Cross-lingual Transfer Learning for Detecting Negative Campaign in Israeli Municipal Elections: a Case Study

Natalia Vanetik[1,*], Marina Litvak[1] and Lin Miao[2]

[1]Department of Software Engineering, Shamoon College of Engineering (SCE), Beer-Sheva, Israel

[2]Department of Computer Science, Beijing Information Science and Technology University, Beijing, China

Abstract

Political competitions are complex settings where candidates use campaigns to promote their chances to be elected. As we can recently observe, some candidates choose 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*. The big challenge in this area is the lack of annotated datasets for training efficient classifiers. Therefore, transfer learning from other relevant domains and other languages could be very useful for this task. Considering the recent success of meta-learning in domain adaptation, we apply it to our task of utilizing available datasets from different domains and languages. This work explores the negative campaign detection task from multiple perspectives: the efficiency of different text representations and classification models, and the effect of transfer learning from offensive language detection in different languages for negative campaign detection in Hebrew. We demonstrate that the lack of training data for negative campaign detection in a low-resourced language such as Hebrew can be compensated to some extent by available datasets for offensive language detection in the same and other languages. We report an empirical case study for political campaigns in Israeli municipal elections.[1]

Keywords

negative campaign, text classification, Hebrew, BERT, meta-learning

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 [1, 2, 3, 4].

In recent years, we witness the intensive use of negative campaigns by political candidates which target the weaknesses and failures of the opponents promising to do the opposite [2, 3, 4].

[1] Our dataset is freely available for researchers at https://github.com/NataliaVanetik1/TONIC.

In: R. Campos, A. Jorge, A. Jatowt, S. Bhatia, M. Litvak (eds.): Proceedings of the Text2Story'23 Workshop, Dublin (Republic of Ireland), 2-April-2023

[*] Corresponding author.

[†] These authors contributed equally.

natalyav@sce.ac.il (N. Vanetik); litvak.marina@gmail.com (M. Litvak); linmiao@bistu.edu.cn (L. Miao)

0000-0002-4939-1415 (N. Vanetik); 0000-0003-3044-3681 (M. Litvak); 0000-0002-9421-8566 (L. Miao)

The application of language technologies in the political sciences is recently in high demand [5]. However, despite some works dedicated to the analysis of elections-related materials [6, 7, 8], we were unable to find any work on automated negative campaign analysis and detection.

Our work reports the results of extensive experiments, aimed at answering multiple research questions: (1) Which supervised model and representation are more effective at automatically detecting negative campaigns in Hebrew? (2) Can we effectively detect negative campaigns with a model trained to identify offensive language? (3) Can meta-learning with different domains and languages boost negative campaign detection in Hebrew?

We adopt and extend the representation models applied in [9, 10, 11], where the gain of semantic vectors and sentiment knowledge for offensive language and negative campaign detection was empirically shown. In order to increase classification accuracy in a mono-domain setting, we use knowledge about cities, country districts (regions), and politicians. We use this information in a meta-learning setting as well. In [10], we have also shown the efficiency of transfer learning for cross-lingual training of offensive language classifiers with Semitic languages. We adopt and explore this idea for this study. The lack of Hebrew datasets is addressed in this study by using cross-domain and cross-lingual transfer learning, in contrast to [11].

Our contribution is multi-fold: (1) we experimented with different representations and classifiers for efficient encoding and classification of texts in Hebrew for negative campaign detection; (2) we explored the efficiency of meta-learning in mono-domain experiments; (3) we explored an efficiency of a transfer learning from offensive language detection in different languages to negative campaign detection; (4) we explored a gain of meta-learning vs. conventional fine-tuning of language models in transfer learning for cross-domain experiments.

2. TONIC dataset

The data was collected from the Facebook accounts of local politicians from several big Israeli cities running for mayor's offices. There was a total of 12 cities and 27 mayoral 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. Collected posts 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 a "negative campaign" only if it contained 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. The majority rule, i.e., the portion of the bigger class in our data, is 0.78 (the distribution between two classes is $78\% - 22\%$, with the majority class being benign texts, and the minority class containing negative campaign texts).

