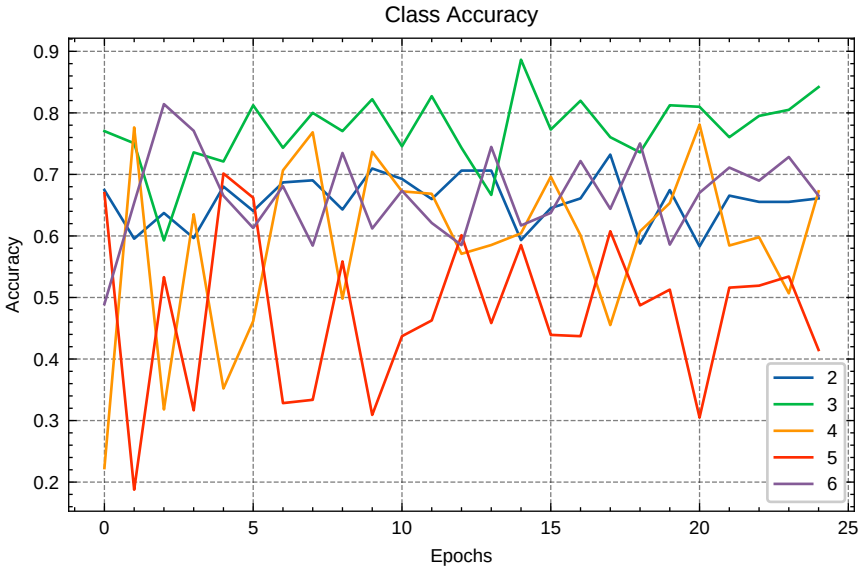


Figure 4.8: Voter set class accuracies, 5 parties

4.2 Answer Prediction

Throughout this section, we show two main kinds of combined results. On one hand, the setting where we run a specific setup three times, with the only difference being how we choose the three question subsets. The choices consist of an easy, medium and hard setting. This means that for the profile building questions and unseen questions we choose the maximally, averagely, or minimally correlated questions respectively. On the other hand, the setting where we compare the average baseline result to an adapted version using the same question subsets.

To find those correlation values, we compute the average Pearson correlation coefficient for each question, by computing it for all questions pair-wise and then averaging for each question. $AvgCorr_{Q_i} = \frac{1}{75}\sum_{j=1}^{75} PCC(Q_i, Q_j)$, where Q_i is the i-th question and PCC Pearson's correlation coefficient. For our experiments we decide on a profile building question set size of 10 and an unseen question set size of 10, leaving 55 context questions. In the case of the hard problem setting, we take the 20 questions with the lowest correlation. As a next step, we randomly sample 10 questions from those 20 and compute the difference in correlation of the two sets summed up. After sampling x times, we decide to use the two sets that had the smallest difference. The idea works analogously for the medium and easy setting.

In the following we call this first triple of executions the baseline, for min, avg, and max respectively. We found the answer prediction task to be hard in general, especially for unseen questions. Hence, similar to e.g., sentiment classification we only predict whether an answer is positive or negative, in our case 100 or 0. To do so, we apply the binary answer reduction describe in the preprocessing section. We set global attention for all <Answer> tokens as well as the <s> token. As previously mentioned, we usually have 4 combinations of data set and questions that we measure, with some exceptions later. Train set on PBQ, test set on PBQ, test set on RPBQ and test set on UQ.

Figure 4.9 immediately shows how the choice of question subsets impacts the prediction accuracies achieved by the model. It is pretty impressive that for the max correlation setting, the model achieves accuracies as good as they get for the test on PBQ after just 3 epochs. In comparison, for the min correlation, running it 25 epochs is helpful. Now we know that our model learns to predict answers of questions it has previously seen during training very well. The f1-macro score, as seen in figure 4.10 is almost always very closely correlated to the accuracies. Therefore, we will not include it if it contains no additional information.

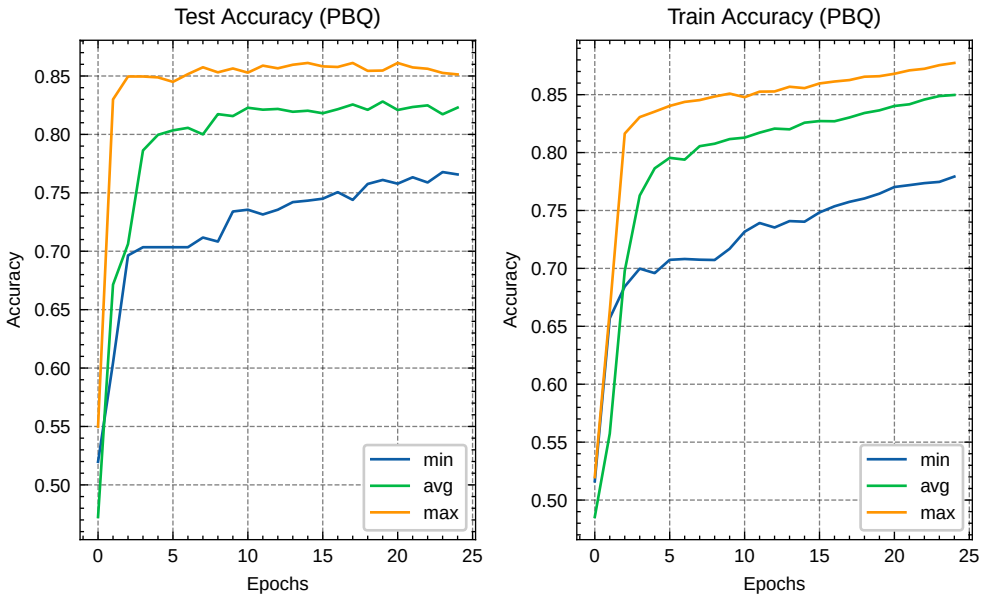


Figure 4.9: Candidate set accuracies baseline, test and train PBQ

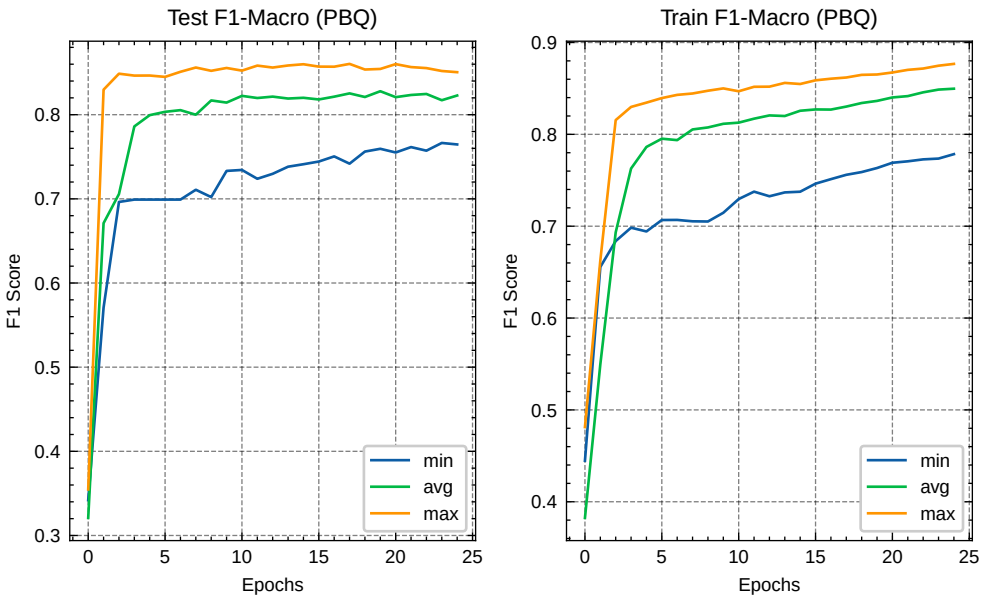


Figure 4.10: Candidate set f1-macro baseline, test and train PBQ

For the test on RPBQ and UQ, in figures 4.11 and 4.12, the effect of the correlation choice is no longer as clear. For the test on UQ the max setting still achieves decent accuracy, while the average case might as well be a random prediction. The min case seems to perform surprisingly well. However, when additionally considering figures 4.12 and 4.14 we realize that the model predicts only label 100 for a few epochs. In combination with the significantly lower f1 score, this implies that label 100 is more frequent than label 0, hence rendering the deceptively decent performance invalid. The representative results are therefore the ones nearing epoch 25. Nevertheless, it is surprising because the hard setting seems to learn a better transferable representation than the average setting.

