Proceedings of AESPEN 2020, pages 9–18
Language Resources and Evaluation Conference (LREC 2020), Marseille, 11–16 May 2020

Analyzing ELMo and DistilBERT on Socio-political News Classification

Berfu Büyüköz, Ali Hürriyetoğlu, Arzucan Özgür
Boğaziçi University, Koç University, Boğaziçi University
İstanbul, Turkey
{berfu.buyukoz, arzucan.ozgur}@boun.edu.tr
ahurriyetoglu@ku.edu.tr

Abstract

This study evaluates the robustness of two state-of-the-art deep contextual language representations, ELMo and DistilBERT, on supervised learning of binary protest news classification (PC) and sentiment analysis (SA) of product reviews. A "cross-context" setting is enabled using test sets that are distinct from the training data. The models are fine-tuned and fed into a Feed-Forward Neural Network (FFNN) and a Bidirectional Long Short Term Memory network (BiLSTM). Multinomial Naive Bayes (MNB) and Linear Support Vector Machine (LSVM) are used as traditional baselines. The results suggest that DistilBERT can transfer generic semantic knowledge to other domains better than ELMo. DistilBERT is also 30% smaller and 83% faster than ELMo, which suggests superiority for smaller computational training budgets. When generalization is not the utmost preference and test domain is similar to the training domain, the traditional machine learning (ML) algorithms can still be considered as more economic alternatives to deep language representations.

Keywords: deep language representations, news text classification, sentiment analysis.

1. Introduction

A challenge the Natural Language Processing (NLP) community faces today is to leverage NLP systems from a well-maintained test environment to more realistic scenarios full of dynamism and diversity (Ettinger et al., 2017; Hürriyetoğlu et al., 2019a). An NLP system should generalize well to data coming from diverse sources differing in time and space.

In the quest of building generalizable systems, the NLP community attempts building task-agnostic models in an unsupervised manner to represent generic syntactic and semantic knowledge of a language. One of the solutions is to create one big universal language representation and use it as the initialization point for any NLP task.

One example of unsupervised language representations is the famous `word2vec` (Mikolov et al., 2013), which creates continuous word vectors for each word in the vocabulary derived from a large corpus in a fully unsupervised manner by utilizing context information regarding the neighboring words. `word2vec` creates fixed vectors for each unique word in the vocabulary. In this sense, it lacks representing the dynamism of the word meaning that changes depending on the enclosing context.

The contextualization notion is the key to create universal language representations that can handle the rich syntactic and semantic space of real-life language usage. In this respect, in the last couple of years, several deep contextual neural architectures have been proposed, which have been shown to perform surprisingly well on a diverse range of downstream NLP tasks (Peters et al., 2018; Devlin et al., 2019; Radford et al., 2019).

However, there is still much to do to understand the true capacity of these representations. The true limits of these networks must be explored to understand how to build the next-generation systems. Digging into these models might even shed light on the general language understanding phenomena itself on a cognitive level (Greenwood, 1992; Kell et al., 2018). For this reason, exhaustive evaluation and interpretation studies are needed to be performed on as many different data and task sets as possible.

This study is conducted to contribute to the extrinsic evaluation of the robustness of two of these representations, namely, ELMo and DistilBERT, by testing them on a binary classification of cross-context socio-political and local news data, where the source and target data differ in the originating country and domain (Hürriyetoğlu et al., 2019a).

This study aims to answer the following questions:

1. How robust are ELMo and DistilBERT in the cross-context socio-political news classification?
2. Are contextual representations better in the cross-context than much smaller and faster traditional baselines?
3. Which one is more scalable in terms of model size and training time: ELMo or DistilBERT?

The following conclusions are reached under the limitations of the experimental setup (See Section 3.):

1. DistilBERT is more robust than ELMo in the cross-context.
2. Both ELMo and DistilBERT outperform the baselines, namely, Multinomial Naive Bayes (MNB) and Linear Support Vector Machine (LSVM), in generalizing to the cross-context.
3. DistilBERT is more efficient than ELMo with 30% smaller size and on average for the two addressed tasks, 83% faster training and testing time.
4. Traditional methods like MNB and LSVM can still compete with contextual embeddings when training and test data do not differ much.

This study compares language representations and model performance on both a sentiment analysis task and a recently proposed task set that is realized around a recent news data set: classifying protest news on local news data sets consisting of multiple sentences and coming from different country sources. A "cross-country" evaluation setting is realized by testing a model on a news text coming from a different country than the training data (Hürriyetoğlu et al., 2019a). While this study is one among many that compare word representations for text classification, this study diverges from most previous works by evaluating cross-context performance in a novel domain.

2. Tasks and Data

The transfer capacity of ELMo and DistilBERT are explored under the light of two distinct text classification tasks, each realized under a cross-context experimental setting. One is a document-level binary text classification task that is to classify English news articles from local newspapers of India and China (Hürriyetoğlu et al., 2019a). The other is to classify sentence-level Rotten Tomatoes movie (Pang and Lee, 2005) and customer reviews (Hu and Liu, 2004).