3. Proposed method for Negative Campaign classification

Our approach follows a standard flow of supervised learning, including text representation, model training, and its application on a test set for the model's evaluation.

Table 1
Collected data by city.

region	city	candidates	posts	pos	neg	avg words in post	avg characters in post
center	Herzliya	2	218	91	127	108.482	645.468
center	Jerusalem	3	412	32	380	72.471	428.964
center	Rishon LeZion	1	183	23	160	103.448	619.989
center	Tel Aviv	1	36	8	28	95.611	545.806
center	Petah Tikva	4	364	68	296	80.184	466.626
center	Hod Hasharon	2	266	45	221	85.128	498.432
south	Ashdod	4	363	139	224	92.377	528.044
south	Ashkelon	3	363	61	302	82.157	482.876
south	Dimona	1	50	7	43	92.280	542.240
south	Beer Sheva	1	14	9	5	192.500	1075.643
north	Netanya	4	316	81	235	72.215	427.886
north	Haifa	1	47	4	43	75.234	440.319
	Total	27	2632	568	2064	85.384	500.771

The following techniques were employed for the *post representation*:

- **Term frequency-inverse document frequency (tf-idf)**, where every post is treated as a separate document and the whole dataset as a corpus.
- **N-grams** of n consecutive words seen in the text, with $n = 1, 2, 3$.
- **BERT sentence embeddings** using one of the pre-trained BERT models—a multilingual model [12] and a Hebrew model [13]. We use BERT embeddings to represent post text, region, and city.
- **Sentiment weights** generated by the HeBERT model [14], producing a probability distribution for positive, negative, and neutral sentiments, for every post.

For classification, we experimented with three different types of *classifiers*:

- **Traditional classsifiers**, including Random Forest (RF) [15], Logistic Regression (LR) [16], and Extreme Gradient Boosting (XGB) [17].
- **Fine-tuned BERT**, including a multilingual model called *bert-base-multilingual-cased* (denoted as mBERT) [18] and AlephBERT [13], a large pre-trained language model for Modern Hebrew. Both models were fine-tuned on the train portion of our data.
- **Meta-learning**, where create a meta-model for detecting unfavorable campaigns when training data for this particular task and language is missing (or not sufficient). To quickly adapt to new target cases, ModelAgnostic Meta-Learning (MAML) [19], a general optimization framework, uses the gradient descent process to create a strong initial model. Therefore, in this study, we used MAML for meta-learning. As the foundation for our meta-learning, we use a pre-trained BERT language model as a base model. The goal of meta-learning is to train a model on a variety of learning tasks, such that it can solve new learning tasks using only a small number of training samples. We use three different criteria to split our data into training tasks: (1) an **account of politician**, where one training task aims at the identification of posts with negative campaigns published by the same politician; (2) a **city**, where a training task focuses on the data generated by politicians from the same particular city; and (3) a **region of the country**, where we

train our model on the annotated posts generated by politicians from the same region of the country.

A full pipeline of our approach is depicted in Figure 1.

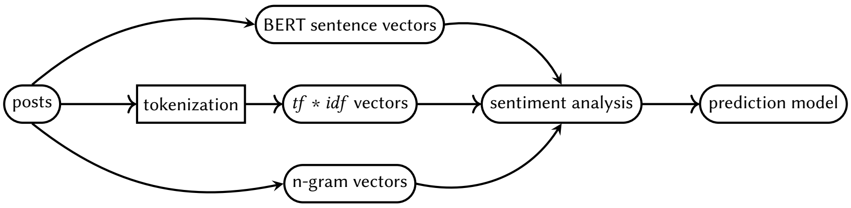

Figure 1: Political posts classification pipeline.