The test results on RPBQ leave even more open questions, due to the average setting performing significantly better when predicting answers for the rephrased profile building questions. A possible explanation could be that using the max setting, the model needs to learn less about the actual meaning of a question to achieve good predictions and can instead rely a lot more on the existing correlation of answers. We conclude that the choice of questions leads to highly varied results and is therefore imperative. Be it when deciding on the set to train on or when designing the questionnaire.

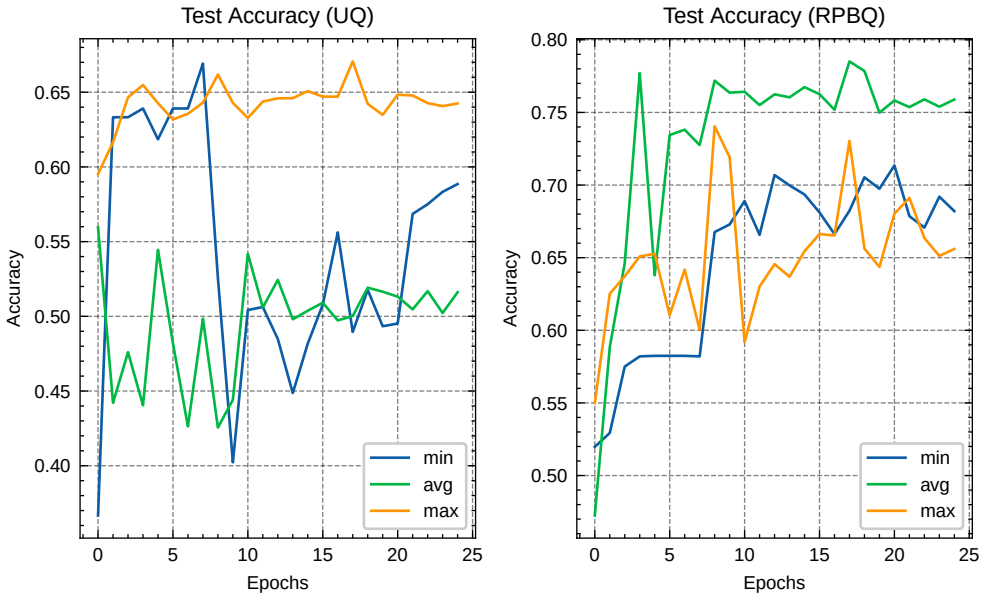


Figure 4.11: Candidate set accuracies baseline, test UQ and RPBQ

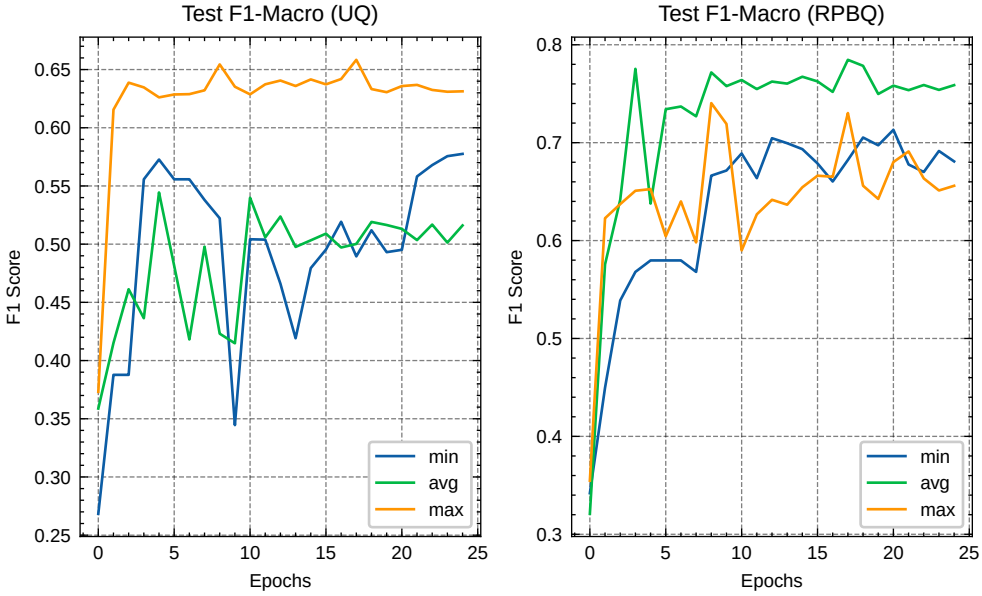


Figure 4.12: Candidate set f1-macro baseline, test UQ and RPBQ

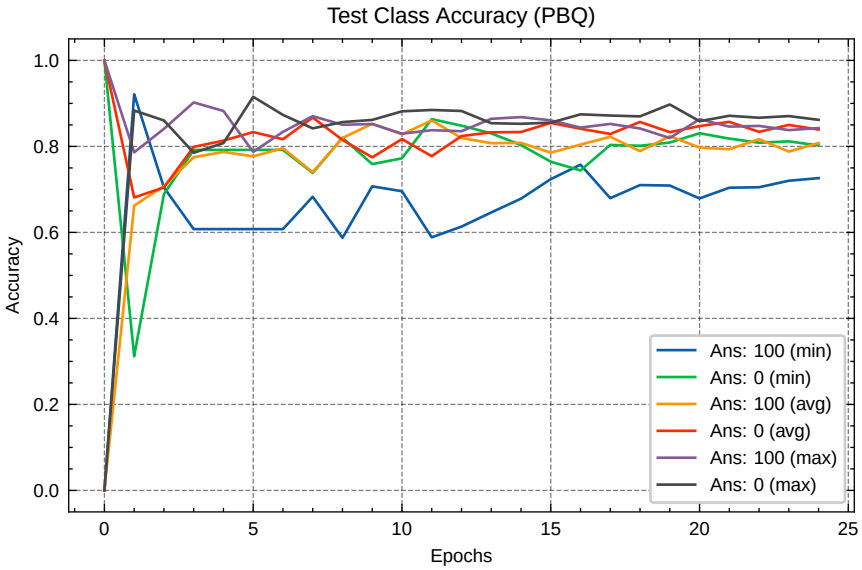


Figure 4.13: Candidate set class accuracies baseline, test PBQ

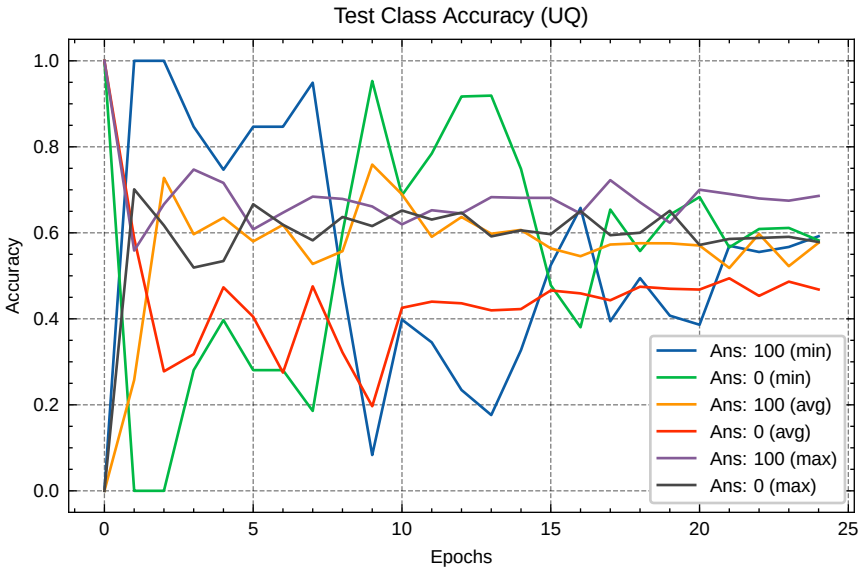


Figure 4.14: Candidate set class accuracies baseline, test UQ

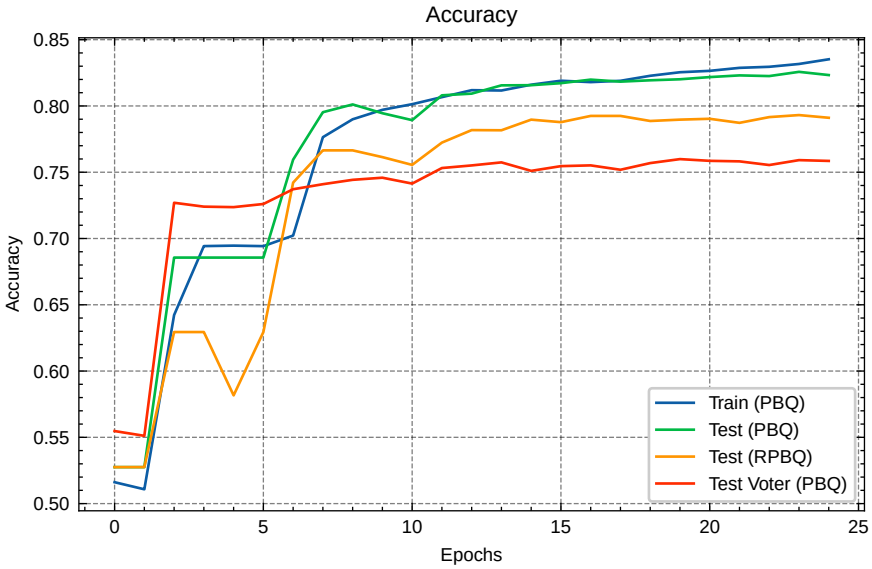


Figure 4.15: Candidate set accuracies incl. voter evaluation (PBQ)

