Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Distributed Computing

Text-based Model on Voting Advice Application

Semester Project

Mian Zhong

mzhong@ethz.ch

Distributed Computing Group
Computer Engineering and Networks Laboratory
ETH Zürich

Supervisors:
Zhao Meng
Prof. Dr. Roger Wattenhofer

March 28, 2023

Abstract

Voting Advice Applications (VAA), a questionnaire to match election candidates to the voters, gain more impact on the voting participation of citizens. To design a VAA model that effectively gives voters reasonable suggestions, modern natural language processing models surprisingly have been hardly applied. As a first step, we present two text classification tasks to predict party affiliation and policy agreement on a novel dataset from Swiss VAA *smartvote*. We apply a transformer-based model BERT to examine how the textual features may help the VAA model, and show that this text-based model is potential for the proposed tasks. Furthermore, we provide model explainability for this high-stake application by an attribution method, and the results reflect the Swiss political spectrum of parties.

Contents

Chapter 1

Introduction

Voting is a key political participation of citizens in democratic societies. Especially in multi-party countries, voting decisions have to be made frequently for various levels from municipal to national [1]. Hence, the increasing need to be better informed and more efficient in casting a vote gives birth to the *Voting Advice Applications (VAA)*. Many countries in Europe have developed their own VAA, such as, *StemWijzer*[1] in Netherlands, *Wahl-O-Mat*[2] in Germany, and *smartvote*[3] in Switzerland. The design of VAA is typically a questionnaire on (current) important political issues or policies. After a voter fills the questionnaire, VAA gives a matching of candidates or parties to the voter based on the candidate-voter similarity from their questionnaire answers, which is known as "issue-voting" [2].

Nowadays, voting decisions become more challenging for citizens because the ever-growing online information would overload the minds of the voters and influence their political participation [3, 4]. Meanwhile, politicians may have updated their persuasion techniques with artificial intelligence technology to better target voters during election campaigns. To face such challenges, new matching algorithms for VAAs, however, have not been proposed and investigated so often. The current matching algorithms are mostly based on some distance metrics calculated on questionnaire answers to indicate candidate-voter similarity. This method may be employed by parties to find an "optimized" answer profile and to be matched more often [1]. The machine learning (ML) methods, such as Random Forest, may alleviate this issue but they often do not offer model explanations on individual decisions. This is less ideal for such high-stake applications. These approaches also have completely ignored the text information of the questionnaire that may render useful insights about the matching decision.

Additionally, the available VAA text data is fastly increasing since the usage of online VAAs has grown in popularity in recent years. A novel direction of VAA could be anticipating an answer for a new policy initiative based on historical

[1] `https://stemwijzer.nl/#intro`
[2] `https://www.wahl-o-mat.de`
[3] `https://smartvote.ch/en/home`

data. For voters, this advice may encourage them not to skip voting since it may save them time to make the decision. For the government, this may provide valuable insights into the upcoming turnouts and the governors may adjust the initiative procedure to be more efficient.

As a consequence, our work investigates the text-based models from natural language processing (NLP) to work with an original Swiss VAA dataset. We aim to examine how the textual feature may help or fail the VAA matching. We use the task of political party classification to compare different models for predicting party affiliation preference: the distance-based, the ML-based, and the text-based models. Furthermore, we propose a novel classification task to approach the direction of using VAA data to predict an agreement level on an unseen policy issue. With respect to the text-based model, both tasks are trained by finetuning an advanced model, BERT model [5]. It is a type of pretrained transformer-based neural network model and has been extended to different languages. Aside from the original English pre-trained model, we use its variants in German and French too for comparison. Crucially, we provide a scope of model explainability using the integrated gradients [6].

Major contributions of this work are listed as follows:

1. Introduce text-based model for VAA matching on a multi-language dataset and compare with different matching algorithms.
2. Design a novel task to predict a voter's agreement level on a policy initiative.
3. Examine the explainability of text-based models on party affiliation and policy agreement predictions.

We will organize the sections as the following. Chapter 2 will survey the relevant works about NLP developments in political science, VAA algorithm advance, and the explainability of neural networks. Next, we will introduce the dataset and formulate the tasks in Chapter 3. In Chapter 4, we present the approaches, including the baseline models and the text-based model. We also present a type of attribution method for investigating the text-based model explainability. Lastly, Chapter 5 and 6 will present the experiment settings and discuss the results. We conclude the work in Chapter 7.

Chapter 2

Relevant Works

Prior studies of VAA in the political science domain primarily critique on the societal or electoral impacts of such applications [3, 4, 7]. Among them, the Swiss VAA *smartvote* has attracted active research attention, such as, to analyze its effects on the turnouts [8], and to examine partisan patterns of voters [9]. These works take as-is VAA models and pay less attention to the rapid advancement in new computational tools. To bridge the gap, our work focuses on the novel model design of VAA and especially the recent advanced network-based machine learning models (Section 2.1). Unlike previous models, we further consider VAA questionnaire text as model input which is inspired by works that have applied NLP tools in the domain of political science (Section 2.2). Lastly, it is critical to have an explainable prediction since the models assist with voting advice for which "why" to vote could be more important than "what" to vote. Thus, we briefly present the network-based model explainability in Section 2.3.

2.1 VAA Model Development

To begin with, Mendez critically studies two VAA models, the *proximity model* and the *directional model*, that conceptualize a policy dimension differently [10]. The directional model [11] assumes that voter emphasize the side more than the fine policy gradations, whereas the proximity model is based on the distance or similarity [12]. We here clarify the two methods with a simplified hypothetical example. Let agreement on a policy X scales discretely from 0 to 100 (from disagree to agree), given two candidates A and B with an answer 49 and 100 respectively and a voter with an answer 55. The directional model would recommend candidate B to the voter since they both disagree on this policy, and the proximity model would suggest candidate A since the distance of the answers between the voter and candidate A is closer. Louwerse and Rosema [13] further evaluate the matching results of different models: (1) proximity model that uses different distance metrics, for example, Euclidean or city block distance; (2) directional model that defines one, two, or eight political dimensions a priori. They find that the best match between a voter and candidates is influenced by the choice of the model implementation, and the comparison with the election results shows whether a party to be appeared in the voter's match also strongly depends on such model selection.

