Detection of Negative Campaign in Israeli Municipal Elections

Natalia Vanetik **Sagiv Talker** **Or Machlouf** **Marina Litvak**

Department of Software Engineering

Shamoon College of Engineering

{natalyav,Orma5,Sagivta,marinal}@ac.sce.ac.il

Abstract

Political competitions are complex settings where candidates use campaigns to promote their chances to be elected. One choice focuses on conducting a positive campaign that highlights the candidate's achievements, leadership skills, and future programs. The alternative is to focus on a negative campaign that emphasizes the *negative aspects of the competing person* and is aimed at *offending opponents or the opponent's supporters*. In this proposal, we concentrate on negative campaigns in Israeli elections. This work introduces an empirical case study on automatic detection of negative campaigns, using machine learning and natural language processing approaches, applied to the Hebrew-language data from Israeli municipal elections. Our contribution is multi-fold: (1) We provide TONIC—daTaset fOr Negative polItical Campaign in Hebrew—which consists of annotated posts from Facebook related to Israeli municipal elections; (2) We introduce results of a case study, that explored several research questions. **RQ1**: Which classifier and representation perform best for this task? We employed several traditional classifiers which are known for their excellent performance in IR tasks and two pre-trained models based on BERT architecture; several standard representations were employed with traditional ML models. **RQ2**: Does a negative campaign always contain offensive language? Can a model, trained to detect offensive language, also detect negative campaigns? We are trying to answer this question by reporting results for the transfer learning from a dataset annotated with offensive language to our dataset. **RQ3**: Does a negative campaign necessarily express negative sentiment? Can sentiment analysis help to detect negative campaigns? We experiment with sentiment labels to enrich data representation and report our findings.

Our dataset and pre-trained models will be freely available for researchers.

1 Introduction

Political competitions aim at promoting the candidates' chances to be elected. The main decision in such competitions regards the nature of the campaign – that is, whether a candidate should apply a positive campaign that highlights the candidate's achievements, leadership skills, and future programs, or focus on a negative campaign that emphasizes the negative sides of the competitors (Bernhardt and Ghosh, 2020; Invernizzi, 2019; Martin, 2004; Skaperdas and Grofman, 1995).

Our work introduces a new dataset of Facebook posts, published by candidates during municipal elections in Israel. We annotated this dataset with binary labels, where we distinguish between negative campaigns and other campaign-related content. In addition to the dataset, we report the results of extensive experiments, aimed at answering multiple research questions: Which supervised model and representation are more effective at automatically detecting negative campaigns? Can we effectively detect negative campaigns with a model trained to identify offensive language? Can sentiment analysis boost negative campaign detection?

Our contribution is multi-fold: (1) We introduce a new annotated dataset in Hebrew for negative campaign detection; (2) We report results of multiple classifiers and their combination with various representations on our dataset; (3) We explore possible relations between sentiment analysis and negative campaign and (4) between offensive language and negative campaign.

2 Related Work

The scholarly literature has investigated various aspects related to negative campaigns (Asunka et al., 2019; Chaturvedi, 2005; Martin, 2004; Skaperdas and Grofman, 1995). It points out that this phenomenon exists in many areas such as competition over jobs in the workplace, yet in the politi-

Proceedings of the 29th International Conference on Computational Linguistic, pages 68–74
October 12–17, 2022.

cal arena there are several special characteristics. A major characteristic is a fact that participants in electoral competition often hold positions of power as well as public and private resources to finance their efforts (Bernhardt and Ghosh, 2020; Invernizzi, 2019). In many cases, they also set the rules of the competition as opposed to other areas where the contest organizer sets the rules of the game. Indeed, in recent years, we witness more and more political candidates that do not play by the rules, both formally and informally. A specific feature of this trend is the intensive use of negative campaigns which target the weaknesses and failures of the opponents promising to do the opposite (Invernizzi, 2019; Martin, 2004; Skaperdas and Grofman, 1995).