We want to find out how a model trained on the candidate set generalizes to the voter set. To gain some insights, we train a model analogously to the average baseline. The difference is that we use a slightly smaller learning rate of 10^{-5} and additionally evaluate the model on a random chosen set of 3'000 complete voter data points. The result, as shown in figures 4.15 and 4.16, is that the model generalizes fairly well to the voter subset. The test on profile building questions achieves 76% accuracy compared to the candidate test set with 82.5%. The accuracy for unseen questions is already bad for the candidate test set, with 56% and even worse for the voter set with 50%. Assuming the model would perform better for the unseen questions w.r.t. the candidate test set, we believe it would also perform decently for the voter generalization on those questions.

The next idea we investigate is if restricting the number of parties considered, in the candidate set, has significant effects on the model's ability to learn political profile representations. The idea came to mind, since it is possible that candidates within the top parties have more unified and shared opinions per party. Whereas, considering too many parties could lead to the data becoming too complex. The model is trained the same as the avg. baseline, except for the fact that we reduce the candidate set to candidates from the top 10 parties only. In figures 4.17, 4.18, and 4.19, we compare the results with the average baseline. It is clear that this approach achieves significantly better results, as it should, due to the task presumably being easier. The accuracy increase is especially large for the UQ and RPBQ with about 7% and 9% respectively. Rather surprisingly, we observe that the class accuracies, on the unseen questions, for class 0 stays pretty much the same whereas it is a lot higher for class 100.

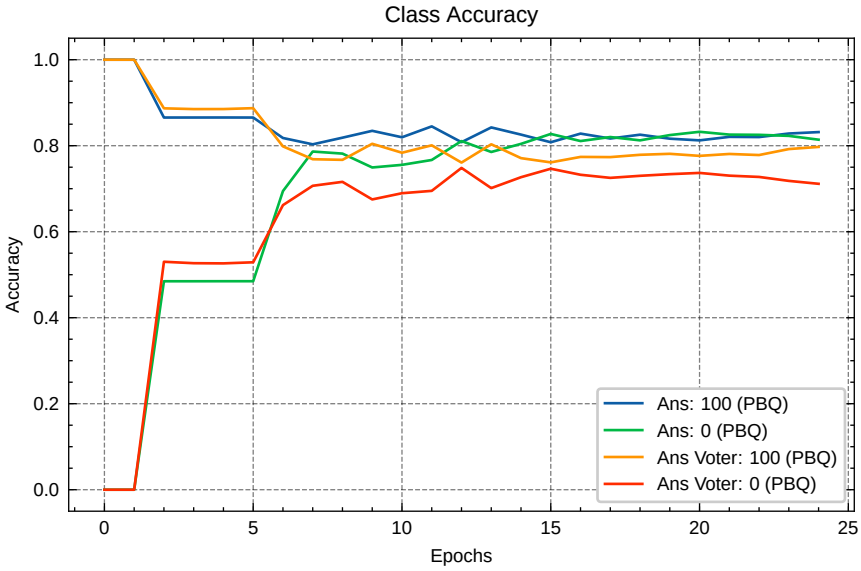


Figure 4.16: Class accuracies incl. voter evaluation (PBQ)

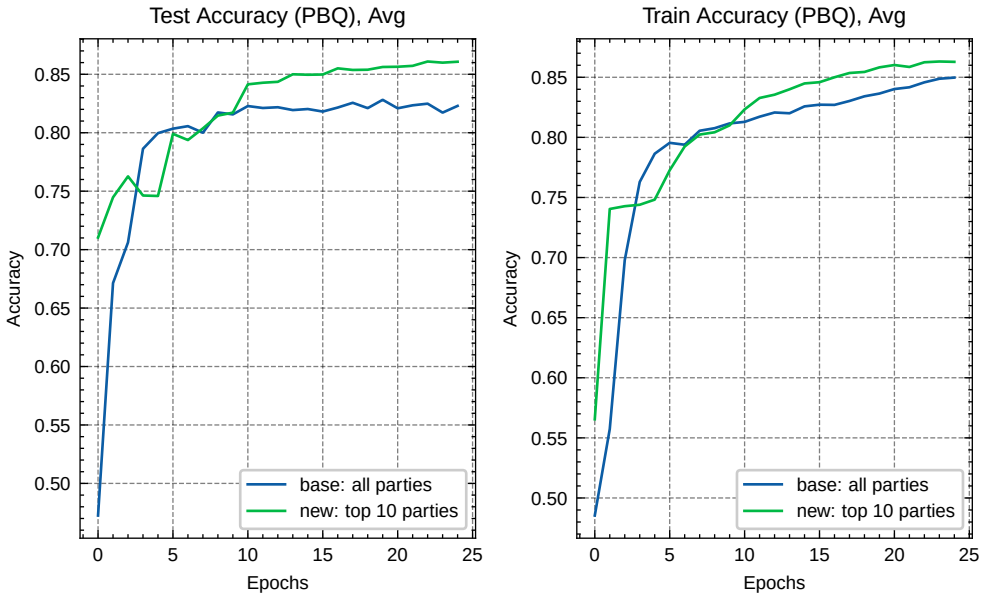


Figure 4.17: Candidate accuracies, avg baseline vs. avg top 10 parties

During the process of training multiple models with different settings, we notice that there are often large jumps in terms of adjustments. This makes it seem like the convergence, if it converges at all, is very unstable. Consequently, we reduce the learning rate from $3 \cdot 10^{-5}$ to $5 \cdot 10^{-6}$ in order to try to end up with more stable training.

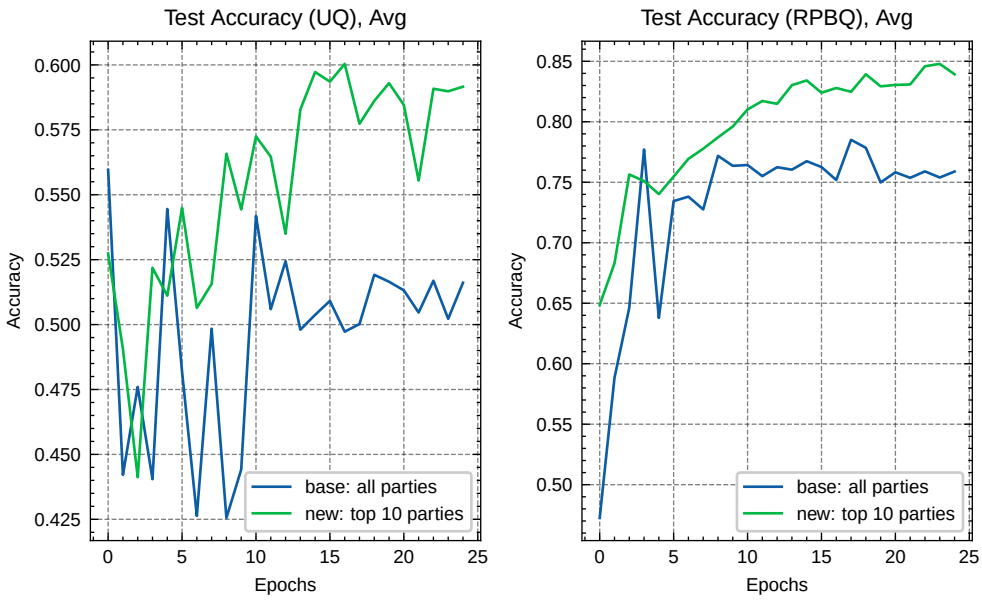


Figure 4.18: Candidate accuracies, avg baseline vs. avg top 10 parties

Figure 4.19: Candidate class accuracies, avg baseline vs. avg top 10 parties

The resulting run does partially achieve this goal for the unseen questions, and is for the most part similar to the average baseline. In figure 4.20 we plot the exception for the unseen questions in comparison with the baseline. It shows that the model with the smaller learning rate achieves substantially higher accuracy, of about 61.5%, on the unseen questions. The remaining results are very similar

though and thus not included.