The "null context" refers to the India news and movie reviews data sets. The "cross-context" refers to the China news and customer reviews data sets. For both tasks, the null context data splits abide by the 75% - 10% - 15% proportions for training, test, and development sets, respectively. All cross-context data are used to test models trained on null-context data's train portion.

2.1. Protest News Classification

The task was designed as an auxiliary task for an active research project (Hürriyetoğlu et al., 2019a), the main motivation of which is to automate creation of protest events database from diverse sources using NLP and Machine Learning (ML) to enable a comparative political and sociological study. A shared task set, namely, CLEF-2019 Lab ProtestNews on Extracting Protests from News, was accordingly organized to address the challenge of building NLP tools that are generalizable to different test data. A cross-country evaluation setting was realized by training a model on local newspapers of India and testing the model on local newspapers of China.

The data consists of local India and China document-level news articles in the English language. Training, validation, and test splits are provided by the shared task organizers. Each news article is annotated as whether it is about a protest event or not. As illustrated in Table 1, the India data is imbalanced with 22% protest class, the China data is even more imbalanced with 5% protest class.

2.2. Challenges of Political Context

In previous work, it is seen that the classification of contentious political events could be confusing to even domain experts and the inter-annotator agreement could be surprisingly low (King and Lowe, 2003). That confusion mostly comes from the ambiguity in political terms. How a political event could be interpreted can highly depend on local culture, language usage, time, space and actors. Adding the style and biases of the author of the news text, even a single annotator may not be completely sure of his/her annotations, let alone agreeing with fellow annotators.

Data Subset	Size	Protest Ratio
Ntrain	3430	0.22
Ndev	457	0.22
Ntest	687	0.22
Ctest	1800	0.05

Table 1: Protest news data statistics. Ntrain, Ndev, and Ntest refer to training, development, and test splits of the null context data of the tasks, respectively. Ctest refers to the cross-context data.

Within the context of contentious politics, "protest" can be very broadly defined as engaging in a political dissent via numerous actions such as demonstrating for rights, rallying for political change, conducting a hunger strike, boycotting rights, and so forth.

> Members of the Communist Party of India (Marxist) staged a **protest** here on Thursday **demanding** the arrest of the remaining accused in the last month's assault on party-goers at the Morning Mist Homestay in Padil and filing of cases under the **Goonda Act** against them. The protest meeting was preceded by a rally from Town Hall to …

Figure 1: India news sample.

2.2.1. Local News Data

Political events are strongly connected to their local context. Concerning protest news classification (PC), it should be noted that protests might manifest through different kinds of actions in different cultures. In Figure 1, the news mentions a protest activity as "Goonda act", which is a term used in the Indian subcontinent for a hired criminal. In this sense, analyzing local data of many countries can be useful and mostly becomes a necessity to converge to a realistic model of what protest means both globally and locally.

2.2.2. Small Data

Contextual language representations are known to have the potential to substantially reduce the required training data size to create satisfactory models via task-specific fine-tuning on small data. As illustrated in Table 1. the protest news data is also fairly small, with the number of training samples less than 10000 (both local and cross-country data sets).

2.2.3. Long Text

The protest news data set consists of fairly long samples with 300 tokens on average.[1] This may affect the model performance in two different ways: A model may fail to learn long term relationships within the text or a model may simply not be able to utilize the whole text due to memory

[1] Here,"token" is used as a generic term for a unit output of a sequence tokenization process.

issues. In this case, very important parts of the data might be lost. For example, the news sample in Figure 2 was classified falsely as "non-protest", since the "protest" keyword was clipped due to the limitation to maximum number of tokens.

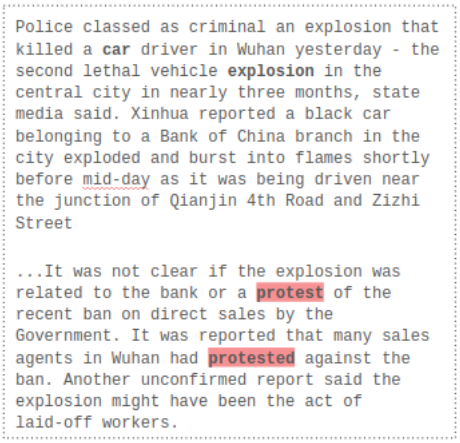

Figure 2: China news sample.

2.3. Sentiment Analysis

The other task addressed in this paper is to classify sentence-level Rotten Tomatoes movie (MR) (Pang and Lee, 2005) and customer reviews (CR) (Hu and Liu, 2004) as "positive" or "negative". The models are trained and tested on sentence-level MR (Pang and Lee, 2005) in the null context, and tested on sentence-level CR (Hu and Liu, 2004) in the cross-context.

Both sentiment data sets were exhaustively used earlier (Kiros et al., 2015; Zhao et al., 2015; Conneau et al., 2017; Conneau and Kiela, 2018; Logeswaran and Lee, 2018; Hill et al., 2016). But in none of these studies a cross-context setting is realized. They obtained the result via direct supervision on the target tasks.

