Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 5579–5587
Marseille, 11–16 May 2020

Text Classification Using Language Modeling: Reproducing ULMFiT

Mohamed Abdellatif, Ahmed Elgammal
Rutgers University - Computer Science
Piscataway, NJ, USA
{mma215, elgammal}@cs.rutgers.edu

Abstract

In this paper, we reproduce some of the experiments of text classification by fine tuning pre-trained language model on the six English data-sets described in Howard and Ruder (2018) (verification). Then we investigate applicability of the model as is (pre-trained on English) by conducting additional experiments on three other non-English data-sets that are not in the original paper (extension). For the verification experiments, we didn't generate the exact same numbers as the original paper, however, the replication results are in the same range as compared to the baselines reported for comparison purposes. We attribute this to the limitation in computational resources which forced us to run on smaller batch sizes and for fewer number of epochs. Otherwise, we followed in the footsteps of the author to the best of our abilities (e.g. the libraries[1] , tutorials[2] , hyper-parameters and transfer learning methodology). We report implementation details as well as lessons learned in the appendices.

Keywords: language models, text classification, transfer learning

1. Introduction

Transfer learning can be defined as making use of the knowledge gained by solving a source problem T_s towards solving another (target) problem T_t. It is either transductive (the data of the target problem is unlabeled) or inductive (the data of the source problem is unlabeled). Definitions from Ruder et al. (2019). Inductive transfer learning has been well studied in Computer Vision e.g. Long et al. (2015) and Sharif Razavian et al. (2014). However, not as much in Natural Language Understanding/Processing. The problem of English text classification is motivated by practical applications like anomaly detection, security and legal applications. Solving the problem in other languages opens up avenues for these applications in the respective languages.

In Computer Vision, works to visualize the filters of a trained Convolutional Neural Network e.g. Mahendran and Vedaldi (2016), Samek et al. (2016) and Yosinski et al. (2014) show that the earlier layers (those closer to the input than the output) capture features that general across different datasets (e.g. edges, contours ..etc) while later layers (those closer to the output than the input) capture more dataset-specific features. A successful strategy for Computer Vision transfer learning has been transferring the earlier layers of a model that was trained on a general dataset to a model yet to be trained on a specific dataset Sharif Razavian et al. (2014). In Natural Language Processing, there has been a recent breakthrough inspired by transfer learning in Computer Vision. Howard and Ruder (2018) and Radford et al. (2018) are two notable works in this direction.

Applying transfer learning by fine-tuning a language model and a classifier significantly outperforms the state of the art results on six datasets bringing the test error down by 18-24% as compared to the baselines used in Howard and Ruder (2018). In this paper, we reproduce some of the results by Howard and Ruder (2018). To verify our replication is sound, we replicate results for the six English datasets citetd by them. Additionally, we extend by experimenting the model that is originally trained on English dataset on three non-English datasets covering the three sub-fields of text classification.

The rest of the paper is organized as follows: section 2. details the key points of the original paper. Section 3. describes our methodology and efforts to replicate as well as extend it. Section 4. describes the experiments parameters and settings. Section 5. presents both verification and extension results. Finally, we conclude the paper as well as point out future directions in sections 6. and 7. respectively. We supplement our replication efforts by publishing the code used online.

2. ULMFiT Description

2.1. Neural Network Architecture

Language model network The language model uses AWD LSTM Merity et al. (2017) which makes use of drop outs in a way to minimize disrupting RNN's ability to maintain long-term dependencies. Two language models are fine-tuned; a forward (regular) and a backward (where the data is read in reverse) one. Both encoders' are saved, reused for a forward and backward classifier later and prediction results are averaged.

Classifier network The encoder of the the (pre-trained and fine-tuned) language model is used plus two (untrained) linear layers. This is why when the classifier is fine-tuned, it is gradually unfrozen to allow for more aggressive training of the layers close to the output without eliminating the weights of the layers close to the input (the fine-tuned encoder of the language model). Instead of passing only the hidden state of the previous time step as an input to the first linear layer, ULMFiT passes it concatenated to pooled versions of as lengthy history of past hidden states as the GPU memory allows. They also modify BPTT, adding a block size parameter, to make training large documents manageable.