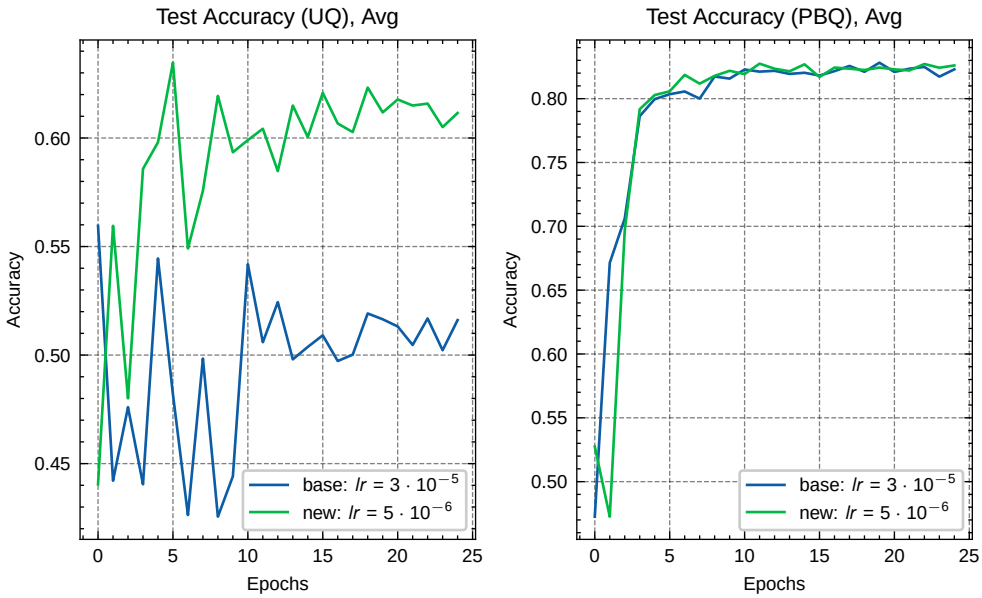


Figure 4.20: Candidate accuracy, learning rate comparison

In the Longformer paper, Beltagy et al. mention that for a question answering task they apply global attention to the entire question in combination with fixed global attention for the context. Inspired by this approach, we try the same. Since, our previous models already use fixed global attention locations throughout the context questions, we just need to add global attention to the tokens of the question we want to predict. With this new addition and everything else the same as for the baseline, we train one model each for the easy, medium and hard setting.

Figures 4.21, 4.22, and 4.23 illustrate the results. Comparing the three trained models with each other, the only exceptional relation we notice is that the difference in accuracy between the max and the rest is very large for the unseen questions. With some results being very similar to the baseline, in particular the test PBQ accuracies are almost equivalent, there are a few stand-out differences. Testing on the RPBQ, the average model achieves accuracies of up to 6% higher. If we compare the max setting for this and the baseline model, the model with the new global attention performs better across the board. Specifically, for the test on the RPBQ the new model reaches an accuracy of 85%, which is equivalent to the accuracy of the test PBQ, compared to the accuracy of the baseline which achieves between 69 and 74%. Previously, there was always a decently large gap in test accuracy when comparing the PBQ and RPBQ.

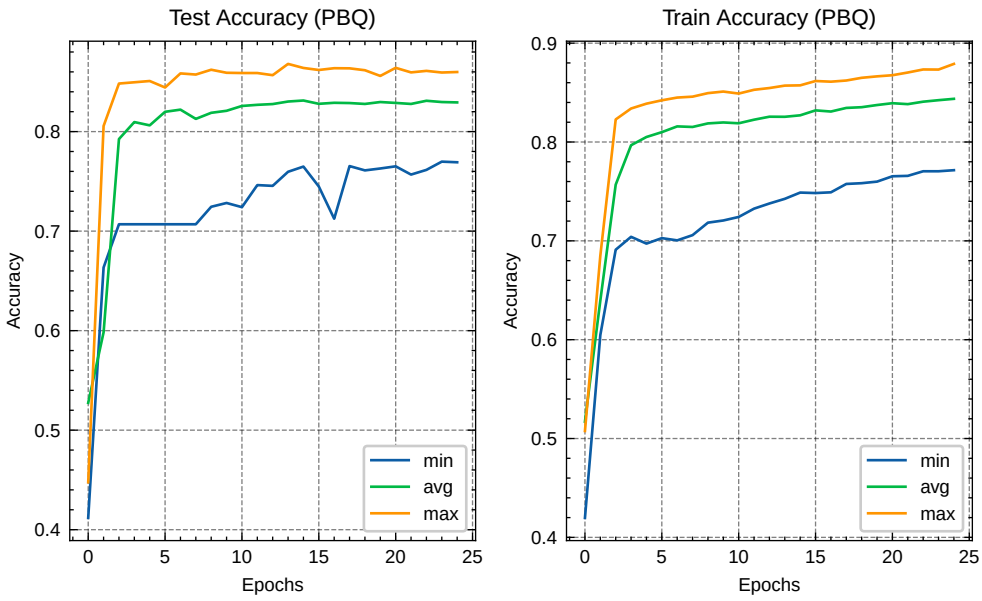


Figure 4.21: Candidate accuracies, new global attention

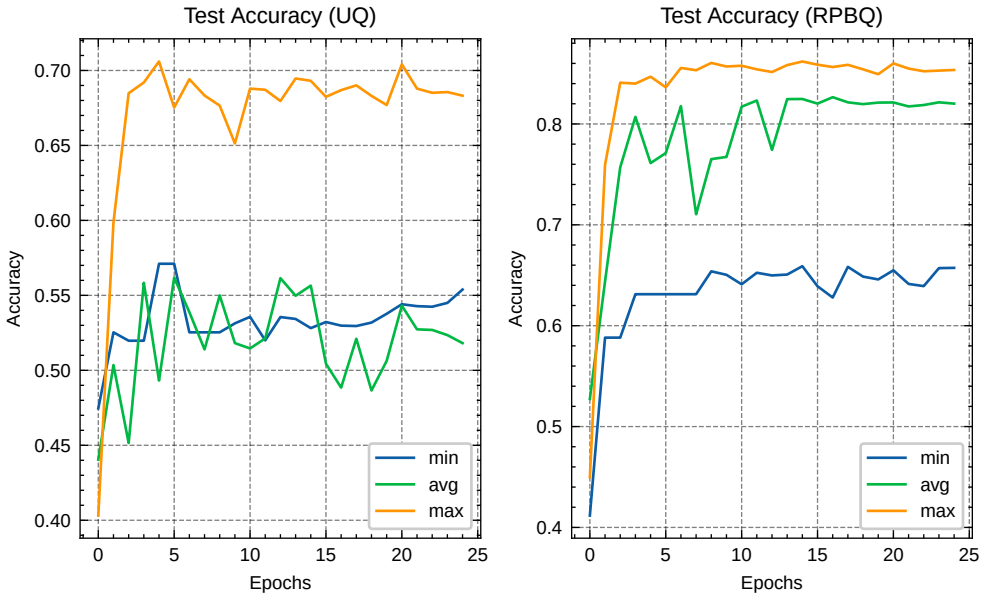


Figure 4.22: Candidate accuracies, new global attention

Additionally, the new maximum correlation based model also reaches accuracies close 70% for the test on unseen questions. This is an improvement of about 3% on average when comparing it to the max baseline.