Data Subset	Size	Positive Ratio
Ntrain	7974	0.5
Ndev	1088	0.5
Ntest	1600	0.5
Ctest	3771	0.64

Table 2: Sentiment data statistics. Ntrain: Training split of MR data set. Ndev: Development split of MR data set. Ntest: Test split of MR data set. Ctest: CR data set as the cross-context test data.

3. Experimental Setup

Four experiments are applied to better understand the cross-context performance of the models. The classifiers are implemented in the Python programming language using the PyTorch library.[2]

[2] `https://pytorch.org/`

3.1. ELMo

ELMo (Peters et al., 2018) is a deep context-dependent representation learned from the internal states of a deep bidirectional language model that is acquired by the joint training of two LSTM layers on both directions. This study makes use of the original pretrained ELMo model with 2 layer bidirectional LSTM layers with 4096 units and 512-dimensional projections, with a total of 93.6 million parameters. ELMo's hidden LSTM layers are weighted averaged and then fed into the classifier layers.

3.2. DistilBERT

DistilBERT (Sanh et al., 2019) is created by applying knowledge distillation to BERT (Devlin et al., 2019), specifically the `bert-base-uncased` model. To create a smaller version of BERT, DistilBERT's creators removed the token-type embeddings and the pooler from the architecture and reduced the number of layers by a factor of 2. In this study, DistilBERT's last four hidden layers are simply averaged and fed into the classifier layers, which is a suggested usage of BERT for text classification tasks.

In this study, `distilbert-base-uncased`[3] with 66 million parameters is compared to the original ELMo model with 93.6 million parameters.[4]

3.3. Classifiers

The classifier architectures are kept simple to focus on what information can be easily extracted from ELMo and DistilBERT. First, a 2-layer FFNN with 512 hidden units is used. Then, to better understand the effect of adding task-trained contextualization, a 2-layer BiLSTM with 512 hidden units is added before the linear output layer. The default maximum sequence length is 256 tokens for PC, 60 tokens for SA. ELMo gets that many full tokens, whereas DistilBERT gets that many WordPiece (Wu et al., 2016) outputs. The architectures are visualized in Figure 3.

3.4. Baseline Models

Optimized LSVM and MNB scores are reported as baselines.[5] LSVM takes the input as tf-idf (term frequency - inverse document frequency) vectors, whereas MNB as a sparse vector of token counts.

The baseline models are much simpler than the neural classifiers described in Figure 3. The baseline models utilize simple word representations which do not preserve word order and context information. By comparing traditional ML algorithms to heavily pretrained large contextual networks, we aim at understanding if the overhead of the deep contextual models is worth to undertake in this task.

3.5. Tokenization

Except for DistilBERT, the sequences are tokenized by Spacy's `en-core-web-sm` tokenizer[6]. DistilBERT uses

[3] `https://github.com/huggingface/transformers`

[4] `https://allennlp.org/elmo`, accessed in March 2020.

[5] `https://scikit-learn.org/stable/`, accessed in March 2020.

[6] `https://spacy.io/`.

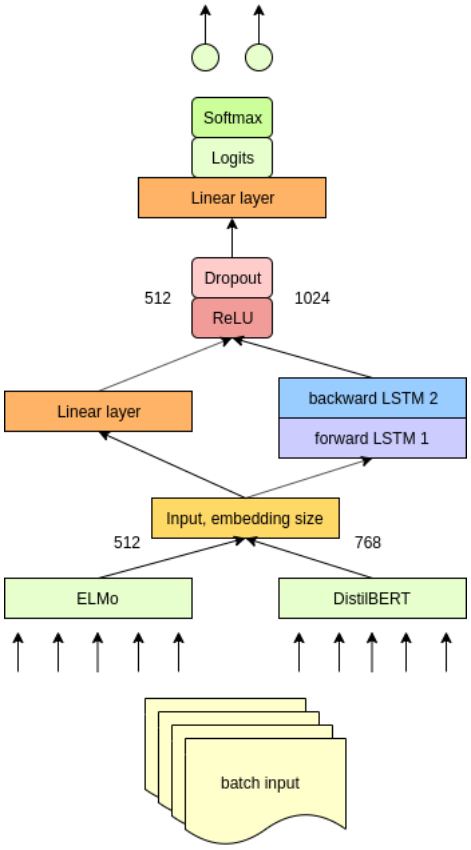

Figure 3: Classifier architecture. These are two distinct classifiers only visualized intersecting on the common layers.

WordPiece tokenization. The first 256 tokens and 60 tokens per sample are given as input to the classifiers for the PC and sentiment analysis (SA) tasks, respectively. Note that the usage of two different tokenizers causes a mismatch between the input of DistilBERT and other models. But WordPiece tokenization is preferred for DistilBERT as it is the default tokenizer of it. No text pre-processing is performed on the texts (such as casing, stop word removal, stemming, etc.). Out-of-sample tokens are not specially treated in training FFNN and BiLSTM since ELMo and DistilBERT already take care of those: the former with character-based tokenization, the latter with WordPiece tokenization. In baseline models, out-of-vocabulary tokens were simply not propagated to the classifier.