The implementation of language technologies in the political sciences is recently in high demand. While computational political scientists are looking for NLP tools to assist automatic analysis of campaign-related content and predict outcomes, computational linguistics explores real-world use cases in political domains. The recent Workshop on Natural Language Processing for Political sciences (PoliticalNLP) (Afli et al., 2022) is an example of the rising popularity of this interdisciplinary research. However, despite some works dedicated to the analysis of elections-related materials (Baran et al., 2022; Abdine et al., 2022; Sanders and van den Bosch, 2022), in this workshop or anywhere else, we were unable to find any work on automated negative campaign analysis and detection.

As a majority of text classification tasks last years are efficiently performed by pre-trained language models and transformers, we follow this approach in our study. We apply BERT, its multilingual (mBERT) and Hebrew (AlephBERT) versions. mBERT serves us both as an encoder (feature extractor) and end-to-end classifier.[1] In addition to the introduction of a new dataset, we explore possible relations between sentiment analysis and negative campaigns and between offensive language and negative campaigns.

3 Case Study

3.1 TONIC dataset

The data was collected from Facebook accounts of local politicians from several big Israeli cities running for mayor's offices. There were total of 12 cities and 27 mayor candidates whose number for elections that took place in 2018. Data statistics appear in Table 1. The data is freely available for download from GitHub at `https://github.com/NataliaVanetik1/TONIC`.

Table 3 displays two instances of comments from the TONIC dataset that have been translated into English.

The collected posts were first manually filtered as related or unrelated to political campaigns, and only campaign-related messages were kept. Those texts were annotated as either negative or not by two independent annotators; in case of a disagreement between them, the third annotator decided on a final label. The annotators were instructed to label a post as "negative campaign" only if it contained a negative (but not necessarily offensive) content about the opponent of the post's owner or her supporter. Kappa agreement between the annotators was 0.862, which is considered to be an excellent agreement. The statistics for campaign-related posts for all cities are given in Table 2.

3.2 Method

Our approach to text representation and classification consists of the following steps:

1. Representing texts with one of the following:
    - tf*idf vectors, where every post is treated as a separate document;
    - character n-grams with $n = 1, 2, 3$;
    - pre-trained BERT vectors obtained from a multilingual BERT model (Sanh et al., 2019).
2. Enhancing the above representations with sentiment weights produced by the pre-trained HeBERT model (Chriqui and Yahav, 2021). This model produces weights as a probability distribution for positive, negative, and neutral sentiments.
3. Training and application of supervised ML models (see Section 3.4) on all of the above data representations.

The approach is depicted in Figure 1.

[1] We did not apply AlephBERT as feature extractor because its implementation does not comply with the sentence transformers package and does not allow extraction of sentence vectors.

Table 1: Collected data by city

city	candidates	post num	avg words in post	avg characters in post
Ashdod	4	644	64.2	367.9
Netanya	4	571	49.2	292.0
Jerusalem	3	516	65.5	386.8
Ashkelon	3	683	61.4	358.4
Petah Tikva	4	669	61.7	359.0
Haifa	1	104	51.7	304.4
Rishon LeZion	1	239	87.2	523.7
Dimona	1	95	57.2	338.2
Hod Hasharon	2	366	71.0	416.1
Tel Aviv	1	233	70.8	410.2
Beer Sheva	1	34	139.8	866.4
Herzliya	2	272	92.4	549.4
Total	27	4426	65.6	385

Table 2: TONIC statistics

post num	pos	neg	majority	avg words	avg chars
2632	568	2064	0.784	85.2	500.6

Table 3: Two sample comments from the TONIC dataset

Translated comment	Label
Good week to all residents of Ashdod! Let's talk about Ashdod-Yam Park. Who does the park belong to? Does the park belong to the residents of the city at all or only to the ultra-Orthodox residents (and non-residents)? Ashdod-Yam Park has been an attraction for the ultra-orthodox public from all over the country for years. I really don't have a problem with it, or most residents of Ashdod, but as soon as the park and its facilities are closed on Shabbat, the message to the non-Orthodox residents of Ashdod is simple: you are not welcome in your city. Unless you are...that's right - ultra-Orthodox. This week the municipality of Ashdod decided that as part of the closing of the park's facilities on Shabbat, the only cafe in the park will also be closed on Shabbat. Another conquest of the ultra-Orthodox businessmen with the kind help of Yehiel Lesri. We must return the city to all residents. The city was not intended only for the ultra-Orthodox. With your help I will be the mayor and then Ashdod will serve you all!	yes
What has already become a procedure, the week is closed with the dear residents of the city! Today we visited the 11th and 12th districts and were happy to meet the residents, to hear what they like in the city, what they dislike and what problems they suffer from. On 30.10.18 we will be able to start providing better service to the resident and take care of the needs of every district and every community in the city. Many thanks to the dear activists who accompany me all along the way.	no