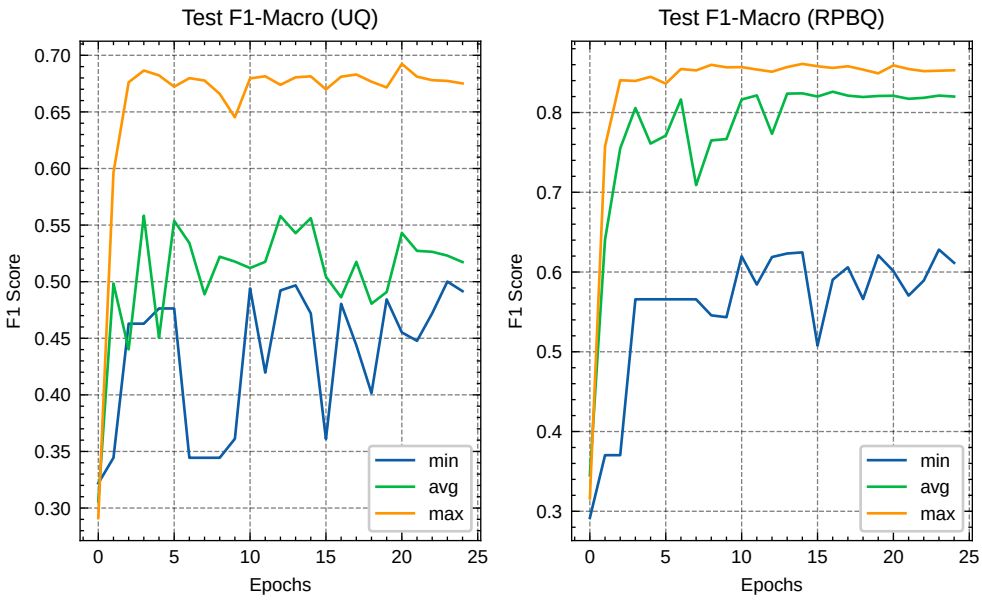


Figure 4.23: Candidate f1-macro, new global attention

In conclusion, the additional global attention on the question to be answered helps with the generalization of our model. We hypothesize that with this added global attention, it is easier for the model to find important correlations between the question to be predicted and the context, even across long distances. This presumably is duet to the attention being direct and not distributed across multiple layers and via few single global locations.

For all of the above answer prediction models, we applied the binary answer reduction to all answers. In order to find out if we lose a lot of information by doing so, we apply the binary answer reduction only to the answers being predicted, while the quarter step reduction is applied to the context answers. We would expect that by keeping more diverse inputs the amount of information contained is higher, and therefore the model can learn more complex inferences. This slightly adjusted training method is run for the easy, medium and hard setting.

The results, illustrated in figures 4.24 and 4.25, show four properties. When comparing the new minimum correlation model with the baseline minimum model, we find the new one to converge slightly faster for the test PBQ and perform significantly better for the RPBQ, namely by at least 3%. For the max setting, we find that the two models perform similarly except for the RPBQ where the new model is far more consistent while also outperforming the baseline. However, the biggest difference by far is between the two average models.

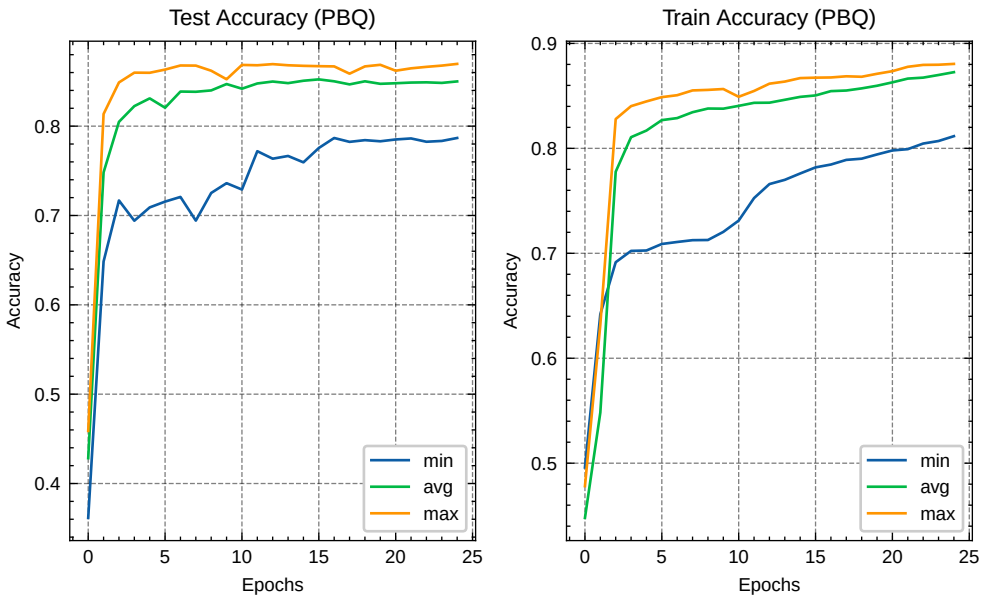


Figure 4.24: Candidate accuracies, new global attention

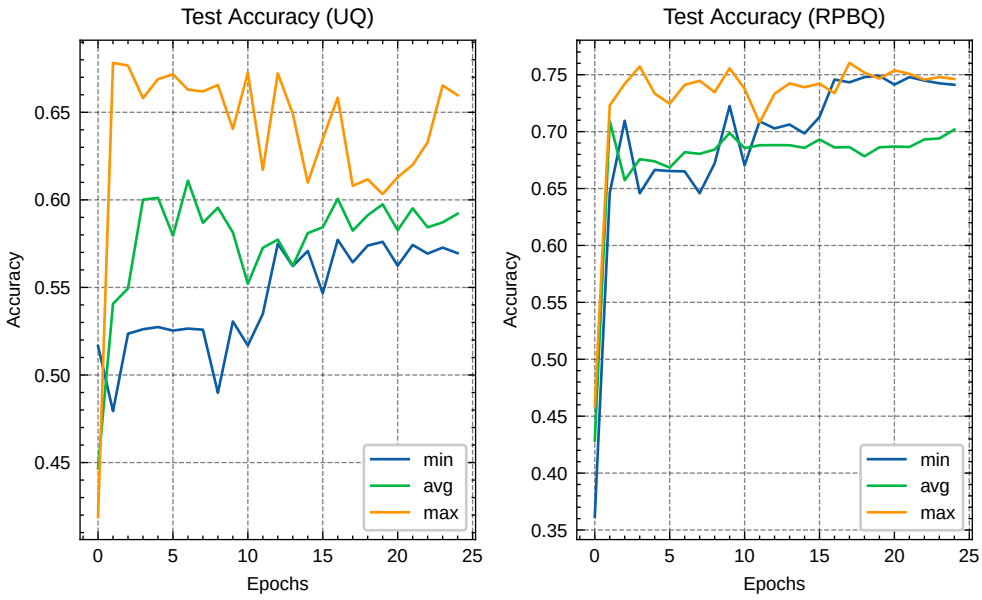


Figure 4.25: Candidate accuracies, new global attention

On the unseen questions, the new model reaches an accuracy of 60.5% which is about 9% higher than the baseline. It further outperforms the baseline on the PBQ by 2-3% but is worse by the same amount on the RPBQ. In conclusion, we find that enough information is lost, due to our original simplification choice, for

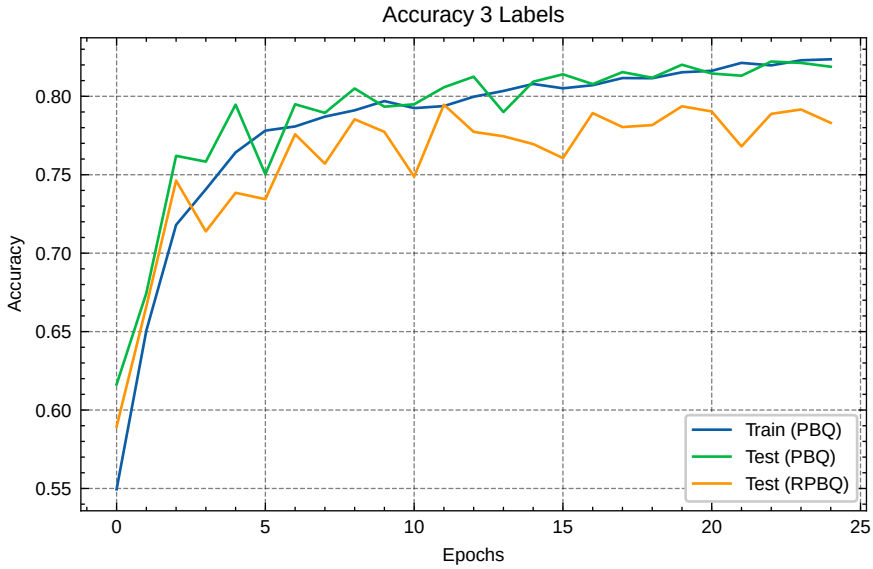


Figure 4.26: Candidate accuracies, new global attention

the change to bring visible differences, primarily on the unseen questions for the average model.