3.6. Hyper-parameter Tuning

The hyper-parameters of each distinct model are optimized on the validation data with the Tree-structured Parzen Estimator algorithm. The implementation of the algorithm is provided by the hyperopt package.[7]

The hyper-parameter tuning for the baselines was straightforward, for there were few possible hyper-parameters to be tuned as seen in Table 4.

[7] `https://github.com/hyperopt/hyperopt`, accessed in March 2020.

HParam	Range
learning rate	5e-5, 1e-3, 1e-1
learning rate decay	0, 0.5
dropout	0, 0.25, 0.5
L2	0, 0.01
Use ReLu?	True, False

Table 3: Hyper-parameter space.

Model	HParam	Range
MNB	alpha	0, 0.25, 0.5, 0.75, 1
MNB	fit-prior	True, False
LSVM	loss	hinge, squared hinge
LSVM	tolerance	1e-2, 1e-3, 1e-4
LSVM	C	0.5, 1

Table 4: Hyper-parameter space of the baselines.

3.7. Training

The training is done on a single V100 NVIDIA GPU with 16 GB RAM. The classifiers are trained for 10 epochs with the Adam optimizer (Kingma and Ba, 2014) using step decay with the patience of 3 epochs. The best model is checkpointed regarding the development set F-score. Then the checkpoints are evaluated on the test data. This procedure is repeated for each classifier with 5 random seeds and the average scores are reported.

3.8. Experiments

All experiments report both null and cross-context results for each task. Each experiment focuses on a particular variation on the classifier architecture that possibly affects the results in its way. First, both ELMo and DistilBERT are used as fixed (with *frozen* weights) word vectors and fed into FFNN. Then, they are fine-tuned to the training data sets together with the FFNN classifier. In the third setting, both models are kept *frozen* (the weights of the language models are not updated during training), but this time paired with a BiLSTM instead of an FFNN. Lastly, they are compared under the combined effect of fine-tuning contextual embeddings and pairing with a 2-layer BiLSTM.

Macro averaged F-score ($\beta = 1$) is used as the primary evaluation metric in both tasks since it provides a more robust evaluation for class-imbalanced data. Also as an additional metric, the F-score drop between null and cross-context is tracked in percentages. That is, for example, if model x null context F-score is f_n and its cross-context setting F-score is f_c, then the drop in F-score is calculated as $(f_n - c_n)/f_n * 100$. This metric helps reveal the true cross-context performance in some cases where absolute F-scores fail to do so.

The dropout rate of the classifier (FFNN or BiLSTM), learning rate, learning rate decay, L2 norm, and whether to use ReLU or not, are the hyper-parameters that were tuned for each model. Hyper-parameters of ELMo and DistilBERT are kept unchanged.

4. Experiment Results

In this section, ELMo and DistilBERT are compared using various classification architectures on two cross-context

text classification tasks.

4.1. Experiment 1 - Frozen Embeddings

Table 5 shows that frozen DistilBERT is on par with or exceeding frozen ELMo in the null context (India and MR test sets). On the other hand, DistilBERT outperforms ELMo in the cross-contexts (China and CR sets) with a smaller "Drop" score in both tasks.

Task	Model	Ntest	Ctest	Drop
PC	ELMo 256	83.6	75.2	10
PC	DBERT 256	**83.8**	**76.8**	**8.2**
SA	ELMo	78	63.6	18.4
SA	DBERT	**79**	**66.8**	**15.4**

Table 5: Experiment 1 results. Frozen ELMo and DistilBERT combined with FFNN.

4.2. Experiment 2 - Fine-tuned Embeddings

In this stage, ELMo and DistilBERT are fine-tuned on the training data sets together with the FFNN classifier.
Fine-tuned ELMo does not fit into a single GPU with 256 tokens per input sample. In this case, ELMo could manage up to 150 tokens per input. For a fair comparison, DistilBERT is trained twice, first with 150 tokens of input, and then as a separate model, with 256 tokens of input.

Task	Model	Ntest	Ctest	Drop
PC	ELMo ft 150	83	72.2	13
PC	DBERT ft 150	80	71	11.3
PC	DBERT ft 256	**83.2**	**76.4**	**8.2**
SA	ELMo ft	76.2	**69**	**9.6**
SA	DBERT ft	**79**	68	14

Table 6: Experiment 2 results. Fine-tuned ELMo and DistilBERT combined with FFNN.

As illustrated in Table 6, when the context is restricted to 150 tokens, fine-tuned ELMo outperforms DistilBERT, but falls behind in 256 tokens especially in cross-context. On the other hand, in SA, DistilBERT surpasses ELMo in the null context, but falls behind in the cross-context. This indicates that in SA, fine-tuning made ELMo more robust to context change in the test set.

4.3. Experiment 3 - External Contextualization via BiLSTM

In this experiment, both models are kept frozen, but this time paired with a BiLSTM instead of an FFNN. BiLSTM adds contextualization on the focused task, thus it is expected to improve results.