While these models are straightforward and easy to understand for end users, they treat the questionnaire statements equally in calculating the match. This may be biased due to the questionnaire statement selection procedure. As a consequence, applying machine learning models for VAA recently has received scholarly attention and some initial works [14, 2, 15] propose new model designs for VAA. To begin with, Katakis et al. [14] proposes a community-based VAA with the collaborative filtering model, commonly used in recommendation systems, to suggest candidates to a voter based on the voting intentions of similar voters of this voter. This model requires that the VAA has the information of the users on their voting choice either from previous election or an intended one. The suggestion is the most popular party or candidate among the similar voters who are formed by the VAA questionnaire answers. The user study showed a higher satisfaction of using such social VAAs than the traditional ones. Furthermore, Moreno et al. [2] make a hybrid model to both consider the candidate-voter similarity in political issues and the voter community-based data. The learning model is initiated with some candidate-voter distance like in the traditional method. Yet, it weighs each VAA question differently by learning and adjusts the parameters of the distance matrices with the community-based data. De Ita Luna et al. [15] similarly apply decision tree models on the political opinion polls collected from the web to identify and characterize the political concerns and preferences of the voters. They apply the model to a dataset relevant to a Mexican election in 2013 and compare it with the actual turnout.

Lastly, the prior work by Bensland et al. [16] is a preface of this work that investigates methods of dimension reduction such as Principle Component Analysis (PCA) and clustering such as K-means to analyze the *smartvote*data. They also start with simple machine learning classification models to refine *smartvote* matching and investigate the question importance of the given *smartvote* questionnaire. Additionally, our work was inspired by a data-driven exploratory work on analyzing Swiss voting data including municipality votes, parliament votes and *smartvote* in 2011 [1]. Etter and Herzen et al. use dimension reduction techniques such as PCA to examine political positions. Moreover, they propose the possibility of crafting an "optimized" candidate's answer profile to be recommended more often to users given the current *smartvote* model. Such findings call for substantial changes in VAAs models in order not to be gamed by purposeful attacks.

2.2 NLP for Political Science

Text analysis has always been an important part of research in political science. However, manual scrutiny impedes the underlying potential of large-scale text data and requires intensive labor and time cost. Automated text methods more and more become a standard tool for political scientists and methodologists [17]. The dictionary approach or bag-of-words model is some of the early automation applied to political texts. For example, Laver et al. [18] used word-document frequency matrix to develop a scoring scheme for extracting policy positions from political speeches. They applied the technique to different languages and replicated the performance of expert-annotated policy position estimates. To advance the task, Lowe et al. [19] introduced an alternative scaling method based on the logarithm of odds-ratios.

Nonetheless, the previously mentioned works do not computationally leverage the syntax (e.g., the structure of the sentence) or the semantics (e.g., word senses) of the political texts. As a result, the drastic development in Natural Language Processing (NLP) brings new opportunities to political text analysis. Many prior works in the field related to partisanship and ideology have adopted modern NLP techniques. For instance, Peterson and Spirling proposed a new polarization measurement using classifier accuracy on British House of Commons speeches, where a higher accuracy infers a higher political polarization [20]; Iyyer et al. uses recursive neural networks on U.S. Congressional debates to detect political ideology [21]; Sim et al. apply Hidden Markov Model (HMM) to investigate the ideological evolution during campaigns [22]. Inspired by these works, we expect that a work on VAAs questionnaire texts via NLP tools may reveal interesting results and provide new insights.

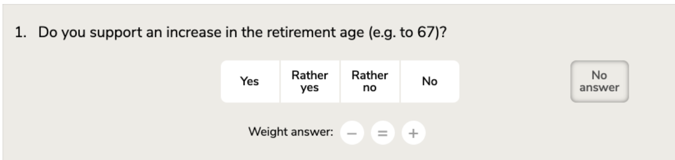
2.3 Explainability of Deep Network Models

While deep networks generally gain more predictive power, a great concern in using them in the domain of public policy is their black-box characteristic and the need to understand what the model has learned. When applying such advanced network-based models, e.g., transformers, on the VAA textual data, we ought to carefully think about and address the model explainability because the results of any VAA model may have an impact on a voter's decision and hence the election turnout. One line of research is to analyze the inside mechanism of these networks. Along this line, prior studies have examined some large pre-trained models. Clark et al. provide a study [23] on the attention-mechanism of BERT model in language modeling, where they find that certain linguistic features like syntax and coreference have corresponded with some attention heads. Li et al. [24] identify linguistic anomaly through studying the surprisal of the model intermediate layers by fitting gaussian models. Another line of previous studies is to propose and evaluate model explainability techniques in NLP, especially for text classification. Among the techniques, gradient-based, perturbation-based, and simplification-based methods are widely studied and Atanasova et al. [25] give a survey to systemically compare these approaches on different datasets.

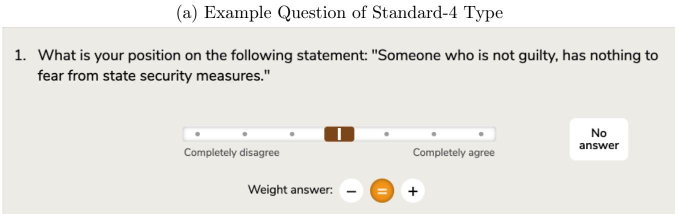
CHAPTER 3

Data and Task

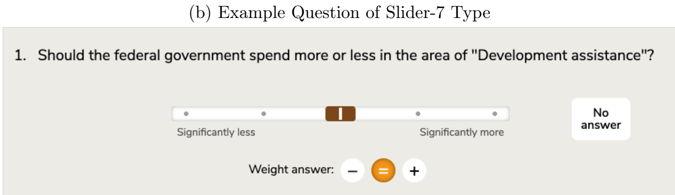
3.1 Dataset: *smartvote* of 2019 national election



(a) Example Question of Standard-4 Type



(b) Example Question of Slider-7 Type



(c) Example Question of Budget-5 Type

Figure 3.1: Example questions of *smartvote* with different types

The *smartvote* dataset consists of 75 questions and the answers from 4,663 political candidates as well as from 427,572 voters for the Swiss national election of 2019. We refer to *participant* as any person who took the questionnaire and *profile* as the answers of a participant to the *smartvote* questionnaire.

The *smartvote* questionnaire is designed into three types of questions: `standard-4`, `slider-7`, and `budget-5` with examples shown in Fig. 3.1. The questions are created by domain experts to cover diverse policy issues and topics that are important to Switzerland. The questions are created in German and French, and an English translation is also provided. For the collected answers, other information about the participants, such as gender and region, is not used in our study. We specifically focus on the questionnaire text and the answers to these questions. Each answer is converted into a scale 0 to 100 which represents an agreement intensity from “strongly disagreement” to “strongly agreement”. For `standard-4`, the response “No”, “Rather No”, “Rather yes”, and “Yes” are encoded as 0, 25, 75, 100 respectively. Similarly for `slider-7`, the response from “Completely diasgree” to “Completely agree” are encoded in 0, 13, 25, 50, 75, 87, 100; and for `budget-5`, the response are encoded in 0, 25, 50, 75, 100 from “Significantly less” to “Significantly more”.