4. Experiments

Our experiments aim to evaluate (1) different models and representations of Hebrew data in the negative campaign domain; (2) transfer learning from the hate speech domain, in Hebrew and other languages; and (3) meta-learning approach in mono-domain and cross-domain learning.

Data and Software Setup

For the monolingual experiments on the TONIC dataset, RF, LR, and XGB are trained on 80% of the dataset and evaluated on the remaining 20%. For the cross-domain monolingual experiments, the models are trained on 100% of the other domain data and tested on 20% of the TONIC dataset. For the cross-domain cross-lingual experiments, we train our models on 100% of the data in another language, and test on the 20% of the TONIC dataset. In all cases, the test portion of the TONIC dataset is the same which allows us to conduct proper statistical significance analysis. 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 [20] called OLaH. Traditional models were implemented in sklearn [21] and neural models were implemented in Keras [22] with the TensorFlow backend [23]. Experiments were performed on google colab [24] with Pro settings.

Mono-domain Evaluation Results

Here we report the results–precision, recall, f1-measure, and accuracy scores–of the evaluation of and comparison of various models and text representations to detect negative campaigns in political posts written in Hebrew. In particular, we explore whether or not BERT sentence embeddings perform better than traditional text representations such as tf-idf and n-grams. We also compare two pre-trained BERT models to determine whether a model specifically trained in Hebrew is preferable.

Table 2 (left) summarizes the results for the conventional models and representations without sentence embeddings. All models were trained and tested on the TONIC training and test sets, respectively. The text representations use either tf-idf or n-grams (*ngX* denotes n-grams for $X = 1, 2, 3$), or their combinations (*tfidf-ngX* denotes a concatenation of tf-idf vectors with

Table 2
Evaluation of traditional models and representations on TONIC: mono-domain (left) and cross-domain monolingual (right).

	mono-domain				cross-domain monolingual			
model	P	R	F1	acc	P	R	F1	acc
RF$_{tfidf+SA}$	0.8908	0.6467	0.6813	0.8444	0.6457	0.5222	0.4931	0.7837
LR$_{tfidf+SA}$	0.8341	0.7243	0.7586	0.8615	0.8926	0.5044	0.4485	0.7856
XGB$_{tfidf+SA}$	0.8656	**0.7662**	**0.8010**	**0.8824**	**0.8933**	0.5088	0.4575	**0.7875**
RF$_{ng1+SA}$	0.8460	0.6626	0.6979	0.8444	0.5171	0.5015	0.4531	0.7761
LR$_{ng1+SA}$	0.8390	0.7601	0.7892	0.8729	0.5181	0.5068	0.4870	0.7495
XGB$_{ng1+SA}$	0.8220	0.7445	0.7726	0.8634	0.6956	**0.5215**	0.4882	**0.7875**
RF$_{ng2+SA}$	0.8633	0.6399	0.6715	0.8387	0.4819	0.4885	0.4785	0.7059
LR$_{ng2+SA}$	0.7633	0.7276	0.7424	0.8368	0.5091	0.5098	0.5090	0.6546
XGB$_{ng2+SA}$	0.7972	0.7417	0.7633	0.8539	0.4990	0.4998	0.4576	0.7685
RF$_{ng3+SA}$	0.8978	0.6260	0.6541	0.8368	0.4989	0.4994	0.4880	0.7230
LR$_{ng3+SA}$	0.7633	0.7276	0.7424	0.8368	0.5091	0.5098	0.5090	0.6546
XGB$_{ng3+SA}$	0.7972	0.7417	0.7633	0.8539	0.5098	0.5018	0.4639	0.7666
RF$_{tfidf+ng1+SA}$	0.8460	0.6299	0.6580	0.8330	0.6090	0.5166	0.4842	0.7799
LR$_{tfidf+ng1+SA}$	0.8390	0.7601	0.7892	0.8729	0.5330	0.5124	0.4948	0.7533
XGB$_{tfidf+ng1+SA}$	0.8567	**0.7681**	**0.8002**	**0.8805**	**0.8933**	0.5088	0.4575	**0.7875**
RF$_{tfidf+ng2+SA}$	0.8935	0.6128	0.6357	0.8311	0.4607	0.4856	0.4568	0.7362
LR$_{tfidf+ng2+SA}$	0.7581	0.7156	0.7325	0.8330	0.5147	0.5161	**0.5147**	0.6546
XGB$_{tfidf+ng2+SA}$	0.8317	0.7545	0.7829	0.8691	0.3916	0.4988	0.4388	0.7818
RF$_{tfidf+ng3+SA}$	**0.9097**	0.6009	0.6183	0.8273	0.5366	0.5109	0.4873	0.7609
LR$_{tfidf+ng3+SA}$	0.7581	0.7156	0.7325	0.8330	0.5147	0.5161	**0.5147**	0.6546
XGB$_{tfidf+ng3+SA}$	0.8485	0.7701	0.7994	0.8786	0.3916	0.4988	0.4388	0.7818