Task	Model	Ntest	Ctest	Drop
PC	ELMo 256	81.6	72.4	11.2
PC	DBERT 256	**84.2**	**78.4**	**7**
SA	ELMo	79	67	15.2
SA	DBERT	**80**	**70.2**	**12.4**

Table 7: Experiment 3 results. Frozen ELMo and DistilBERT combined with BiLSTM.

As Table 7 illustrates, in both tasks DistilBERT outperforms ELMo when paired with a 2-layer BiLSTM. The gap is more visible in the cross-context performance: DistilBERT surpasses ELMo 6 points with a 78.4 Ctest F-score on the PC task.

4.4. Experiment 4 - Combining Fine-tuning with BiLSTM

In this experiment, ELMo and DistilBERT are compared under the combined effect of fine-tuning and the usage of 2-layer BiLSTM. In PC, ELMo could handle at most 150 tokens per input. Therefore, the comparison is done under that much of a sequence length.

Task	Model	Ntest	Ctest	Drop
PC	ELMo ft 150	**82**	72	12.2
PC	DBERT ft 150	81.8	**72.2**	**11.8**
SA	ELMo ft	78.2	67.4	13.8
SA	DBERT ft	**80**	**70.2**	**12.4**

Table 8: Experiment 4 results. Fine-tuned ELMo and DistilBERT combined with BiLSTM.

In Experiment 2, DistilBERT was underperforming on sequences of length 150 in PC. Now, as illustrated in Table 8 DistilBERT catches up with ELMo. This indicates that DistilBERT benefits from BiLSTM.

4.5. Comparison to Baselines

For fairness, both ELMo's and DistilBERT's best and worst-performing configurations are compared to the hyper-parameter-tuned MNB and LSVM baselines. The best performing models are indicated with the keywords "highest", the worst-performing with "lowest" in Table 10. Two models are reported as the "highest" of ELMo in SA as one owns better "Drop" scores.

Task	Model Tag	Model Name
PC	ELMo (lowest)	ELMo + BiLSTM 256
PC	ELMo (highest)	ELMo 256
PC	DBERT (lowest)	DBERT ft 256 (lowest)
PC	DBERT (highest)	DBERT 256
SA	ELMo (lowest)	ELMo
SA	ELMo (highest 1)	ELMo ft
SA	ELMo (highest 2)	ELMo + BiLSTM
SA	DBERT (lowest)	DBERT
SA	DBERT (highest)	DBERT ft + BiLSTM

Table 9: Names of worst and best performing models.

Table 10 demonstrates that in PC, while LSVM cannot catch up with any model, MNB performs fairly on par with ELMo's worst-performing model. Apart from that, MNB is effectively surpassed by the best of ELMo and DistilBERT in all categories. In SA, MNB is inferior to all models. The results of LSVM and ELMo's lowest are close to each other. But the best of ELMo and all variants of DistilBERT surpass the LSVM baseline with an apparent gap in the cross-context robustness.
It is also visible that DistilBERT outperforms ELMo on both tasks with both of its worst-performing and best-

Task	Model	Ntest	Ctest	Drop
PC	LSVM 256	79	64	19
PC	MNB 256	80	73	9
PC	ELMo (lowest)	81.6	72.4	11.2
PC	ELMo (highest)	83.6	75.2	10
PC	DBERT (lowest)	83.2	76.4	**8.2**
PC	DBERT (highest)	**83.8**	**76.8**	**8.2**
SA	MNB	78	57	27
SA	LSVM	77	62	19
SA	ELMo (lowest)	78	63.6	18.4
SA	ELMo (highest 1)	76.2	69	**9.6**
SA	ELMo (highest 2)	79	67	15.2
SA	DBERT (lowest)	79	66.8	15.4
SA	DBERT (highest)	**80**	**70.2**	12.4

Table 10: Comparison with the baselines.

performing variants. This can be viewed as an indicator of the possible superiority of DistilBERT.

4.6. Average Scores

To view the experiments from a wider perspective, the models are also compared under the arithmetic average of all variations. As Table 11 displays, ELMo is found to be superior to DistilBERT on average when both use only 150 tokens of protest news input.[8] But in SA when full context is available DistilBERT performs better regardless of short sequence length. On average of common variations, DistilBERT is dominant in both tasks. This can be seen as an indicator of DistilBERT's overall superiority.

Task	Average	Ntest	Ctest	Drop
PC	ELMo 150	81.95	73.25	10.55
PC	DBERT 150	80.95	72.8	10
PC	ELMo	**82.17**	73.43	10.57
PC	DBERT	81.97	**74.4**	**9.2**
SA	ELMo	77.85	66.75	14.25
SA	DBERT	**79.5**	**68.8**	**13.6**

Table 11: Average scores of ELMo and DistilBERT.

4.7. Training Time and Model Size

Training times and model sizes are compared by averaging all model configurations common to ELMo and DistilBERT. Training and inference time are summed up to a single number. According to Table 12, DistilBERT is 30% smaller and 83% faster than ELMo on the average of both tasks. In terms of classifier size (excluding embeddings) DistilBERT is 13% smaller than ELMo. On the other hand, MNB and LSVM are far more efficient than DistilBERT in size and speed by being 99% smaller and 96% faster.