Party candidates were asked to provide all 75 questions and were allowed to leave comments on each question. The candidates are from 69 parties, where the distribution is highly imbalanced. For the voters, smartvote provides two types of questionnaires, “deluxe” and “rapid”. The “deluxe” version contains 75 questions, and the “rapid” version contains 31 questions from the “deluxe” version by *smartvote* experts. The voters cannot leave comments on the questions. Lastly, all participants are able to adjust weight to indicate the importance of the policy question to themselves. They are also able to select “No answer” as their answer.

There are 3,926 candidates and 90,807 voters (with deluxe version) who provide full 75 answers, which will be used in the experiments. Further preprocessing for different tasks will be described in the subsequent sections. Table 3.1 shows the statistics of the dataset.

	75 Answers	<75 Answers	Total
#Candidates	3926	737	4663
percentage	84.19%	15.81%	100%
#Voters(Deluxe)	90,807	179,040	269,847
percentage	21.24%	41.87%	63.11%
#Voters(Rapid)	-	-	157,253
percentage			36.78%
#Voters(Undefined)	-	-	472
percentage			0.11%

Table 3.1: Statistics of *smartvote* dataset of year 2019.

3.2 Task Formulation

We use the *smartvote* data to study two tasks: (1) to identify the party affiliation of a given participant and (2) to predict the agreement of a given participant to an unseen policy question. We model each task as a multi-class classification task and specify the details as follows.

Party affiliation classification. Within a profile, a candidate indicates his or her party affiliation and a voter also indicates a preference for a party to vote for. Hence for this task, given a participant's profile, we use this pre-selected party as the gold standard and classify the profile[1].

Policy agreement classification. We consider an answer to a policy question as ordinal classes to indicate agreement. For example, the simplest way is to convert participants' answers into agree (< 50)/disagree(> 50) as a binary classification task. This may be more fine-grained to be a multi-class classification to show different levels of agreement. We hold out a set of questions to serve as the unseen policy questions. Given a profile of the rest questions, we predict the agreement class of each unseen policy question.

[1] In this dataset, the party preference of a voter is only encoded in integers, and the actual corresponding Swiss party is missing.

Chapter 4

Methodology

In this chapter, we present our approaches to the two classification tasks. Our baselines are distance-based and answer-featured machine learning models (Section 4.1). These models only consider the participants' answers to the questionnaire as input features, and the question texts are completely ignored. To mind the gap, we use an advanced text-based model, namely BERT, by inputting both questions and answers as textual features for the proposed tasks (Section 4.2). In Section 4.3, we further introduce a notion of model explainability that is used to examine the textual features.

Notation Let a VAA questionnaire contain N questions $Q_1, \ldots, Q_N$. For a candidate c, we denote the answers in a vector form $\mathbf{c}^A = (c_1, \ldots, c_N)$, and similarly for a voter v, we denote the answers as $\mathbf{v}^A = (v_1, \ldots, v_N)$ with weights $(w_1, \ldots, w_N)$.

4.1 Baseline Models

Current algorithms of VAAs generally vectorize the answers of candidates and voters and measure a distance between the vectors as their representation of similarity [26]. The measurement is based on some distance metrics, such as City block and Euclidean distance. We refer to such method as *distance-based* model and introduce the model used by *smartvote* in Section 4.1.1. Using the same input, we follow by experimenting with two common machine learning models: Support Vector Machine and Random Forest in Section 4.1.2.

4.1.1 Distance-Based Model

The distance-based model measures the distance of the answers between the voter and the candidates. It recommends the candidates or the party who have the smallest distance to the voter. The distance metric used in *smartvote* is a weighted Euclidean distance shown as the following,

$$d(\mathbf{v}^A, \mathbf{c}^A) = \sqrt{\sum_{i=1}^{N} w_i \cdot (v_i - c_i)^2}$$

Once the distance d is calculated, *smartvote* normalizes it to a matching score between $[0, 100]$, and the candidates who have a higher score are ranked higher in the *smartvote*'s recommendation for the voter. The score s is measured as the following,

$$s(v, c) = 100 \cdot \left(1 - \frac{d(\mathbf{v}^A, \mathbf{c}^A)}{\mathrm{maxDist}(v)}\right)$$

$$\mathrm{maxDist}(v) = \sqrt{\sum_{i}^{N} (100 \cdot w_i)^2}$$

Party affiliation classification. For this task, we implement two baseline models based on the distance matrix. The first one is to construct an "average candidate" model, herein referred to as D_{mean}, by taking the mean of the answers of the same party in the training set for each party. Then, the party affiliation prediction of a given answer profile would be the party of an average candidate that has the highest matching score between the profile and the candidate. The second one is to use the party of the closest candidate in the training set who has the highest matching score to the given answer profile, herein referred as $D_{closest}$

Policy agreement classification. For this task, we have two baseline models similar to the previous task. The first one is based on an average candidate. Given a participant, to predict an agreement answer for an unseen question, we first classify the party affiliation of the profile (on the seen questions) of this participant with D_{mean}[1] and give prediction based on the answer of this average candidate. The second one is simply to use the agreement of the closest candidate as the prediction.

4.1.2 Answer-featured Machine Learning Models

Support Vector Machine

Support vector machine (SVM) is a kernel-based learning method which optimizes the error on margin [27]. A margin is defined as the smallest distance between the decision boundary and the training points. SVM is trained to choose the decision boundary that maximizes the margin. For our multi-class tasks, SVM model uses "one-versus-one" strategy to train a SVM classifier for every two classes.

[1]Note that we only use a subset of the questions, so the result might be different

Random Forest

Random forest is a commonly used ensemble predictor that combines randomized decision trees [28]. The randomness comes at building a tree from a bootstrap sample and splitting a node by randomly selecting a subset of features. The final prediction is an average over the predictions of the individual decision trees.

Party affiliation classification. We simply construct a training matrix by using the answer vector to N questions (e.g., candidate answer $\mathbf{c}^A$) as the feature vector of each participant. This matrix is fit to an SVM or a random forest classifier with the party affiliation indicated by the participant as the labels.

Policy agreement classification. We predict the agreement of an unseen policy issue based on the result of party affiliation classification. Once the party affiliation is predicted, we use the majority answer from this party to be the agreement prediction.