Table 3
Evaluation of mono-domain training on TONIC with BERT sentence embeddings.

	mBERT				AlephBERT			
model	P	R	F1	acc	P	R	F1	acc
RF$_{bert}$	0.8607	0.7052	0.7452	0.8615	0.8283	0.7231	0.7564	0.8596
LR$_{bert}$	0.8072	0.7699	0.7859	0.8634	0.8145	0.7938	0.8034	0.8710
XGB$_{bert}$	0.8059	0.7731	0.7874	0.8634	0.8160	0.7799	0.7956	0.8691
RF$_{bert+loc}$	0.8796	0.6957	0.7377	0.8615	0.8725	0.7152	0.7568	0.8672
LR$_{bert+loc}$	0.8251	0.7716	0.7933	0.8710	0.7990	0.7814	0.7896	0.8615
XGB$_{bert+loc}$	0.8523	**0.7864**	0.8125	0.8843	0.8518	0.8016	0.8227	0.8880
RF$_{bert+region}$	0.8461	0.6909	0.7287	0.8539	0.8504	0.7235	0.7611	0.8653
LR$_{bert+region}$	0.8097	0.7743	0.7896	0.8653	0.8205	0.7994	0.8092	0.8748
XGB$_{bert+region}$	0.7782	0.7324	0.7508	0.8444	0.8160	0.7799	0.7956	0.8691
RF$_{bert+region+loc}$	0.8718	0.6782	0.7178	0.8539	0.8705	0.7108	0.7522	0.8653
LR$_{bert+region+loc}$	0.8228	0.7672	0.7895	0.8691	0.7974	0.7878	0.7924	0.8615
XGB$_{bert+region+loc}$	0.8702	**0.7869**	**0.8182**	**0.8899**	0.8562	0.8028	**0.8250**	**0.8899**
RF$_{tfidf+bert}$	0.8792	0.5777	0.5827	0.8159	0.8611	0.6562	0.6915	0.8444
LR$_{tfidf+bert}$	0.8340	0.7740	0.7979	0.8748	0.8194	0.7919	0.8043	0.8729
XGB$_{tfidf+bert}$	0.8418	0.7569	0.7875	0.8729	0.8432	0.7765	0.8025	0.8786
RF$_{tfidf+bert+ng1}$	0.9057	0.5789	0.5843	0.8178	**0.8891**	0.6423	0.6756	0.8425
LR$_{tfidf+bert+ng1}$	0.8316	0.7621	0.7886	0.8710	0.8400	**0.8130**	0.8253	0.8861
XGB$_{tfidf+bert+ng1}$	0.8221	0.7521	0.7784	0.8653	0.8432	0.7765	0.8025	0.8786
RF$_{tfidf+bert+ng2}$	**0.9130**	0.6184	0.6438	0.8349	0.8816	0.6543	0.6903	0.8463
LR$_{tfidf+bert+ng2}$	0.7619	0.7169	0.7346	0.8349	0.7881	0.7532	0.7681	0.8520
XGB$_{tfidf+bert+ng2}$	0.8320	0.7470	0.7771	0.8672	0.8408	0.7872	0.8092	0.8805
RF$_{tfidf+bert+ng3}$	**0.9130**	0.6184	0.6438	0.8349	0.8677	0.6694	0.7074	0.8501
LR$_{tfidf+bert+ng3}$	0.7619	0.7169	0.7346	0.8349	0.7881	0.7532	0.7681	0.8520
XGB$_{tfidf+bert+ng3}$	0.8174	0.7509	0.7761	0.8634	0.8385	0.7752	0.8002	0.8767