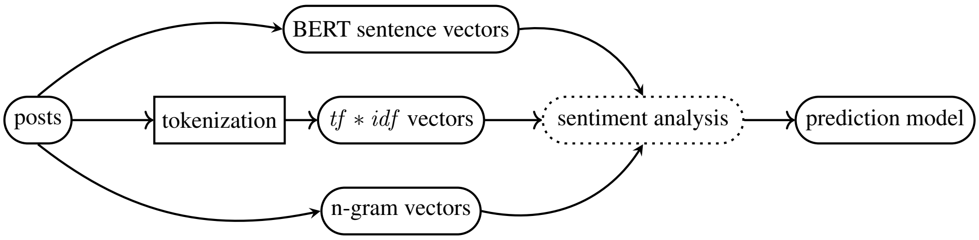

Figure 1: Political posts classification pipeline.

3.3 Data representation

We employed three different representation models for input texts, as follows.

Tf-idf, short for term frequency-inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. The tf-idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word. In our case, we treated every post as a separate document and the whole dataset as a corpus.

N-grams are the sequences of n consecutive words seen in the text, where n is a parameter. In our evaluation, we used the values $n = 1, 2, 3$.

BERT sentence embeddings of length 512 were obtained using the pre-trained multilingual BERT model trained in 104 languages, including Hebrew.

3.4 Models

We applied three traditional ML algorithms—Random Forest (RF) (Pal, 2005), Logistic Regression (LR) (Wright, 1995), and Extreme Gradient Boosting (XGB) (Chen et al., 2015). All three were applied to texts represented by each of three representations, described in Section 3.3.

Also, we employed the BERT transformer (Devlin et al., 2018) trained for sentence classification. We applied two different pre-trained models for our task. The first one is a multilingual model called *bert-base-multilingual-cased* (denoted as mBERT) introduced in (Devlin et al., 2018). The second is the AlephBERT (Seker et al., 2021), a large pre-trained language model for Modern Hebrew, which is trained on a larger vocabulary and a larger dataset than any Hebrew pre-trained language model before. Both of these models were fine-tuned on the train portion of our data.

3.5 Experiments

Our experiments aim at evaluation of and comparison of various models and text representations for the purpose of detecting negative campaigns in political posts. Additionally, we explore two research questions.

In the first one, we want to understand whether general offensive language data in the same language (Hebrew) can be used for transfer learning with the proposed methodology. For answering that, we perform cross-domain experiments with a dataset with Hebrew messages annotated with offensive language.

In the second question, we explore whether the sentiment analysis can boost the negative campaign detection accuracy. For answering that, we compare between performance scores of our models with and without sentiment labels in the data representation.

3.6 Data Setup

For the experiments on TONIC, RF, LR, and XGB were trained on 80% of the data and evaluated on the remaining 20%. Fine-tuned BERT was trained a 75% of the data with the validation set containing 5% of the data, and it was tested on the remaining 20%. Fine-tuning was run for 10 epochs with batch size 16.

For the cross-domain experiments, we used the Hebrew offensive language dataset (Litvak et al., 2021) called OLaH. It is composed of Facebook comments written in Hebrew and annotated by humans. The dataset contains 2,025 annotated comments, out of which 821 are labeled positive (i.e., they do contain offensive content).

3.7 Software Setup

For the purpose of reproducibility, we present below the setup of our experiments. All non-neural models are implemented in sklearn (Pedregosa et al., 2011) python package. Our neural model is implemented with Keras (Chollet et al., 2015) with the TensorFlow backend (Abadi et al., 2015). Experiments were performed on google colab (Bisong, 2019) with standard settings and GPU runtime type. Runtime for every experiment setting (mono- or cross-domain) was less than 10 minutes.