4.8. New State-of-the-art in CLEF-2019 Lab ProtestNews

Combining contextual embeddings with standard shallow neural networks (FFNN and BiLSTM) and applying hyper-

[8]For fairness, DistilBERT's fine-tuned models making use of 256 length input are excluded from the computation because there is no equivalent model on the ELMo side.

Task	Model	ESize	MSize	Ttime
PC	MNB	-	**1.1**	**12**
PC	ELMo	358	75.55	1690
PC	DBERT	254	65.8	318
SA	LSVM	-	**0.133**	**10**
SA	ELMo	358	75.55	979
SA	DBERT	254	65.8	237

Table 12: Average training time and model sizes of ELMo and DistilBERT. ESize: Embedding size. MSize: Model size. TTime: Train time. Sizes are in Megabytes. Train time is in seconds.

parameter tuning helped outrun the prior results in the CLEF-2019 Lab ProtestNews in cross-context while getting comparable results in null context. As shown in Table 13, F-score in China test set increased from 65 to 76.8 F-score; "Drop" is diminished from 22% to 8.2%.

Model	Ntest	Ctest	Drop
(Radford, 2019)	83	65	22
DBERT 256	**83.8**	**76.8**	**8.2**

Table 13: Comparison with CLEF-2019 Lab ProtestNews results. The prior state-of-the-art is exceeded in cross-context.

5. Randomization Test

The randomization test (Yeh, 2000) is applied to the results to check if the models significantly differ in terms of scores. The randomization test is performed by calculating p-values for all combinations of predictions obtained by training with different seeds. For example, when two models of ELMo and DistilBERT are compared, 25 different p-values are produced by using 25 different pairs of 5 ELMo and 5 DistilBERT outcomes. The harmonic mean of these p-values is used as the ultimate statistic of the test to smooth the disproportional effect of large p-values occuring in arithmetic mean.

The harmonic mean of a series equals to zero if the series contains any zero value. For more realistic evaluation, the harmonic mean of non-zero p-values are also reported (Ntest-p, Ctest-p, Drop-p). For example, if a randomization output contains at least one zero value, the true harmonic mean becomes automatically zero. In that case, we also include the harmonic mean found after excluding zero values. The results are reported in Tables 14 and 15 by separating those alternative results by / (e.g. $0/0.01$). Nevertheless, zero values should not be entirely ignored since their existence points out that rejection of the null hypothesis is indeed very much probable.

It should be noted that for PC two-tailed randomization tests general statistics (both positive and negative class) show that there is no significant difference between ELMo and DistilBERT's Ntest and Ctest performance ($p = 0.38$ and $p = 0.59$, respectively). But, since negative class ratio is much larger than positive class ratio in protest news data (see Table 1), it dominates two-tailed tests. We conducted

one-tailed test only with positive (protest) class instances in PC task to get more realistic results for the positive class.
Tables 14 and 15 suggest that, according to the randomization tests, DistilBERT is significantly better ($\alpha = 0.05$) in both null and cross-context for positive class in PC and both classes in SA. ELMo, in turn, is observed to be superior to the baselines in cross-context. But ELMo is on par with the baselines in null context.

Models	Task	Ntest-p	Ctest-p	Drop-p
ELMo-DBERT	PC	**0/0.01**	**0/0.004**	**0/0.006**
ELMo-MNB	PC	**0.007**	**0/0.009**	**0.004**

Table 14: PC - One-tailed randomization test p-value results on the best performing model variations of ELMo, DistilBERT, and MNB. A value is made bold if it can reject the null hypothesis. Ntest-p, Ctest-p, and Drop-p stand for "positive (protest) class statistics."

Models	Task	Ctest	Drop
ELMo-DBERT	SA	**0/0.017**	**0.02/0.02**
ELMo-LSVM	SA	**0/0**	**0/0.009**

Table 15: SA - One-tailed randomization test p-value results on the best performing model variations of ELMo, DistilBERT, and LSVM. A value is made bold if it can reject the null hypothesis.

6. Related Work

This section overviews the previous work that is focused on understanding the generalization capacity of the contextual language representations.

6.1. Evaluating Transfer Capacity of Language Models

The evaluation studies before this work are generally designed around a diverse set of downstream tasks (Devlin et al., 2019; Liu et al., 2019; Tenney et al., 2019; Peters et al., 2019) or ablation studies (Liu et al., 2019). Sun et al. (2019) focus on the effective tuning methods of pretrained representations, while Howard and Ruder (2018) propose a set of parameter tuning techniques specifically to leverage text classification performance. Han and Eisenstein (2019) apply unsupervised domain adaptation by further pretraining contextual representations on the masked language model on the target domain. Tenney et al. (2019) observed that contextual embeddings substantially improve over traditional baselines on learning the syntactic structure of text, but that there is only a small improvement in learning semantics on token and sentence level tasks.