Until this point, we have restricted our model to only predict two classes. We change this now and train an average model where we apply the ternary answer reduction to all answers. Figures 4.26 and 4.27 show that the model actually achieves good accuracies for the PBQ and RPBQ. The test on unseen questions is not included because there the accuracy is only around 53% while the class accuracy for class 50 is 0. This bad performance on the unseen questions could very well originate from the choice of question subsets. E.g., if the questions that have 50 as an answer option are not well distributed across the three sets, it would explain the bad generalization and could also imply the surprisingly high accuracy for class 50, on the PBQ in figure 4.27. No matter what the reason may be, this certainly encourages that it is worth exploring further in order to comprehend where the boundaries of the possible lie regarding this data set.

In appendix A we included a triple of runs where we use a learning rate of $5 \cdot 10^{-6}$ and no weights for the cross-entropy loss. The results are not very special, some specific parts outperform the baseline slightly, and the runs seem to have a slightly less large jumps during training. This could entail more stable results or training. In the future it may therefore be worth it to worry more about the hyperparameters, such that the results become more consistent, and possibly better.

Figure 4.27: Candidate accuracies, new global attention

4.3 Discussion

Something that we have not really mentioned yet, is that there were quite a few training runs, for answer prediction, that did not converge at all. We experimented with a few elements that seemingly helped reduce the frequency of such cases. Currently, we have multiple speculations as to why it may happen. First, since we use the [CLS] token for result aggregation, which is used for a different objective during pretraining, its purpose changes completely during finetuning. Therefore, we could observe that during the first epoch the model eventually only predicts one label, and sometimes it never learns to predict anything else than just that one label. It sometimes happened to switch entirely from predicting only the one label to only the other label. Most likely, the model gets stuck in a local optimum during training.

The reason we do not include any findings for answer prediction on the voter set is because it was suffering from this exact issue. This in combination with it being too much data for our model to handle in a feasible amount of time resulted in the focus on the more reliable candidate set.

Throughout the thesis, we hinted at the importance of the question subset choice. The results, especially before we started using the introduced correlation measure to choose subsets, were varying greatly. Something that we hypothesized about is that there can exist small sets of questions that only genuinely

correlate amongst each other but not with the rest of the questions. Think of a political topic that has supporters and opponents across the entire political spectrum. An example could be regarding vaccines and Covid-19 restrictions, considering that the affiliation of supporters and opponents seems to not really follow any easily understandable distribution. Therefore, if the two questions concerned with these topics are in the questionnaire and happen to be part of the same subset, it will be extremely difficult to learn to predict answers for this topic. It would mean that if in a questionnaire many such small groupings of questions exist, one has to know about and separate them, otherwise it could lead to surprisingly large differences of model performance when just swapping around a few questions between the three sets.

As continuously mentioned in the above results, we found many small changes which increase how well the model predicts answers compared to the baseline. Because it would take a lot of computing power and time to find the best combination of the presented ideas, we leave it for future work with the data set.

In the related work chapter, we included the official recommendation method that Smartvote uses. We would advocate that the one standout property their method has is the ease of computation and traceability of the results. When comparing this to let's say the party classification model we introduced, it is going to be rather hard to convince voters that the reasoning used by our model is not flawed, despite the fact that the model could possibly deliver better results. It would also be hard to show that our model is trustworthy and accurate. The appalling interpretability issues are an active area of research. With breakthroughs in the field, it could make deep learning ideas more attractive to apply in the sensitive field of politics.

It is certainly possible to utilize our findings also when designing future questionnaires of this or similar kind. One could for example hand out a more fleshed out version of the questionnaire to candidates. It would allow us to train models on specific subsets of those question with the goal of ensuring that the most representative subset is chosen based on the achieved prediction capabilities. Candidates fill out a questionnaire most responsibly and are likely willing to answer more questions compared to voters that wanting some quick advice. Hence, using model training and evaluation as a final selection method for questionnaire design could definitely help. Because it would enable the designers to find the set of questions containing the most information, which may be correlated in a way not easily perceivable by a human.

Overall, many results show surprisingly high capabilities when looking at the models we explored. Therefore, we can think of a few new ideas that could reshape the future of politics and voting. In Switzerland, given the semi-direct democracy, people are very privileged to have a direct say in some regards. Due to that, the interest for change may not be high here. When looking at the United States on the other hand, where the presidential election takes a super long time to be entirely counted, people in reality do not have a lot of options to influence decisions made and must trust the representatives they elect.
A very futuristic idea could be that at the beginning of the year everyone can, if they want to, fill out a detailed questionnaire which evaluates the political opinions of all the citizens. Based on the distribution of political opinions that are collected in this way, it would be possible to predict what the citizen's opinions with respect to all the issues are. Something like this could also assist in Switzerland, if you are given the choice every year, if you want to regularly participate in the votes or if you want to fill out the questionnaire. Naturally, such an idea could be extended to the option that everyone can fill out the questionnaire, then see what their automatic votes would be for all the respective topics and if they disagree they could still change it to what they wish. We believe that a system like this could increase the voter turnout by a lot, simply by using the fact that no frequent recurring action is necessary to participate in every vote. Such a questionnaire would also result in very interesting insights about the political opinion distribution of an entire country, thus resulting in many ways one could incorporate such knowledge in decision-making.

Chapter 5

Conclusion

Our experiments show that a lot of unused potential exists in the field of voting advice applications that could eventually lead to new possibilities regarding democracy. As part of this thesis, we were able to successfully train a model that classifies a set of answers to a questionnaire into up to 10 different parties with very high accuracy.

Furthermore, we demonstrate that it is possible to train models that are able to predict answers to questions which were left unanswered, similar to data imputation, with very high accuracy. Moreover, it is even possible to transfer the learned representations to questions that were never seen during training and possibly not even known issues at that time of training. Meaning that it enables the model to predict answers to questions simply by using the model's language understanding of the context questions and answers. Of course, our model is by no means perfect, but we have to take into consideration that this task is not at all what the questionnaire was designed for in the first place. By this, we mean that because we remove a set of questions to train and predict on, the information contained in the context input to our model is lowered. Hence, it is impossible to make a final judgement on the model's actual capabilities if it were given inputs designed for the task it should solve. Our results certainly point to very capable future opportunities in the field.

Based on our research, we find that the transformer based language models can learn representations of political profiles when given an information rich data set. Additionally, our results imply that the longer context capabilities of Longformer help a lot in capturing a meaningful representation of the entire input sequence e.g., when compared to a model based on BERT.

Trained models, like the one we propose in this thesis, show promising opportunities for how citizens could have a more direct impact on a nation's decisions by capturing an entire population's opinions. Models predicting future opinions could also increase voter turnout, by employing for example the idea describe

in the discussion section, where every citizen is presented with the opportunity to fill out a comprehensive questionnaire every year. A choice of options as we describe could lead to the population's opinion distribution being better represented for votes if it reduces the effort and hence increases the voter turnout.

We conclude that there exist many possibilities for the future of democracy, VAAs, voter participation, and more detailed analysis of votes as well as elections that start right here, with the findings of our trained Longformer models using the Smartvote data.

Chapter 6

Future Work

There are many things to be explored in future work. Starting with designing better questionnaires or a different form of survey to provide better information wealth and completeness. Analyzing how models, like the ones we trained for answer prediction, could most efficiently find or decide on the data elements providing the most representative information.

Another important area to explore is which other tasks exist that could be successfully solved with such models. To this end, it is important to know what kind of voting advice people around the world actually need. In Switzerland for example, the creation of candidate lists for the national council elections are always a difficult task. Usually it either that takes a lot of time or, voters are potentially sloppy when deciding on their choices. This is also the main objective that Smartvote tries to aid with, even though we believe that the method used, is most likely by far not as good as it could be.

Provided that the data we have access to is the questionnaire for the Swiss national council elections from 2019 and the fact that parliament website[1] makes all the votes of national council members publicly available, we have the following ideas: We could train a model in the same way with a subset of the votes from parliament as training and the rest as test data. This would allow for us to use the entirety of the Smartvote data as context and thus investigate the actual limitations, without having to remove a subset as we do in this thesis. Another possibility is to train a model on the smartvote data only and try to measure how true the candidates stay to their claimed opinions after being elected.

We know that Smartvote's official recommendation method relies on the euclidean distance in a multidimensional space, and we have access to all the candidate's answers to which a new voter input is compared. This implies, that we could model the task in a Vonroi diagram way, allowing us to know which

[1] https://www.parlament.ch

combinations of answers end up in which polygon. Given such a modelling, we could formulate a kind of optimization problem, that has the goal of maximizing the number of answer combinations being assigned to its polygon. If we further assume that the candidates opinions or answers do not significantly change over the course of four years, such an optimization problem allows for a party to maliciously plan the answers combinations its candidates have to enter in order to maximize the number of votes the party receives. If an attack like this is possible, that leads to the transparent and traceable way of computing recommendations being very counterproductive.