3.8 Evaluation Results

Table 4: Mono-domain evaluation results on the TONIC dataset

Model	P	R	F1	Acc
AlephBERT	0.7318	0.6616	0.6949	0.7040
mBERT	0.7288	0.6792	0.7031	0.7590
RF_{tfidf}	0.8004	0.5490	0.6513	0.8008
$RF_{tfidf+SA}$	0.8507	0.6550	0.7401	0.8425
RF_{ng1}	0.7774	0.5597	0.6508	0.8027
RF_{ng1+SA}	0.8517	0.6877	0.7610	0.8539
RF_{ng2}	0.8157	0.5414	0.6508	0.7989
RF_{ng2+SA}	0.8372	0.6168	0.7103	0.8273
RF_{ng3}	0.771	0.5239	0.6239	0.7913
RF_{ng3+SA}	0.8482	0.6506	0.7364	0.8406
RF_{bert}	0.8485	0.5383	0.6587	0.7989
$RF_{bert+SA}$	0.8508	0.7355	0.7890	0.8691
LR_{tfidf}	0.7731	0.5358	0.6329	0.7951
$LR_{tfidf+SA}$	**0.8530**	0.7399	0.7924	0.8710
LR_{ng1}	0.7341	0.6571	0.6935	0.8159
LR_{ng1+SA}	0.8474	**0.7928**	**0.8192**	**0.8843**
LR_{ng2}	0.6551	0.6491	0.6521	0.7685
LR_{ng2+SA}	0.7455	0.7501	0.7478	0.8273
LR_{ng3}	0.6551	0.6491	0.6521	0.7685
LR_{ng3+SA}	0.7455	0.7501	0.7478	0.8273
LR_{bert}	0.7864	0.6075	0.6855	0.8178
$LR_{bert+SA}$	0.8096	0.7851	0.7972	0.8672
XGB_{tfidf}	0.8195	0.5948	0.6893	0.8178
$XGB_{tfidf+SA}$	0.8151	0.7724	0.7932	0.8672
XGB_{ng1}	0.8007	0.5892	0.6789	0.8140
XGB_{ng1+SA}	0.8303	0.7879	0.8085	0.8767
XGB_{ng2}	0.7168	0.5978	0.6519	0.8027
XGB_{ng2+SA}	0.8121	0.7787	0.7950	0.8672
XGB_{ng3}	0.7241	0.5991	0.6557	0.8046
XGB_{ng3+SA}	0.8072	0.7699	0.7881	0.8634
XGB_{bert}	0.6733	0.5798	0.6231	0.7894
$XGB_{bert+SA}$	0.8208	0.7887	0.8044	0.8729

Evaluation results for the TONIC dataset as both train and test sets are presented in Table 4. We can make the following conclusions from these results.

First, there is a shred of strong evidence that sentiment labels boost classification performance. Second, the best recall, f-measure, and accuracy were produced by LR with unigrams enriched with sentiment labels, and the best precision was obtained by the same LR but with tf-idf and sentiment labels.