4.2 Text-Based Model: BERT

So far, none of the above-mentioned models considers the question texts in the VAA's questionnaire. This textual information may render critical insights to understand the voter-candidate similarity and improve future questionnaire design. As a consequence, we consider text-based models to incorporate the question texts as input for the two classification tasks. Among the text-based models, the transformer model [29] is acclaimed for its high accuracy in many natural language processing tasks since its emergence.

4.2.1 Background

We give a brief background on the transformer model and the attention mechanism it uses. Based on this, BERT model [5] is developed which uses a bidirectional self-attention mechanism to build a transformer-based encoder that takes contextual information from both directions.

Transformer-based model and Attention. The architecture of the transformer model follows an encoder-decoder design shown in Fig. 4.1. The core component is a stack of multi-head attention layers that are different from previously used convolutional or recurrent layers. Within the layers, an attention component is a dot-product function on mapping a query matrix Q (q by d_k),

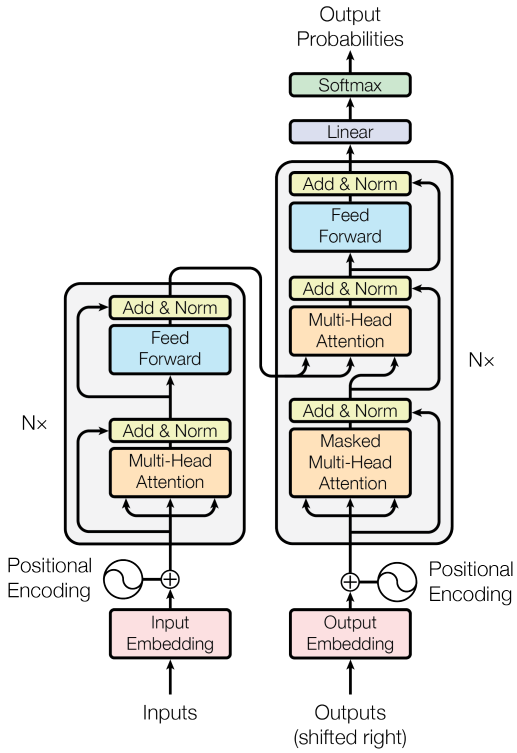


Figure 4.1: Transformer Model Architecture [29]. The core component is the multi-head attention block shown in color orange. This attention mechanism maps a query matrix with a pair of key-value matrices.

and a pair of key-value matrices K (v by d_k), V (v by d_v) as follows,

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V.$$

Additionally, as attention function dismisses the sequence order, a fixed positional embedding is used to inject such information,

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}})$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}}),$$

where d_{model} refers to the model embedding size, i refers to the dimension, and pos refers to the position in the sequence.

BERT Model Based on the transformer architecture, the work of BERT introduces the "pre-training and then fine-tuning" scheme: a large language model is firstly pre-trained on some unsupervised tasks, and the downstream tasks are simply approached by adding an additional layer to fine-tune it. For instance, the original BERT is pre-trained with the masked language modeling task and the next sentence prediction task. An advantage of fine-tuning is little modification in model architechture for different downstream tasks. Following [5], many domain-specific applications have found performance gain by building BERT variants, such as finance, legal, and biomedical areas [30, 31, 32].

4.2.2 Implementation

For our tasks, we apply BERT [5] and its variations GermanBert [33] and French-Bert [34] pre-trianed on German and French language respectively on our data. We achieve our tasks by adding a classification layer on top and finetuning on the pre-trained BERT models end-to-end.

Party affiliation classification. In this task, we classify the party affiliation of the questionnaire participants using the textual concatenation of all question-answer pairs as the input. For instance, given a $\mathbf{c}^A$, we concatenate a string sequence in the format of $[Q_1; c_1; \ldots; Q_N; c_N]$ as the input for the candidate c to the embedding layer of the BERT model. We use the corresponding language of question text when we initialize the model with the different pre-trained weights.

Policy agreement classification. To address this task, we train a classifier to predict a participant's agreement level on a new question given a set of observed question-answer pairs. To do so, we (randomly) divide the questions in two sets: prompt questions Q_{prompt} and unseen questions Q_{new}. The question-answer pairs of the prompt questions serve as the known information about a participant on some policy issues. Then, we construct a sample for this task by appending an unseen question at the end and let the answer of this question be the label. To illustrate, given a $\mathbf{c}^A$, we build the input string as $[Q_1^{prompt}; c_1^{prompt}; \ldots; Q_k^{prompt}; c_k^{prompt}; Q_j^{unseen}], \forall Q_j^{unseen}$, with the label c_j^{unseen}.

4.3 Model Explainability: Integrated Gradients

To analyze models trained for our classification tasks, we adopt an attribution method proposed by [6], *Integrated Gradients*, that provides a channel to examine behaviors of predictions to input features. The method is applied post-training to be model-agnostic. It requires an input baseline to be defined in order to compare with the real data samples. The baseline intuitively is some input that gives "no information" for the classifier, e.g., a black image for image classification or a string of padding tokens for text classification. Using it not only offers where in the input attributes to the prediction but also may render exploratory insights about the input.

Integrated gradients measures the straight-line path from predicted output to baseline. Formally, given a deep network as $F : \mathbb{R}^n \rightarrow [0, 1]$ and an input $\mathbf{x} \in \mathbb{R}^n$, we define a baseline $\mathbf{x}' \in \mathbb{R}^n$ according to the modeling setting and hence obtain the integrated gradient along the i-th dimension for $\mathbf{x}, \mathbf{x}'$ as the following,

$$(\mathbf{x}_i - \mathbf{x}'_i) \times \int_{\alpha=0}^{1} \frac{\partial F(\mathbf{x}' + \alpha(\mathbf{x} - \mathbf{x}'))}{\partial \mathbf{x}_i} d\alpha.$$

We apply the method on the word embedding layer get token attributions and specify the evaluation of the tasks as the following.

Party affiliation classification. We aggregate token attributions per question to be the question attribution. We use this to examine which questions attribute to t to provide a scope of the relationship between party affiliation and political topics within the questions.

Chapter 5

Experiments

In this chapter, we present the experiment settings. The data pre-processing of the text-based models is detailed in Section 5.1 and the trainings of the different models are described in Section 5.2.

5.1 Pre-processing

Party affiliation classification. To study this task, we use the candidates data of the six major parties. The dataset contains 2116 samples and is split into a train, a validation and a test set with a ratio of 70%/15%/15%. Out of 69 parties, we limit ourselves to the six parties in the candidate data that have the most samples, namely, *CVP*, *SVP*, *SP*, *glp*, *FDP*, *Grüne*. These candidates comprise 45% of the candidate data and we brief the party information in Appendix A.