6.2. Cross-context Protest Event Text Analysis

A task set (Hürriyetoğlu et al., 2019a) was proposed to collect protest event information from news texts to create systems that learn transferable information to extract relevant information from multiple countries with the ultimate motivation to create tools to enable comparative sociology and political studies on social protest phenomena. The task set consists of three tasks: news articles classification, event sentence detection, and event information extraction.
The protest news classification task was realized in the CLEF-2019 Lab ProtestNews on Extracting Protests from News (Hürriyetoğlu et al., 2019c) in the context of generalizable natural language processing[9]. From the results gathered from 12 teams, it was observed that Neural Networks obtained the best results and a significant drop in cross-country performance is observed on the news from China (Hürriyetoğlu et al., 2019b). The best performing model on average for the null and cross-context trained a BiLSTM with `fastText` (Joulin et al., 2017; Mikolov et al., 2018) embeddings on a multitask learning objective (Radford, 2019). Safaya (2019) attained the smallest score drop between null and cross-contexts using BiGRU and `word2vec`. Another study utilized ELMo with a fully connected multi-layer Neural Network, reaching comparable results (Maslennikova, 2019).

6.3. Sentiment Analysis

Sentiment analysis is a frequently studied classification task. MR and CR are a couple of exhaustively used data sets for this task. Successful models on this task involve combining `word2vec` with self-adaptive hierarchical sentence representations (Zhao et al., 2015); sentence representations that are learned by supervised training on a Natural Language Inference data (Conneau et al., 2017; Bowman et al., 2015); and a multi-channel system consisting of two bi-directional recurrent neural networks fed by tunable word vectors (Logeswaran and Lee, 2018).

7. Discussion

DistilBERT is better at utilizing longer sequences than ELMo. Fine-tuned ELMo cannot handle as many tokens as DistilBERT can, due to excessive RAM usage. This deteriorates ELMo's performance, especially in the cross-context. Moreover, fine-tuning causes training ELMo to take 1.5X longer, while the effect is negligible in DistilBERT.
Null context performance and cross-context performance do not necessarily grow together. For some specific configurations, when DistilBERT outran ELMo in the null context, ELMo happened to outperform DistilBERT in the cross-context or vice versa. Similarly, fine-tuning could improve null context performance, but caused a drop in the cross-context performance. Even usage of longer context can cause such an effect. These observations indicate that it is important to check the robustness of a model on multiple dimensions to understand true generalization power.
It should be emphasized that the limitations of the experimental setup and the scope must always be noted when the observations of this study are concerned. All conclusions are valid only under the specific experimental setup of this study, comprising the aforementioned binary classification tasks and the data sets. The results might be completely different, even in the case when the models are pretrained with

[9] `http://clef2019.clef-initiative.eu/`, accessed in December 2019.

other corpora. So it must be underlined that the comparison results are special to the model-unlabeled data combinations (ELMo combined with One Billion Word Benchmark, DistilBERT combined with English Wikipedia and Toronto BookCorpus).

8. Conclusion

In this study, ELMo and DistilBERT are compared on their fine-tuning performance on two binary text classification tasks. The main focus was to see how much can these models be benefited from in a practical way without any modification to the pretraining outputs.

Overall, DistilBERT is found to generalize better than ELMo on the cross-context setting. While DistilBERT and ELMo seem to have close performance in terms of absolute F-score, DistilBERT apparently outperforms ELMo in F-score drop in percentages. In addition, DistilBERT is 30% smaller in embedding size and 83% faster in training time than ELMo. No significant difference could be detected between ELMo and DistilBERT in the null context. The baselines are outran by both models in the cross-context robustness. But the baselines could occasionally get comparable results with ELMo in the null context. Also, they are very economic with 99% smaller size and 96% faster training and testing time when compared to DistilBERT.

As a result, when the transfer power of a model is a priority, it is worth to prefer contextual neural models over traditional ML methods despite much longer training times and memory overhead. On the other hand, traditional ML methods might still be preferred as low-cost options when there is no anticipated discrepancy between training and test data.

9. Future Work

The main focus in this study was to compare ELMo and DistilBERT without any intervention to the pretrained models, although the models were actually pretrained on entirely different corpora - ELMo on One Billion Words Benchmark (Chelba et al., 2013), DistilBERT on English Wikipedia and Toronto BookCorpus (Zhu et al., 2015). If the models were also pretrained from scratch on the same corpus, it would be ensured that they utilize the same knowledge to learn the context. This would enable a fairer comparison.

By leveraging unsupervised data into training, classifiers could adapt to many different cross-context settings more effectively and much faster. Unsupervised domain adaptation seems to be a wise direction to take (Han and Eisenstein, 2019).

Currently NLP evaluation and comparison studies are realized under varying conditions defined by specific priorities and research interests of every other study, including this particular one. This prevents making proper comparisons between observations of studies, which could enable progress based on a much more confident common ground. Defining standard evaluation pipelines to be adopted within the NLP field in general can be a way to overcome this dilemma.

10. Acknowledgements

We are grateful to Koç University Emerging Markets Welfare research team, which is funded by the European Research Council (ERC) Starting Grant 714868 awarded to Dr. Erdem Yörük for their generosity in providing the data and sharing invaluable insight. We thank the Text Analytics and Bioinformatics (TABI) Lab members in Boğaziçi University for their inspiring feedback and support. The numerical calculations reported in this paper were partially performed at TUBITAK ULAKBIM, High Performance and Grid Computing Center (TRUBA resources). GEBIP Award of the Turkish Academy of Sciences (to A.O.) is gratefully acknowledged

11. Bibliographical References

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal, September. Association for Computational Linguistics.