n-grams of size$X = 1, 2, 3$). All the systems are significantly better than the majority rule. Also, the XGB classifier with tf-idf, unigrams, and sentiment labels outperforms the other classifiers.

Confusion matrix of the top-performing model (XGB$_{bert+region+loc}$) contains TP = 75, TN = 391, FP = 22, and FN = 39, with $precision = 0.77$ and $recall = 0.66$. 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 campaigns). As a result, TN is the most important accuracy compound, while FN represents the biggest amount of errors. In a 10 misclassified case sample that we manually examined, more than half of the errors (6), including four samples incorrectly identified as negative campaigns when we actually found them to be neutral and two samples incorrectly labeled as neutral, were the result of incorrect labeling by our annotators.

Table 3 shows the scores for the same models over sentence embeddings, produced by two different BERT models–multilingual BERT [25] and Hebrew-language AlephBERT [13]. We can see that enriched sentence embeddings of cities and regions' names boost the classification performance. XGB outperforms the other classifiers as in the previous experiment. We cannot recommend one particular BERT model, because both models seem to provide sentence embeddings with similar quality. However, when we compare these BERT models fine-tuned on the classification task on TONIC (see Table 4), AlephBERT, which is trained solely in Hebrew, significantly outperforms multilingual BERT producing accuracy which falls below the majority rule. Nonetheless, both models are outperformed by the best traditional models, probably due to less information encoded in the text representation. While both BERT classifiers use only self-produced embeddings, traditional models also utilize sentiment labels, and embeddings representing the cities and regions of the candidates.

Table 4 contains the results of meta-learning where tasks are specified by three different criteria.

Table 4
Meta-learning and fine-tuned BERT evaluation on TONIC dataset.

	Fine-tuned BERT				Meta-learning				
model	P	R	F1	acc	task split by	P	R	F1	acc
mBERT	0.6079	0.5816	0.5817	0.6641	politician	**0.6390**	**0.7994**	**0.7103**	**0.7994**
					location	0.6126	0.7827	0.6873	0.7827
					region	0.5659	0.7523	0.6459	0.7523
AlephBERT	**0.8589**	**0.7964**	**0.8190**	**0.8634**	politician	0.6055	0.7781	0.6810	0.7781
					location	0.6173	0.7857	0.6914	0.7857
					region	0.6126	0.7827	0.6873	0.7827

We can see that multilingual BERT achieves the best accuracy score; however, for all the options for task division, meta-learning scores are very close to the majority rule, that evidence that there is not much information that can be efficiently learned and transferred between tasks. We can also see that for a fine-tuned BERT, AlephBERT has a clear advantage over the multilingual BERT model in all parameters.

According to the scores in Tables 2 and 3 (we omitted meta-learning models because of their low performance), the top performing model is XGB, applied on bert embeddings enriched by region and location embeddings. In general, the XGB classifier outperforms other classifiers in most cases.

Cross-domain Mono-lingual Evaluation Results

Cross-domain mono-lingual (all models were trained and tested on Hebrew data) experiments in Table 2 (right) 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. Only a few models trained on offensive language data achieved accuracy that is slightly higher than or equal to the majority rule. Additionally, we can see that F1 scores are really low, meaning that these models simply 'guess' the majority rule.