Policy agreement classification. In this task, each example consists of 60 prompt questions with their answers, and a unseen question with its answer as label. We also split with a ratio of 70%/15%/15% for training, validation, and testing data. We convert the labels into 5 classes: 0 (Strongly disagree), 25 (Disagree), 50 (Neutral), 75 (Agree), and 100 (Strongly Agree).

For both tasks, we add special token `[QUESTION]` and `[ANSWER]` to indicate question texts and answer texts. We use the special token `[NA]` to represent `NULL` answer.

5.2 Training

SVM and RF Model We use `scikit-learn`[1] to implement SVM and random forest models. For SVM, we grid search over both linear and radial basis function kernel and different regularization parameters.

[1] https://scikit-learn.org/stable/index.html

Bert Model Our implementation is based on `PyTorch`[2] and `Transformers`[3] libraries. For both tasks, the model is trained for 50 epochs with a learning rate 1×10^{-5} and a batch size of 4. We use Adam optimizer with a weight decay of 0.01. To examine the integrated gradients, we use the **`Captum`**[4] library to implement the attribution method on the word embedding layer of BERT model. We use the a string of `[PAD]` tokens as the baseline.

[2]https://pytorch.org/
[3]https://huggingface.co/transformers/
[4]https://captum.ai/

Chapter 6

Results

In this chapter, we present the experiment results on the two tasks.

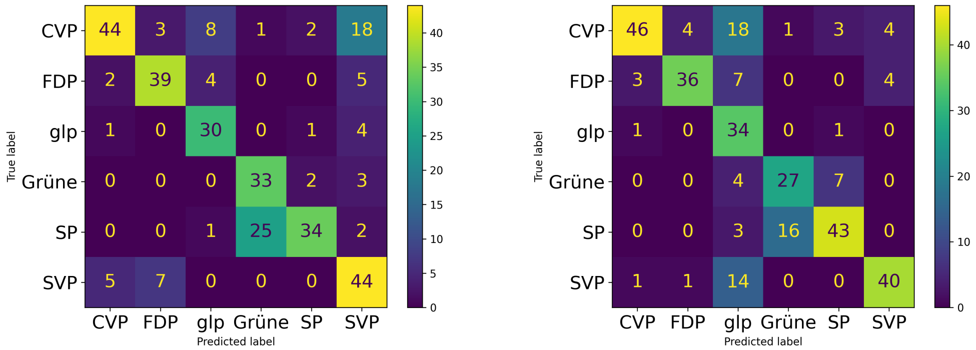
6.1 Results on Party affiliation classification.

In Table 6.1, we compare the accuracy scores on the candidate test data of the different models. It illustrates the total accuracy and the per-class accuracy. Using the feature-based and the text-based models have a higher accuracy than the distance-based models for classifying the party affiliation. Surprisingly, the fine-tuned BERT models overall do not outperform the SVM and random forest models, and the random forest model has the highest accuracy score. Regarding the per-class accuracy, BERT models perform worse than the random forest and it is not a particular party that worsens the performance of BERT models.

	CVP	SVP	SP	glp	FDP	Grüne	Total
D_{mean}	87.4	86.2	89.6	94.0	93.4	90.3	70.4
$D_{closest}$	89.0	**92.5**	90.6	84.9	94.0	91.2	71.1
SVM	**90.6**	88.4	94.7	96.5	**94.7**	94.3	79.6
RF	89.9	88.4	**96.5**	**97.2**	94.3	**95.9**	**81.1**
BERT	85.8	85.8	93.4	95.6	92.1	94.3	73.6
BERT_{de}	86.7	87.4	89.9	95.9	93.3	91.1	72.3
BERT_{fr}	87.7	88.4	88.4	92.5	93.4	84.3	67.2

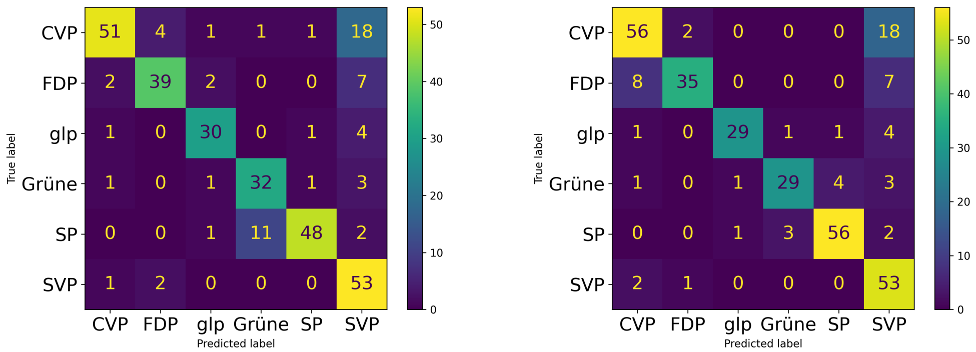
Table 6.1: Classification accuracy (%) on test set. We only conduct our experiments on the 6 major swiss parties.

We plot the confusion matrices of the distance-based models (Fig. 6.1), the feature-based ML models (Fig. 6.2), and the text-based BERT models (Fig. 6.3). All models tend to mistakenly classify (1) the CVP and the SVP party which are both considered as leaning towards right wing, and (2) the Grüne and the



(a) Average Candidate Model (b) Closest Candidate Model

Figure 6.1: Distance-based Models

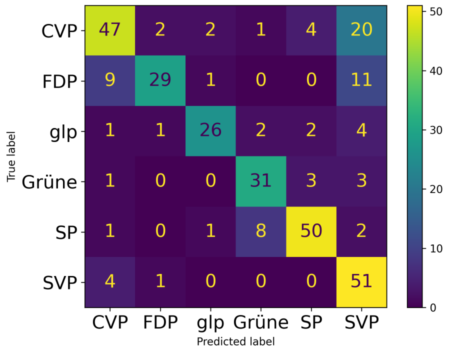


(a) SVM Model (b) Random Forest Model

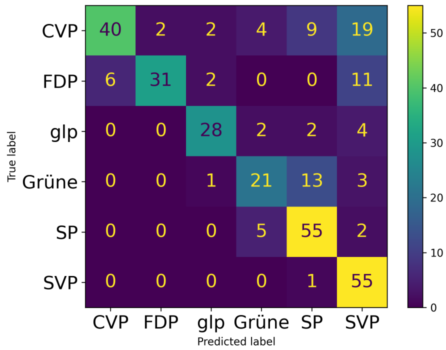
Figure 6.2: Feature based

SP party which embrace more liberal politics. Interestingly, for the latter case, BERT pre-trained on different languages have different behaviors. BERT$_{fr}$ tends to predict the SP party wrongly into the Grüne party, but BERT$_{de}$ behaves the opposite.