Table 5: Cross-domain evaluation results: OLaH→TONIC

Model	P	R	F1	Acc
AlephBERT	0.4988	0.3916	0.4387	0.7818
mBERT	0.5135	**0.6025**	0.5545	0.7799
RF_{tfidf}	0.4536	0.4959	0.4738	0.7723
$RF_{tfidf+SA}$	0.5356	0.5054	0.5201	0.7723
RF_{ng1}	0.4530	0.4779	0.4651	0.7192
RF_{ng1+SA}	0.5599	0.5079	0.5326	0.7761
RF_{ng2}	0.4779	0.4886	0.4832	0.7211
RF_{ng2+SA}	0.5222	0.5072	0.5146	0.7552
RF_{ng3}	0.4727	0.4867	0.4796	0.7230
RF_{ng3+SA}	0.5031	0.5009	0.5020	0.7552
RF_{bert}	0.3910	0.4952	0.4370	0.7761
$RF_{bert+SA}$	0.4628	0.4971	0.4793	0.7742
LR_{tfidf}	0.3918	0.5000	0.4393	0.7837
$LR_{tfidf+SA}$	0.3914	0.4976	0.4382	0.7799
LR_{ng1}	0.4723	0.4850	0.4786	0.7154
LR_{ng1+SA}	0.5222	0.5072	0.5146	0.7552
LR_{ng2}	0.5417	0.5396	0.5406	0.6964
LR_{ng2+SA}	0.5484	0.5382	0.5433	0.7192
LR_{ng3}	0.5417	0.5396	0.5406	0.6964
LR_{ng3+SA}	0.5484	0.5382	0.5433	0.7192
LR_{bert}	0.4623	0.4942	0.4777	0.7647
$LR_{bert+SA}$	0.5592	0.5039	0.5301	0.7799
XGB_{tfidf}	0.5352	0.5027	0.5184	0.7780
$XGB_{tfidf+SA}$	0.7265	0.5076	0.5976	0.7856
XGB_{ng1}	0.4617	0.4914	0.4761	0.7552
XGB_{ng1+SA}	**0.7946**	0.5163	**0.6259**	**0.7894**
XGB_{ng2}	0.5315	0.5174	0.5244	0.7362
XGB_{ng2+SA}	0.6214	0.5262	0.5699	0.7799
XGB_{ng3}	0.5609	0.5388	0.5496	0.7400
XGB_{ng3+SA}	0.5789	0.5162	0.5458	0.7742
XGB_{bert}	0.4680	0.4892	0.4784	0.7419
$XGB_{bert+SA}$	0.5460	0.5113	0.5281	0.7666

Cross-domain experiments in Table 5 show that using an offensive language dataset as a training set decreases classification accuracy for all the models, indicating that the task of detecting negative campaigns is different from the task of offensive language detection. Despite enhancing data with SA obviously improve results, only a few models trained on offensive language data achieved accuracy that is slightly higher than or equal to the majority rule. XGB with unigrams and sentiment labels achieved the best precision, f-measure, and accuracy, while the best recall was obtained by mBERT.

3.9 Error Analysis

We used the top-performing model ($LRng1 + SA$) to analyze the misclassification errors in the mono-domain classification instance (Logistic Regression with unigrams and sentiment labels as a text representation). This model's confusion matrix is as follows: $TP = 72$, $TN = 394$, $FP = 19$, and $FN = 42$, with precision of 0.79 and recall of 0.63 respectively. These results show that the model does a good job of identifying and eliminating negative samples (non-negative campaigns), but it misses positive samples (negative campaign). As a result, TN is the most important accuracy compound, while FN represents the biggest amount of errors.

In a 30 misclassified case sample that we manually examined, 22 cases are from the FN group and only 8 cases are from the FP category. The majority of errors (23), including 19 samples incorrectly identified as negative campaigns when we actually found them to be neutral and 4 samples incorrectly labeled as neutral, were the result of incorrect labeling by our annotators. Due to a variety of factors, the model incorrectly classified four neutral posts as negative campaigns, including one sample that was actually negative but was correctly categorized as neutral because it wasn't addressed to a specific person, and two samples that contained words that were likely to have influenced the classification. One sample was incorrectly categorized for an unidentified reason; the cause is likely due to the negative campaign writing style, which is characterized by frequent mentions of individuals. The model missed three unfavorable marketing materials, most likely as a result of the neutral vocabulary (no offensive content in these samples).

4 Future Work and Conclusions

In this paper, we introduce a new dataset that can help researchers to study negative campaigns. The dataset contains only Hebrew-written content posted by Israeli politicians on Facebook. We report the results of extensive experiments which include multiple classifiers and representations and answer two research questions: whether transfer learning from offensive language to negative campaign can be efficiently applied and whether sentiment analysis can boost negative campaign detection. We can conclude that traditional models with unigrams and sentiment labels as text representations performed best in both scenarios. This is probably due to a small training set which is not sufficient for efficient fine-tuning of pre-trained transformers with a large number of parameters, but big enough to train a relatively simple classification function with fewer parameters. Also, unigrams seem to be most efficient in representing Hebrew texts — due to the rich morphology of Hebrew and ambiguous tokenization, simple BOW (and tf-idf) cannot provide enough semantic information. It also might be the case of political rhetoric which is similar across candidates and campaigns of different political parties. Based on our results, we can conclude that sentiment analysis obviously boosts negative campaign detection. However, there is no strong relation between offensive language and negative campaigns. Therefore, transfer learning with models trained to detect offensive content is inefficient for the detection of a negative campaign.