Table 5 shows the results of the traditional models with BERT embeddings as a text representation for transfer learning from the offensive language detection in Hebrew. From Table 2 (right) and Table 5, we can conclude that (1) the XGB classifier mostly performs better than other classifiers and (2) its performance is slightly higher with BERT embeddings than with tf-idf vectors and n-grams.

Table 5
Cross-domain mono-lingual evaluation of traditional models with BERT sentence embeddings.

	mBERT				AlephBERT			
model	P	R	F1	acc	P	R	F1	acc
$\text{RF}_{tfidf-bert}$	0.6424	0.5032	0.4479	0.7837	**0.7946**	0.5163	0.4743	0.7894
$\text{LR}_{tfidf-bert}$	0.7265	0.5076	0.4568	0.7856	0.7265	0.5076	0.4568	0.7856
$\text{XGB}_{tfidf-bert}$	0.6726	0.5171	0.4800	0.7856	0.6726	0.5171	0.4800	0.7856
$\text{RF}_{tfidf-bert-ng1}$	0.3910	0.4952	0.4370	0.7761	0.6429	0.5064	0.4561	0.7837
$\text{LR}_{tfidf-bert-ng1}$	0.5463	0.5160	0.4980	0.7590	0.5463	0.5160	0.4980	0.7590
$\text{XGB}_{tfidf-bert-ng1}$	**0.7465**	0.5271	0.4973	**0.7913**	0.7609	**0.5315**	0.5053	**0.7932**
$\text{RF}_{tfidf-bert-ng2}$	0.4990	0.4998	0.4576	0.7685	0.4680	0.4955	0.4494	0.7666
$\text{LR}_{tfidf-bert-ng2}$	0.5073	0.5081	**0.5069**	0.6471	0.5073	0.5081	**0.5069**	0.6471
$\text{XGB}_{tfidf-bert-ng2}$	0.7304	**0.5302**	0.5042	0.7913	0.7304	0.5302	0.5042	**0.7913**
$\text{RF}_{tfidf-bert-ng3}$	0.6937	0.5107	0.4648	0.7856	0.4671	0.4909	0.4564	0.7495
$\text{LR}_{tfidf-bert-ng3}$	0.5073	0.5081	**0.5069**	0.6471	0.5073	0.5081	**0.5069**	0.6471
$\text{XGB}_{tfidf-bert-ng3}$	0.7304	**0.5302**	0.5042	0.7913	0.7304	0.5302	0.5042	**0.7913**

Table 6 shows the results of meta-learning trained on hate speech data and tested on the TONIC dataset. Two BERT models are initiated with the weights generated by meta-learning. The table also contains the scores of fine-tuned BERT without meta-learning.

We can see that (1) best traditional models perform better than both fine-tuned language models and meta-models when trained in foreign languages, the only exception is the recall and F1 scores of meta-learning which is evidence of its better ability to recognize the positive samples–negative political campaign–but fail at filtering out neutral posts (also confirmed by lower Precision); (2) AlephBERT performs better with meta-learning than multilingual BERT; (3) meta-learning outperforms fine-tuned language models in terms of both precision and recall.

Table 6
Meta-learning cross-domain mono-lingual evaluation.

	Fine-tuned BERT				Meta-learning			
BERT model	P	R	F1	acc	P	R	F1	acc
mBERT	0.5000	0.3918	0.4394	**0.7837**	**0.6620**	0.6793	0.6701	0.6793
AlephBERT	**0.5142**	**0.5818**	**0.4823**	0.7761	0.6126	**0.7827**	**0.6873**	**0.7827**

Cross-domain Cross-lingual Evaluation Results

Table 7 shows the evaluation of traditional models for the cross-domain cross-lingual scenario. In this setting, we train our models on hate speech datasets in other languages - English and Arabic. The only text representation that we can use here is multilingual BERT sentence embeddings generated by the pre-trained BERT model *bert-base-multilingual-cased* [18].