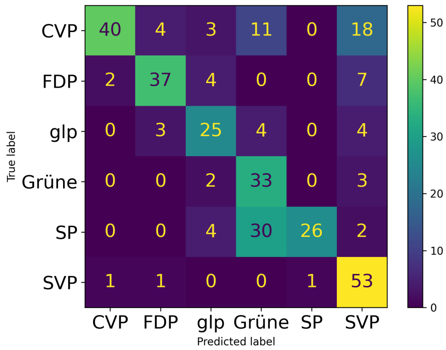
From the results, some samples might be challenging for the learning models to distinguish the party affiliation. This motivates us to look into what questions may be more salient or rather meaningless in characterizing partisan differences. We demonstrate the results of applying integrated gradients on classifying six party affiliations via a heatmap visualization (See Fig. 6.4). The lighter color indicates more positive attribution to this class, while the darker color indicates a more negative attribution. We observe that like-minded parties such as *glp* and *Grüne*, which are known for promoting extended ecological protection and liberal society, share similar attributions to questions. Moreover, it reflects Swiss political spectrum such that neutral parties display a less drastic variation in



(a) BERT Model



(b) BERT$_{de}$ Model



(c) BERT$_{fr}$ Model

Figure 6.3: Text-based Models

question attributions, e.g. *FDP* and *glp*, whereas the conservative party, *SVP*, demonstrates a large variation. The tool is also available to examine the results in a qualitative perspective. As an illustration, we can see that Q51 is among the lightest cell for classifying the party *Grüne* as well as the party *glp*. The original question is about ecology which follows one of the core topics promoted by both parties: "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)". Interestingly, this question has dark color for the party *SP* which has a higher attribution to classify an input not to be *SP*. Another qualitative example is the two most positively attributed questions on predicting the party *CVP*, Q44 ("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)") and Q72 ("Should the federal government spend more or less in the area of 'Social services'?").

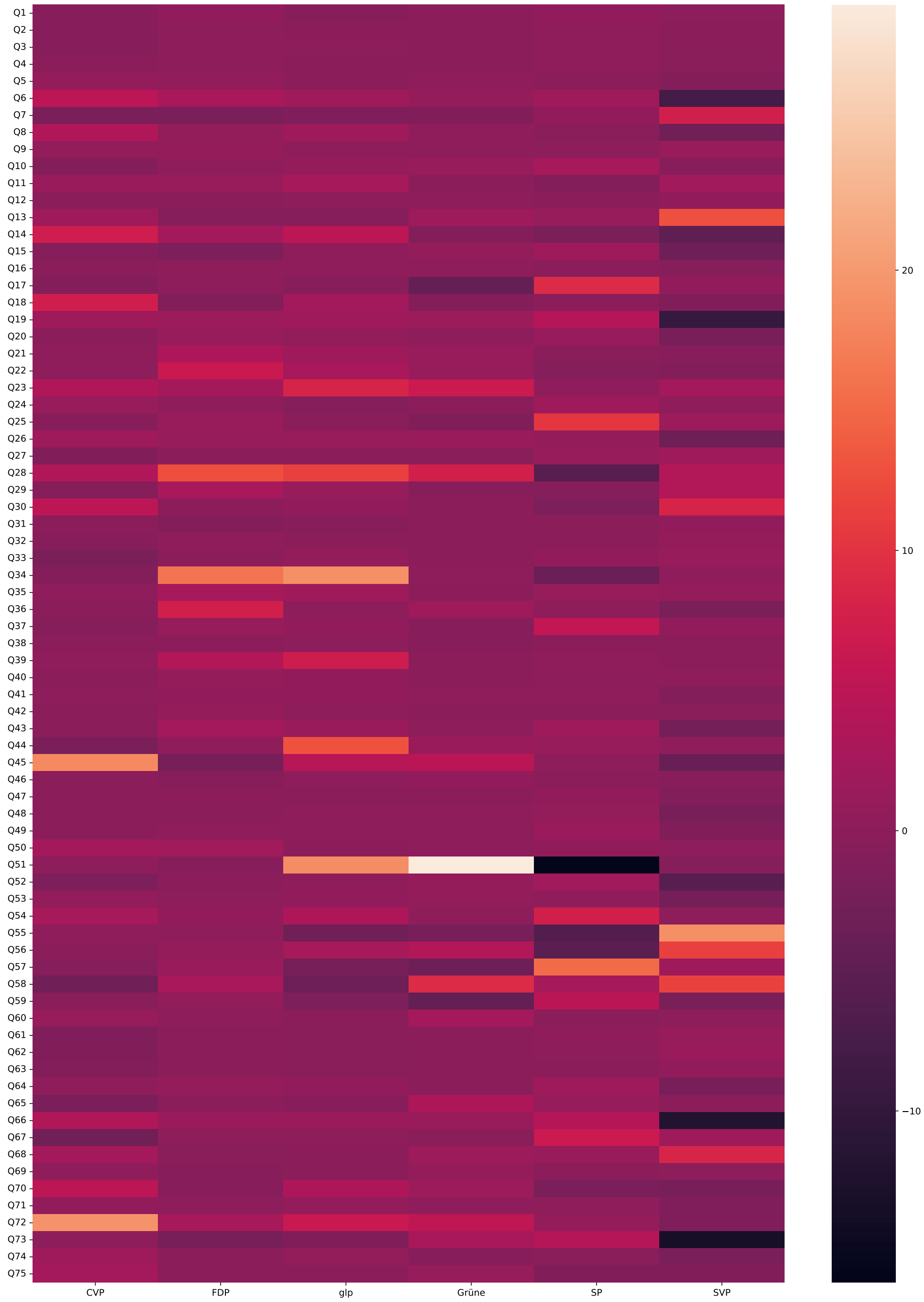


Figure 6.4: Question Attribution on Party Affliation Classification via Integrated Gradients

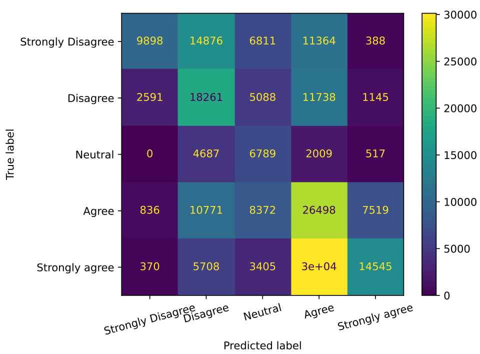
	Accuracy	F1-score	Precision	Recall
D_{mean}	0.37	0.36	0.44	0.39
$D_{closest}$	0.42	0.42	0.43	0.42
RF	0.38	0.37	0.45	0.41
BERT	0.45	0.44	0.44	0.45

Table 6.2: Classification accuracy (%) on voter data set.