Lastly, the stability of training is certainly an area that needs to be improved and looked into. For example, in which ways it is related to data set restriction or just to training decisions that need to be tuned to work more consistently.

Bibliography

[1] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," *arXiv preprint arXiv:2004.05150*, 2020.

[2] "StemWijzer," Feb 2022, [Online; accessed 26. Feb. 2022]. [Online]. Available: https://stemwijzer.nl

[3] "Wahl-O-Mat," Feb 2022, [Online; accessed 26. Feb. 2022]. [Online]. Available: https://www.wahl-o-mat.de/

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[6] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[7] J. Zhang, Y. Zhao, M. Saleh, and P. Liu, "Pegasus: Pre-training with extracted gap-sentences for abstractive summarization," in *International Conference on Machine Learning*. PMLR, 2020, pp. 11 328–11 339.

[8] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-demos.6

[9] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019,

pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Appendix A

Additional Model Figures

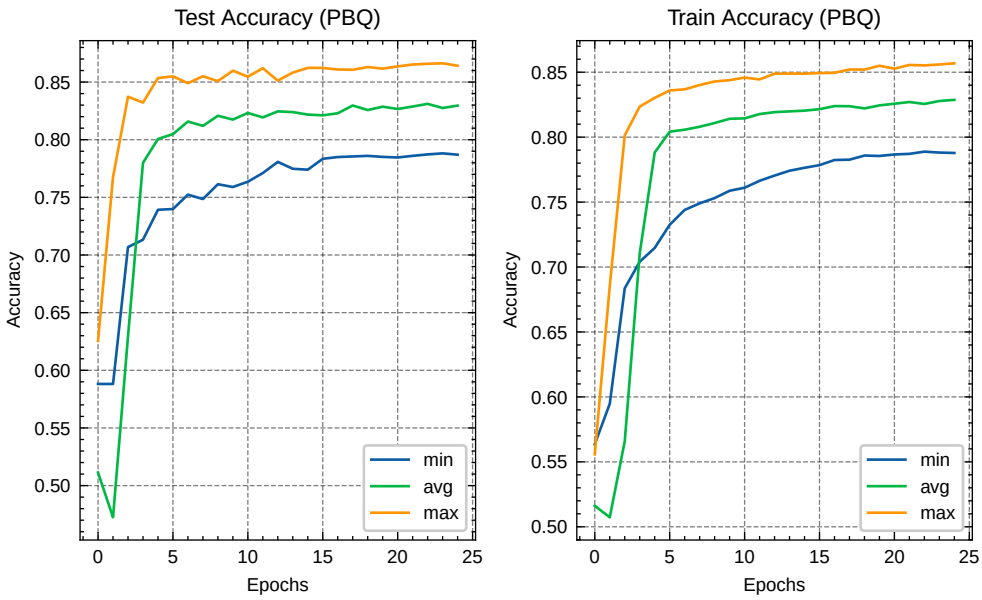


Figure A.1: Candidate accuracies, lr=5 10 ^{6}, no weights for cross-entropy loss

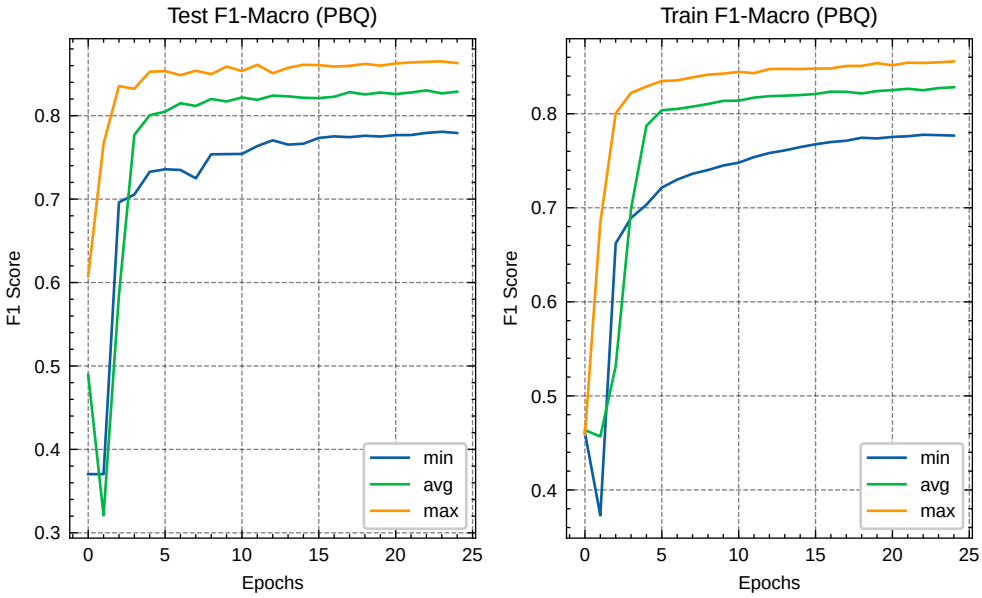


Figure A.2: Candidate f1-macro, lr=5 10 ^{6}, no weights for cross-entropy loss

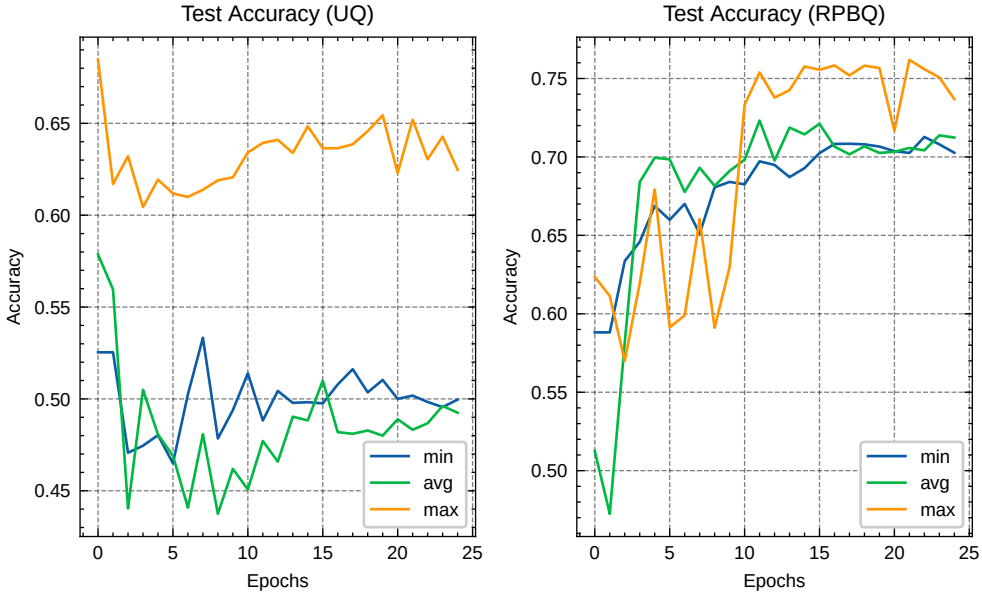


Figure A.3: Candidate accuracies, lr=5 10 ^{6}, no weights for cross-entropy loss

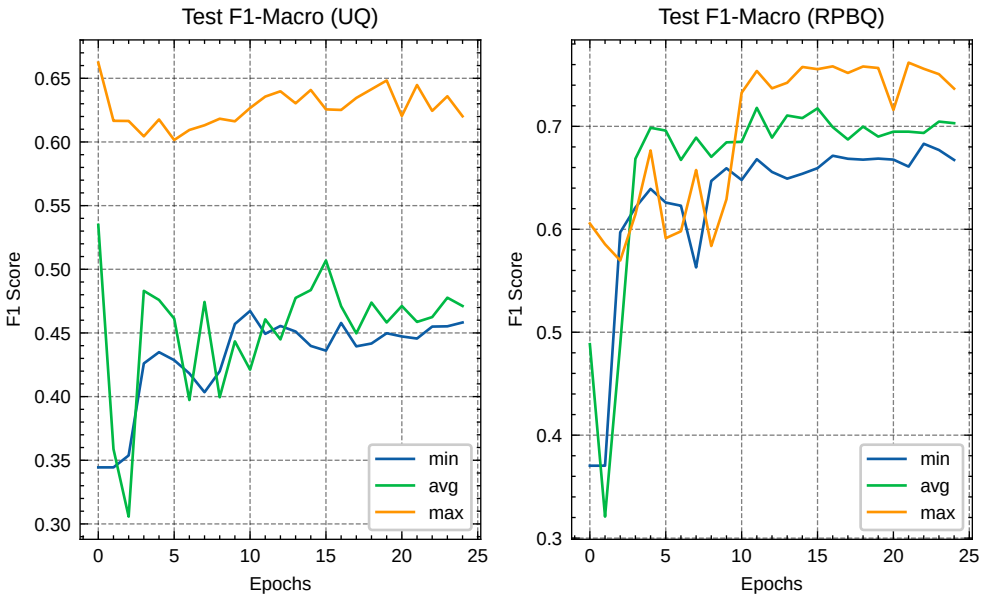


Figure A.4: Candidate f1-macro, lr=5 10 ^{6}, no weights for cross-entropy loss

Appendix B

Data

B.1 Questions

The following table displays all the questions of the data set, including the question type and id. Note that these are the original questions, including any errors they contain.