Table 7
Cross-domain cross-lingual evaluation of traditional models.

	OLID dataset, En				OLaA dataset, Ar			
Model	P	R	F1	acc	P	R	F1	acc
RF	**0.6096**	**0.5085**	0.1965	0.2296	0.4804	0.4812	0.4807	0.6546
LR	0.5082	0.5005	0.1872	0.2220	0.4224	0.4672	0.4365	**0.7173**
XGB	0.5535	0.5053	0.1978	0.2296	0.5109	0.5072	**0.5019**	0.7154

Table 8 shows the results of meta-learning trained on hate speech data in other languages (Arabic and English) and tested on the TONIC dataset. An English-language dataset is the Offensive Language Identification Dataset (OLID) [26], which is a collection of 14,100 tweets (we used 13,240 annotated tweets from its training set). We used the OLaA dataset in Arabic, which we collected and introduced in [9] previously. OLaA is a collection of 9,000 comments from Twitter annotated for hate speech. We used a multilingual BERT model [18] for these experiments. For comparison, we also show the scores of this BERT model fine-tuned on Arabic and English hate-speech data and tested on TONIC.

Both experiments evidence that meta-learning adapts pre-trained models much better to the new domains than traditional fine-tuning and it can be efficiently applied for transfer learning from other domains and even languages. In particular, we can observe the following: (1) fine-tuned language models and meta-learning perform better than best traditional models when trained on foreign languages; (2) meta-learning outperforms fine-tuned language models.

Table 8
Cross-domain cross-lingual evaluation of meta-learning.

		Fine-tuned BERT				Meta-learning			
Data	lang	P	R	F1	acc	P	R	F1	acc
OLaA	Ar	0.4785	0.4250	0.4392	0.7400	0.6245	0.7903	0.6977	0.7903
OLID	En	**0.6102**	**0.7812**	**0.6852**	**0.7812**	**0.6342**	**0.7964**	**0.7061**	**0.7964**

5. Future Work and Conclusions

Based on the results of extensive experiments aimed to answer various research questions (see Section 1), we can conclude that (1) the best combination of text representation and classification model for negative campaign detection in Hebrew texts is XGB with sentence embeddings enriched with region and location information; (2) transfer learning with models trained to detect offensive content is inefficient for the detection of a negative campaign; meaning that there is no strong relation between offensive language and negative campaigns; (3) transfer learning from different languages can be applied to Hebrew in the negative campaign detection task, while training on a large set in a foreign language can be even more efficient than training on Hebrew; and (4) meta-learning outperforms language models traditionally fine-tuned in cross-domain and cross-lingual scenarios, but not in a mono-lingual setting. We also observe that in a monolingual setting that employs either a fine-tuned BERT or BERT sentence embedding, the AlephBERT model trained on Hebrew is preferable to a multilingual BERT model. In the future, we plan to apply our analysis to elections for the Israeli government, to explore the common characteristics and differences between political campaigns in different countries, and to study possible relations between the candidate's gender, perceived strength, initial support, etc. and their engagement in a negative campaign.

References

[1] D. Bernhardt, M. Ghosh, Positive and negative campaigning in primary and general elections, Games and Economic Behavior 119 (2020) 98–104.

[2] G. M. Invernizzi, Electoral competition and factional sabotage, Available at SSRN 3329622 (2019).

[3] P. S. Martin, Inside the black box of negative campaign effects: Three reasons why negative campaigns mobilize, Political psychology 25 (2004) 545–562.

[4] S. Skaperdas, B. Grofman, Modeling negative campaigning, American Political Science Review 89 (1995) 49–61.

[5] H. Afli, M. Alam, H. Bouamor, C. B. Casagran, C. Boland, S. Ghannay (Eds.), Proceedings of The LREC 2022 workshop on Natural Language Processing for Political Sciences, European Language Resources Association, Marseille, France, 2022. URL: https://aclanthology.org/2022.politicalnlp-1.