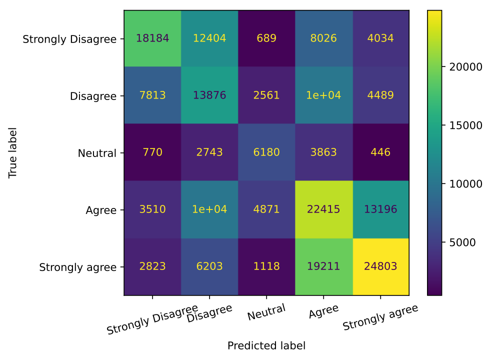
6.2 Results on Policy Agreement Prediction

In Table 6.2, we display the accuracy performance of predicting on the voter test dataset with the models trained on the candidate data. The F1-score, precision, and recall are macro scores. The BERT model has the best performance out of all evaluation metrics. Compared with the average candidate model that mimics some reality situations of passing a policy initiative, i.e., let the preferred party representatives to vote, using the text-based model has shown better accuracy to predict a voter's agreement level. Adding semantic information is potential to help predicting a voter's preference. Among the baseline models, using the closest candidate model is comparatively accurate as the BERT model. However, this approach has the limitation that all the candidates are required to provide agreement answers on the unseen policy initiatives.

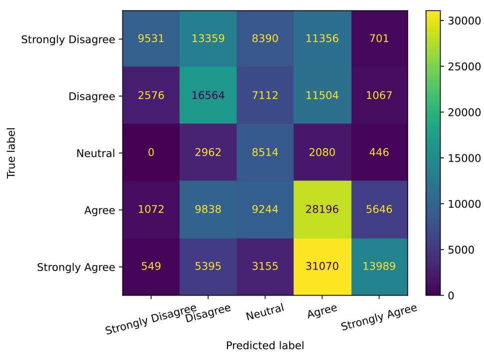
Similarly, we show the confusion matrices of the models respectively in Fig. 6.5. BERT model has better accuracy in predicting the "Strongly Disagree" and "Strongly Agree" than the average candidate and the random forest model. Since we simplify the task as a classification task, the ordinal information of agreement level is less considered in training. Hence, we should be cautious about that this BERT model have more error examples of classifying "Strongly Disagree" and "Disagree" to "Strongly Agree". On the other side, using the average candidate model or the random forest model suffered less in this type of mistake. We should also consider new training objectives in the future work to inject the ordinal class information.



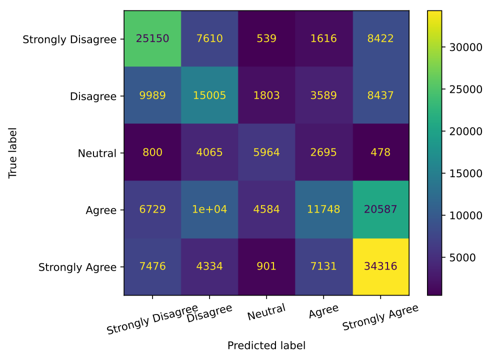
(a) Distance Model (Avg Candidate)



(b) Distance Model (Closest Candidate)



(c) Random Fores Model



(d) BERT Model

Figure 6.5: Confusion Matrices on Policy Agreement Prediction.

Chapter 7

Conclusion

Making voting decision has always been a complicated practice. The massive online information at the present time makes the decision even harder. Despite VAA may assist the procedure, its algorithm is worth investigating by adapting new computational tools. In this work, we are among the first to bring NLP tools into this domain and work with the VAA questionnaire text data. We propose two classification tasks on party affiliation prediction and agreement prediction on unseen political issues. Furthermore, we fine-tune the multi-language BERT models on these tasks and compare with the models that do not use textual features. Importantly, since our tasks are high-stake, we also provide model explainability of the BERT model predictions by using the integrated gradients.

Bibliography

[1] V. Etter, J. Herzen, M. Grossglauser, and P. Thiran, "Mining democracy," in *Proceedings of the second edition of the ACM conference on Online social networks - COSN '14*. Dublin, Ireland: ACM Press, 2014, p. 1–12. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2660460.2660476

[2] G. R. Moreno, J. Padilla, and E. Chueca, "Learning VAA: A new method for matching users to parties in voting advice applications," *Journal of Elections, Public Opinion and Parties*, vol. 0, no. 0, pp. 1–19, 2020. [Online]. Available: https://doi.org/10.1080/17457289.2020.1760282

[3] M. X. D. Carpini, "Voters, candidates, and campaigns in the new information age: An overview and assessment," *Harvard International Journal of Press/Politics*, vol. 1, no. 4, pp. 36–56, 1996. [Online]. Available: https://doi.org/10.1177/1081180X96001004005

[4] H. G. de Zúñiga, A. Veenstra, E. Vraga, and D. Shah, "Digital democracy: Reimagining pathways to political participation," *Journal of Information Technology & Politics*, vol. 7, no. 1, pp. 36–51, 2010. [Online]. Available: https://doi.org/10.1080/19331680903316742

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[6] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 3319–3328. [Online]. Available: https://proceedings.mlr.press/v70/sundararajan17a.html

[7] K. Gemenis and M. Rosema, "Voting advice applications and electoral turnout," *Electoral Studies*, vol. 36, pp. 281–289, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0261379414000742

[8] J. Pianzola, "Swing voting due to smartvote use? Evidence from the 2011 Swiss federal elections," *Swiss Political Science Review*, vol. 20, pp. 651–677, 2014. [Online]. Available: https://doi.org/10.1111/spsr.12120

[9] A. Ladner, "Do VAAs encourage issue voting and promissory representation? Evidence from the Swiss smartvote: VAAs, issue voting, and promissory representation," *Policy Internet*, vol. 8, 10 2016. [Online]. Available: https://doi.org/10.1002/poi3.137

[10] F. Mendez, "What's inside the black box? A critical assessment of how voting recommendations are produced by VAAs," in *XXII Congress of the International Political Science Association*, 2012. [Online]. Available: https://doi.org/10.5167/uzh-112393

[11] G. Rabinowitz and S. E. Macdonald, "A directional theory of issue voting," *The American Political Science Review*, vol. 83, no. 1, pp. 93–121, 1989. [Online]. Available: http://www.jstor.org/stable/1956436

[12] M. Tomz and R. P. V. Houweling, "Candidate positioning and voter choice," *The American Political Science Review*, vol. 102, no. 3, pp. 303–318, 2008. [Online]. Available: http://www.jstor.org/stable/27644522

[13] T. Louwerse and M. Rosema, "The design effects of voting advice applications: Comparing methods of calculating matches," *Acta Politica*, vol. 49, no. 3, p. 286–312, Jul 2014.