ID	Type	Question
3387	Slider-7	What is your position the following statement: "Someone who is not guilty, has nothing to fear from state security measures."
3388	Slider-7	What is your position the following statement: "Punishing criminals is more important than reintegrating them into society."
3389	Slider-7	What is your position the following statement: "It is best for a child, when one parent stays home full-time for childcare."
3391	Standard-4	Should the federal government provide more support for the integration of foreigners?
3392	Standard-4	Should cannabis use be legalized?
3398	Standard-4	Should Switzerland terminate the Schengen Agreement with the EU, in order to reintroduce more security checks directly on the border?
3399	Slider-7	What is your position the following statement: "Wealthy individuals should contribute more to the funding of the state."
3412	Standard-4	Do you support an increase in the retirement age (e.g. to 67)?
3413	Standard-4	Should the federal government provide more financial support for the creation of childcare facilities outside the family?
3414	Standard-4	An initiative calls for the introduction of paid paternity leave for four weeks. Do you support this proposal?

ID	Type	Question
3415	Standard-4	Should the conversion rate of the occupational pension fund be reduced in order to adjust for increases in life expectancy?
3416	Standard-4	Do you support cantonal efforts to reduce social welfare benefits?
3417	Standard-4	Should the federal government provide more support for the construction of non-profit housing?
3418	Standard-4	Should insured persons contribute more to healthcare costs (e.g. by increasing the minimal deductible)?
3419	Standard-4	Would you support the introduction of an opt-out solution of for organ donation?
3420	Standard-4	Should compulsory vaccination of children be introduced based on the Swiss vaccination plan?
3421	Standard-4	An initiative calls for health insurance subsidies to be designed so that no one needs to spend more than ten percent of their disposable income on health insurance premiums. Do you support this proposal?
3422	Standard-4	An initiative wants to give the federal government more powers to introduce measures to reduce healthcare costs (Introduction of a cost barrier). Do you support this proposal?
3423	Standard-4	Should the government increase its efforts to support equal education opportunities (e.g. through vouchers for private tutoring for students from low-income families)?
3424	Standard-4	Are you in favour of schools granting/allowing exemptions from individual subjects or events for religious reasons (e.g. PE/swimming, sex education, etc.)?
3425	Standard-4	Should the federal government expand its financial support for continued education and retraining?
3426	Standard-4	According to the Swiss integrated schooling concept, children with learning difficulties or disabilities should be taught in regular classes. Do you approve of this concept?
3427	Standard-4	Should foreigners who have lived in Switzerland for at least ten years be given the right to vote and be elected at the municipal level?
3428	Standard-4	Is limiting immigration more important to you than maintaining the bilateral treaties with the EU?
3429	Standard-4	Should sans-papiers be able to obtain a regularized residence status more easily?
3430	Standard-4	Are you in favor of further tightening the asylum law?
3431	Standard-4	Should the requirements for naturalization be increased?
3432	Standard-4	Should same-sex couples have the same rights as heterosexual couples in all areas?

ID	Type	Question
3433	Standard-4	Should the rules for reproductive medicine be further relaxed?
3434	Standard-4	Are you in favour of stricter monitoring of pay equity for women and men?
3435	Standard-4	Would you be in favour of a doctor being allowed to administer direct active euthanasia in Switzerland?
3436	Standard-4	In your opinion, is lowering taxes at the federal level a priority for the next four years?
3437	Standard-4	Do you support a further reduction in contributions paid by financially strong cantons to financially weak cantons within the framework of financial equalisation (NFA)?
3438	Standard-4	Should married couples be taxed separately (individual taxation)?
3439	Standard-4	Are you in favour of restricting competition between the cantons with regard to corporate tax rates?
3440	Standard-4	Should private households be free to choose their electricity supplier (complete liberalisation of the electricity market)?
3441	Standard-4	Are you in favour of introducing a general minimum wage of CHF 4'000 for all employees for full-time employment?
3442	Standard-4	Should investment controls be introduced in order to better protect Swiss companies from takeovers by foreign investors?
3443	Standard-4	Are you in favour of a complete liberalisation of business hours for shops?
3444	Standard-4	Should the protection against dismissal for older employees be extended?
3445	Standard-4	Should the federal government provide more support for public services (e.g. public transport, post offices) in rural regions?
3446	Standard-4	Should the expansion of the mobile network according to the 5G standard continue?
3447	Standard-4	Should online brokerage services (e.g. "Airbnb" accommodations, "Uber" taxi services) be regulated more strongly?
3448	Standard-4	An initiative calls for Switzerland to stop using fossil fuels by 2050. Do you support this proposal?
3449	Standard-4	Currently, a CO2 charge is levied on fossil combustibles (e.g. heating oil, natural gas). Should this charge be extended to motor fuels (e.g. petrol, diesel)?
3450	Standard-4	Should the federal government provide more support for renewable energies?
3451	Standard-4	Should high traffic motorways be expanded to six lanes?

ID	Type	Question
3452	Standard-4	Are you in favour of introducing "Road Pricing" for motorised individual transport on busy roads?
3453	Standard-4	Do you support the relaxation of the current measures to protect large predators (lynx, wolves, bears)?
3454	Standard-4	Should the current moratorium on genetically modified plants and animals in Swiss agriculture be extended beyond 2021?
3455	Standard-4	Should direct payments only be granted to farmers that provide an extended ecological performance record (e.g. no synthetic pesticides and limited use of antibiotics)?
3456	Standard-4	Are you in favour of extending landscape protection (e.g. stricter rules for building outside existing building zones)?
3457	Standard-4	Are you in favour of stricter animal welfare regulations for livestock (e.g. permanent access to outdoor areas)?
3458	Standard-4	Should campaign finance for political parties and referendums be openly declared?
3459	Standard-4	Should the introduction of electronic voting in elections and referendums (e-voting) be further pursued?
3460	Standard-4	Are you in favour of lowering the voting age to 16?
3461	Standard-4	Should the Federal Council's proposal to tighten the conditions for admission to the civil service be abandoned?
3462	Standard-4	Should the export of war materials from Switzerland be banned?
3463	Standard-4	Are you in favour of Switzerland acquiring new fighter jets for the armed forces?
3464	Standard-4	Do you support an expansion of the legal possibilities for using DNA analysis in investigations?
3465	Slider-7	What is your position the following statement: "In the long term, everyone benefits from a free market economy in the long term."
3466	Slider-7	What is your position the following statement: "The ongoing digitalization offers significantly more opportunities than risks."
3467	Slider-7	What is your position the following statement: "Stronger environmental protection is necessary, even if its application limits economic growth."
3468	Standard-4	Should Switzerland start membership negotiations with the EU?
3469	Standard-4	Should Switzerland strive for a free trade agreement with the USA?

ID	Type	Question
3470	Standard-4	An initiative calls for liability rules for Swiss companies with regard to compliance with human rights and environmental standards abroad to be tightened. Do you support this proposal?
3471	Standard-4	Are you in favour of Switzerland's candidacy for a seat on the UN Security Council?
3472	Budget-5	Should the federal government spend more or less in the area of "Development assistance"?
3473	Budget-5	Should the federal government spend more or less in the area of "National defence"?
3474	Budget-5	Should the federal government spend more or less in the area of "Public security"?
3475	Budget-5	Should the federal government spend more or less in the area of "Education and research"?
3476	Budget-5	Should the federal government spend more or less in the area of "Social services"?
3477	Budget-5	Should the federal government spend more or less in the area of "Road traffic (motorised individual transport)"?
3478	Budget-5	Should the federal government spend more or less in the area of "Public transport"?
3479	Budget-5	Should the federal government spend more or less in the area of "Agriculture"?