[6] M. Baran, M. WÃ³jcik, P. Kolebski, M. Bernaczyk, K. Rajda, L. Augustyniak, T. Kajdanowicz, Electoral agitation dataset: The use case of the polish election, in: Proceedings of The LREC 2022 workshop on Natural Language Processing for Political Sciences, European Language Resources Association, Marseille, France, 2022, pp. 32–36. URL: https://aclanthology.org/2022.politicalnlp-1.5.

[7] H. Abdine, Y. Guo, V. Rennard, M. Vazirgiannis, 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, European Language Resources Association, Marseille, France, 2022, pp. 62–71. URL: https://aclanthology.org/2022.politicalnlp-1.9.

[8] E. Sanders, A. van den Bosch, Correlating political party names in tweets, newspapers and election results, in: Proceedings of The LREC 2022 workshop on Natural Language Processing for Political Sciences, European Language Resources Association, Marseille, France, 2022, pp. 8–15. URL: https://aclanthology.org/2022.politicalnlp-1.2.

[9] M. Litvak, N. Vanetik, Y. Nimer, A. Skout, Offensive language detection in semitic languages, in: 1st CFP:Multimodal and Multilingual Hate Speech Detection workshop at KONVENS 2021, 2021, pp. 7–13.

[10] M. Litvak, N. Vanetik, C. Liebeskind, O. Hmdia, R. A. Madeghem, Offensive language detection in hebrew: can other languages help?, in: Proceedings of the Language Resources

and Evaluation Conference, European Language Resources Association, Marseille, France, 2022, pp. 3715–3723. URL: https://aclanthology.org/2022.lrec-1.396.

[11] M. Litvak, N. Vanetik, S. Talker, O. Machlouf, Detection of negative campaign in israeli municipal elections, in: Proceedings of the Third Workshop on Threat, Aggression and Cyberbullying (TRAC 2022), 2022, pp. 68–74.

[12] V. Sanh, L. Debut, J. Chaumond, T. Wolf, Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, arXiv preprint arXiv:1910.01108 (2019).

[13] A. Seker, E. Bandel, D. Bareket, I. Brusilovsky, R. S. Greenfeld, R. Tsarfaty, Alephbert: A hebrew large pre-trained language model to start-off your hebrew nlp application with, arXiv preprint arXiv:2104.04052 (2021).

[14] A. Chriqui, I. Yahav, Hebert & hebemo: a hebrew bert model and a tool for polarity analysis and emotion recognition, arXiv preprint arXiv:2102.01909 (2021).

[15] M. Pal, Random forest classifier for remote sensing classification, International journal of remote sensing 26 (2005) 217–222.

[16] R. E. Wright, Logistic regression, in: L. G. Grimm, P. R. Yarnold (Eds.), Reading and understanding multivariate statistics, American Psychological Association, 1995, pp. 217–244.

[17] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, et al., Xgboost: extreme gradient boosting, R package version 0.4-2 1 (2015) 1–4.

[18] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, CoRR abs/1810.04805 (2018). URL: http://arxiv.org/abs/1810.04805. arXiv:1810.04805.

[19] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: International conference on machine learning, PMLR, 2017, pp. 1126–1135.

[20] M. Litvak, N. Vanetik, Y. Nimer, A. Skout, I. Beer-Sheba, Offensive language detection in semitic languages, in: Multimodal Hate Speech Workshop 2021, 2021, pp. 7–12.

[21] 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, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[22] F. Chollet, et al., Keras, https://github.com/fchollet/keras, 2015.

[23] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL: https://www.tensorflow.org/, software available from tensorflow.org.

[24] E. Bisong, Building machine learning and deep learning models on Google cloud platform: A comprehensive guide for beginners, Apress, 2019.

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

[26] M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, R. Kumar, Predicting the Type and Target of Offensive Posts in Social Media, in: Proceedings of NAACL, 2019, p. 1415–1420.