[14] I. Katakis, N. Tsapatsoulis, F. Mendez, V. Triga, and C. Djouvas, "Social voting advice applications—definitions, challenges, datasets and evaluation," *IEEE Transactions on Cybernetics*, vol. 44, no. 7, p. 1039–1052, Jul 2014. [Online]. Available: https://ieeexplore.ieee.org/document/6600899

[15] G. De Ita, A. López López, and J. Pérez, "Predicting preferences of voters from opinion polls by machine learning and game theory," *Research in Computing Science*, vol. 77, no. 1, p. 121–131, Dec 2014. [Online]. Available: https://doi.org/10.13053/rcs-77-1-11

[16] S. Bensland, K. Saleh, and P. Ronchetti, "A machine learning analysis of the Swiss political spectrum and candidate recommendation process," 2020.

[17] J. Grimmer and B. M. Stewart, "Text as data: The promise and pitfalls of automatic content analysis methods for political texts," *Political Analysis*, vol. 21, no. 3, p. 267–297, 2013. [Online]. Available: https://doi.org/10.1093/pan/mps028

[18] M. Laver, K. Benoit, and J. Garry, "Extracting policy positions from political texts using words as data," *American Political Science Review*, vol. 97, no. 2, p. 311–331, 2003. [Online]. Available: https://doi.org/10.1017/S0003055403000698

[19] W. Lowe, K. Benoit, S. Mikhaylov, and M. Laver, "Scaling policy preferences from coded political texts," *Legislative Studies Quarterly*, vol. 36, no. 1, pp. 123–155, 2011. [Online]. Available: http://www.jstor.org/stable/23053241

[20] A. Peterson and A. Spirling, “Classification accuracy as a substantive quantity of interest: Measuring polarization in westminster systems,” *Political Analysis*, vol. 26, no. 1, p. 120–128, 2018. [Online]. Available: https://doi.org/10.1017/pan.2017.39

[21] M. Iyyer, P. Enns, J. Boyd-Graber, and P. Resnik, “Political ideology detection using recursive neural networks,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Baltimore, Maryland: Association for Computational Linguistics, Jun. 2014, pp. 1113–1122. [Online]. Available: https://aclanthology.org/P14-1105

[22] Y. Sim, B. D. L. Acree, J. H. Gross, and N. A. Smith, “Measuring ideological proportions in political speeches,” in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, Oct. 2013, pp. 91–101. [Online]. Available: https://aclanthology.org/D13-1010

[23] K. Clark, U. Khandelwal, O. Levy, and C. D. Manning, “What does BERT look at? an analysis of BERT’s attention,” in *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 276–286. [Online]. Available: https://aclanthology.org/W19-4828

[24] B. Li, Z. Zhu, G. Thomas, Y. Xu, and F. Rudzicz, “How is BERT surprised? Layerwise detection of linguistic anomalies,” in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug 2021, p. 4215–4228. [Online]. Available: https://aclanthology.org/2021.acl-long.325

[25] P. Atanasova, J. G. Simonsen, C. Lioma, and I. Augenstein, “A diagnostic study of explainability techniques for text classification,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, 2020, p. 3256–3274. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-main.263

[26] C. van der Linden and Y. Dufresne, “The curse of dimensionality in voting advice applications: reliability and validity in algorithm design,” *Journal of Elections, Public Opinion and Parties*, vol. 27, no. 1, pp. 9–30, 2017. [Online]. Available: https://doi.org/10.1080/17457289.2016.1268144

[27] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[28] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, p. 5–32, 2001. [Online]. Available: http://dx.doi.org/10.1023/A:1010933404324

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[30] D. Araci, "FinBERT: Financial sentiment analysis with pre-trained language models," 2019.

[31] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "LEGAL-BERT: The muppets straight out of law school," Online, pp. 2898–2904, Nov. 2020. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.261

[32] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, Sep 2019. [Online]. Available: http://dx.doi.org/10.1093/bioinformatics/btz682

[33] B. Chan, S. Schweter, and T. Möller, "German's next language model," in *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec 2020, p. 6788–6796. [Online]. Available: https://aclanthology.org/2020.coling-main.598

[34] S. Schweter, "Europeana BERT and ELECTRA models," Nov 2020. [Online]. Available: https://doi.org/10.5281/zenodo.4275044

Appendix A

Overview on Swiss Parties

We present an overview of the six Swiss parties: *CVP*, *SVP*, *SP*, *glp*, *Grüne*, and *FDP* that we use in the task of Party affiliation classification.. In Table A.1, we provide the original full names of these parties and their political orientation. We also include a short list of the core political topics each party is promoting about as a reference. The data is based on information from 2019 and mostly refers to the following link: https://politpro.eu/en/switzerland.

Abbreviation	Full Name in German	Full Name in English	Political Orientation	Core topics
CVP	Christlichdemokratische Volkspartei der Schweiz	Christian Democratic People's Party	Centre-right	- More promotionism; - Against multiculturalism; - Strict crime control; - Strong nationalism.
SVP	Schweizerische Volkspartei	Swiss People's Party	Right-wing	- Strict immigration policy; - Strong nationalism - Against multiculturalism; - Strict crime control.
SP	Sozialdemokratische Partei der Schweiz	Social Democratic Party	Centre-left	- Advocates liberal lifestyles; - Redistribution from rich to poor; - More government intervention in markets and economy; - More investment in public services.
glp	Grünliberale Partei Schweiz	Green Liberal Party	Liberal politics	- Advocates liberal lifestyles; - Open world without borders; - Environmental protection more important than economic growth; - More civil liberties.
Grüne	GRÜNE Schweiz	Green Party	Green party	- Advocates liberal lifestyles; - More government intervention in markets/economy; - Environmental protection more important than economic growth; - Redistribution from rich to poor.
FDP	FDP. Die Liberalen	FDP. The liberals	Liberal politics	- Less government intervention in markets/economy; - Less regulation of market; - Tax cuts; - Against redistribution from rich to poor.

Table A.1: Overview on Swiss Political Parties.