In the future, we plan to apply our analysis to elections to the Israeli government. We also would like to see whether cross-lingual and cross-country learning is efficient for negative campaign detection. We'd like to explore the common characteristics and differences between political campaigns in different countries. We hypothesize that an engagement of a candidate in a negative campaign can be dependent on the candidate's gender, perceived strength, initial support, etc. We intend to study these possible relations in the future.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Hadi Abdine, Yanzhu Guo, Virgile Rennard, and Michalis Vazirgiannis. 2022. Political communities on twitter: Case study of the 2022 french presidential election. In *Proceedings of The LREC 2022 workshop on Natural Language Processing for Political Sciences*, pages 62–71, Marseille, France. European Language Resources Association.

Haithem Afli, Mehwish Alam, Houda Bouamor, Cristina Blasi Casagran, Colleen Boland, and Sahar Ghannay, editors. 2022. *Proceedings of The LREC 2022 workshop on Natural Language Processing for Political Sciences*. European Language Resources Association, Marseille, France.

Joseph Asunka, Sarah Brierley, Miriam Golden, Eric Kramon, and George Ofosu. 2019. Electoral fraud or violence: The effect of observers on party manipulation strategies. *British Journal of Political Science*, 49(1):129–151.

Mateusz Baran, Mateusz WÃ³jcik, Piotr Kolebski, MichaÅ, Bernaczyk, Krzysztof Rajda, Lukasz Augustyniak, and Tomasz Kajdanowicz. 2022. Electoral agitation dataset: The use case of the polish election. In *Proceedings of The LREC 2022 workshop on Natural Language Processing for Political Sciences*, pages 32–36, Marseille, France. European Language Resources Association.

Dan Bernhardt and Meenakshi Ghosh. 2020. Positive and negative campaigning in primary and general elections. *Games and Economic Behavior*, 119:98–104.

Ekaba Bisong. 2019. *Building machine learning and deep learning models on Google cloud platform: A comprehensive guide for beginners*. Apress.

Ashish Chaturvedi. 2005. Rigging elections with violence. *Public Choice*, 125(1):189–202.

Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, et al. 2015. Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4.

François Chollet et al. 2015. Keras. https://github.com/fchollet/keras.

Avihay Chriqui and Inbal Yahav. 2021. Hebert & hebemo: a hebrew bert model and a tool for polarity analysis and emotion recognition. *arXiv preprint arXiv:2102.01909*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Giovanna Maria Invernizzi. 2019. Electoral competition and factional sabotage. *Available at SSRN 3329622*.

Marina Litvak, Natalia Vanetik, Yaser Nimer, Abdulrhman Skout, and Israel Beer-Sheba. 2021. Offensive language detection in semitic languages. In *Multimodal Hate Speech Workshop 2021*, pages 7–12.

Paul S Martin. 2004. Inside the black box of negative campaign effects: Three reasons why negative campaigns mobilize. *Political psychology*, 25(4):545–562.

Mahesh Pal. 2005. Random forest classifier for remote sensing classification. *International journal of remote sensing*, 26(1):217–222.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Eric Sanders and Antal van den Bosch. 2022. Correlating political party names in tweets, newspapers and election results. In *Proceedings of The LREC 2022 workshop on Natural Language Processing for Political Sciences*, pages 8–15, Marseille, France. European Language Resources Association.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Amit Seker, Elron Bandel, Dan Bareket, Idan Brusilovsky, Refael Shaked Greenfeld, and Reut Tsarfaty. 2021. Alephbert: A hebrew large pre-trained language model to start-off your hebrew nlp application with. *arXiv preprint arXiv:2104.04052*.

Stergios Skaperdas and Bernard Grofman. 1995. Modeling negative campaigning. *American Political Science Review*, 89(1):49–61.

Raymond E Wright. 1995. Logistic regression.