Chelba, C., Mikolov, T., Schuster, M., Ge, Q., Brants, T., Koehn, P., and Robinson, T. (2013). One billion word benchmark for measuring progress in statistical language modeling. Technical report, Google.

Conneau, A. and Kiela, D. (2018). SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May. European Language Resources Association (ELRA).

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark, September. Association for Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). 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)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Ettinger, A., Rao, S., Daumé III, H., and Bender, E. M. (2017). Towards linguistically generalizable NLP systems: A workshop and shared task. In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*, pages 1–10, Copenhagen, Denmark, September. Association for Computational Linguistics.

Greenwood, A. (1992). Computational neuroscience: A window to understanding how the brain works. In *Science at the Frontier*, chapter 9, pages 199–232. The National Academies Press, Washington, DC.

Han, X. and Eisenstein, J. (2019). Unsupervised domain adaptation of contextualized embeddings for sequence labeling. *arXiv:1907.11692*.

Hill, F., Cho, K., and Korhonen, A. (2016). Learning distributed representations of sentences from unlabelled data. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1367–1377, San Diego, California, June. Association for Computational Linguistics.

Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia, July. Association for Computational Linguistics.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA. ACM.

Hürriyetoğlu, A., Yörük, E., Yüret, D., Yoltar, Ç., Gürel, B., Duruşan, F., and Mutlu, O. (2019a). A task set proposal for automatic protest information collection across multiple countries. In Leif Azzopardi, et al., editors, *Advances in Information Retrieval*, pages 316–323, Cham. Springer International Publishing.

Hürriyetoğlu, A., Yörük, E., Yüret, D., Yoltar, Ç., Gürel, B., Duruşan, F., Mutlu, O., and Akdemir, A. (2019b). Overview of clef 2019 lab protestnews: Extracting protests from news in a cross-context setting. In Fabio Crestani, et al., editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 425–432, Cham. Springer International Publishing.

Hürriyetoğlu, A., Yörük, E., Yüret, D., Yörük, E., Yoltar, Ç., Gürel, B., Duruşan, F., Mutlu, O., Akdemir, A., Gessler, T., and Makarov, P., (2019c). *CLEF-2019 Lab ProtestNews on Extracting Protests from News*. accessed in December 2019.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2017). Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain, April. Association for Computational Linguistics.

Kell, A. J., Yamins, D. L., Shook, E. N., Norman-Haignere, S. V., and McDermott, J. H. (2018). A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron*, 98(3):630 – 644.e16.

King, G. and Lowe, W. (2003). An automated information extraction tool for international conflict data with performance as good as human coders: A rare events evaluation design. *International Organization*, 57:617–642, July.

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12.

Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Skip-thought vectors. In C. Cortes, et al., editors, *Advances in Neural Information Processing Systems 28*, pages 3294–3302. Curran Associates, Inc.

Liu, N. F., Gardner, M., Belinkov, Y., Peters, M. E., and Smith, N. A. (2019). Linguistic knowledge and transferability of contextual representations. 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)*, pages 1073–1094, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Logeswaran, L. and Lee, H. (2018). An efficient framework for learning sentence representations. In *International Conference on Learning Representations*.

Maslennikova, E. (2019). Elmo word representations for news protection. *In Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum*, 07.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, et al., editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Mikolov, T., Grave, E., Bojanowski, P., Puhrsch, C., and Joulin, A. (2018). Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Pang, B. and Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan, June. Association for Computational Linguistics.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June. Association for Computational Linguistics.

Peters, M. E., Ruder, S., and Smith, N. A. (2019). To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14, Florence, Italy, August. Association for Computational Linguistics.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language models are unsupervised multitask learners. Technical report, Open AI.

Radford, B. (2019). Multitask models for supervised protest detection in texts. *In Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum*, 07.

Safaya, A. (2019). Event sentence detection task using attention model. *In Working Notes of CLEF 2019 - Conference and Labs of the Evaluation Forum*, 07.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. (2019). Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS EMC^{2}Workshop*.

Sun, C., Qiu, X., Xu, Y., and Huang, X. (2019). How to fine-tune bert for text classification? In Maosong Sun, et al., editors, *Chinese Computational Linguistics*, pages 194–206, Cham. Springer International Publishing.

Tenney, I., Xia, P., Chen, B., Wang, A., Poliak, A., McCoy, R. T., Kim, N., Durme, B. V., Bowman, S. R., Das, D., and Pavlick, E. (2019). What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Łukasz Kaiser, Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Yeh, A. (2000). More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th Conference on Computational Linguistics - Volume 2*, COLING '00, pages 947–953, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zhao, H., Lu, Z., and Poupart, P. (2015). Self-adaptive hierarchical sentence model. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 4069–4076. AAAI Press.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *The IEEE International Conference on Computer Vision (ICCV)*, December.