1. FastAI: https://www.fast.ai/
2. FastAI ULMFiT notebook: https://bit.ly/2vNACHi

2.2. Transfer Learning Methodology

Discriminative fine-tuning To train a neural network language model of multiple layers l, instead of solving one optimization problem using a single learning rate, they solve l optimization problems using l different learning rates. This is motivated by the fact that different layers capture different types of information Yosinski et al. (2014).

Gradual unfreezing An empirical compromise is presented between too aggressive (causes catastrophic forgetting) and too cautious (causes over-fitting and slow convergence) fine tuning. The differentiated learning rate allows for more aggressive (high values of) learning rates for layers closer to the output, since they capture more data-specific features. It also allows for more cautious (lower values of) learning rates of the layers closer to the input, since they capture more general features of the data. Gradual unfreezing of the network (bottom up) is applied and empirical values for learning rates are presented.

Slanted triangular learning rates Increasing the learning rate from a small value to a large one then back to the small value again before it is annihilated is shown to act as regularization Smith (2017). The transition from high values of learning rates to lower ones as training time progresses acts as escaping bumps of the loss function and slowing down (decreasing learning rate) as flatter areas are found. Increasing the chances of quick (higher learning rates at the beginning) synchronization between reaching smooth area of the loss function and decreasing the learning rate allows for more accuracy and faster convergence Smith and Topin (2019). The same concept is applied in Howard and Ruder (2018) with the modifications of shorter increase and longer decrease period.

3. Replication Methodology

The original paper presents main experiments on six data-sets spanning sentiment analysis, question and topic classification subcategories. In addition, it presents ablation experiments on three representative data-sets (one from each sub-category).

We present replication of the main experiments on nine data-sets (the six English ones plus three non English). Additionally, we replicate only two of the ablation experiments (the effect of pre-training and bi-directionality) on the same three representatives of the categories in both the English and the non-English datasets. We report the replication results (below the horizontal dashed line) as well as those of ULMFiT and its baselines (above the horizontal dashed line).

The extension experiments are meant to investigate the effect of applying ULMFiT as is (with input tokenization, numericalization and model pre-training based on English) on non English languages. We chose a non English dataset for each one of the sub categories originally tested by ULMFiT (sentiment analysis, question classification and topic classification). The choice of the non-English languages is explained later this in this section. We also report the results of applying the replicated version of ULMFiT on them as well as the previous work done on those datasets.

3.1. 11st Products Reviews (in Korean)

Motivated by experimenting ULMFiT with a CJK language, we use the Korean dataset compiled by Zhang and LeCun (2017). They crawled the Korean online shopping website 11st.co.kr [1] retrieving users' reviews (consists of a score that can be 1 up to 5 stars associated with text description of the users' feedback).

As CJK text may span thousands of letters, it imposes its own challenges building up a representation suitable for language models and classifiers to work with. Zhang and LeCun (2017)'s answer to this problem was encoding on byte-level (byte) and randomizing the text so that they could use English alphabet (randomized). They tried different encodings, we report here the 1-hot as it is the closest to the representation that we did. The classifier they used has an encoder that consists of 4-convolutional layers with input size n. They experiment with two versions: large input size (n=2048) and small input size(n=1944). We report their results for 1-hot encoding (byte large, byte small, randomized large and randomized small).

We consider it as a sentiment analysis (like yelp full in the verification experiments) and report validation error rate.

3.2. Russian Portal News Articles (in Arabic)

Considering number of native Arabic speakers who use the Internet versus the available Arabic content, there is a gap Allagui (2009). This motivates working towards better and more Arabic content online as well as research in Arabic. Arabic is technically different from English in the sense that it is written and read right to left. However, the total length of alphabet is close which makes the vocabulary size as well as pre-processing manageable. The dataset is a group of news articles in 40 topics collected from the Russian news agency RTA portal. The dataset has 23k articles, most of which (85.5%) have a single label (13% of the 23k articles have two labels while the remaining 1.5% has more than two).

The problem is approached by Al-Salemi et al. (2019) as a multi-label classification that is first transformed to a single label problem (step 1) then approached as a single label classification problem (step 2). They tried different algorithms (and yet combinations) for the two steps. We report some of the algorithms they experimented. From the common transformation based multi lable approaches they tried four (binary relevance, classifier chain, calibrated ranking by pairwise comparison and label powerset). For the classification algorithms, they tried with KNN (k nearest neighbors), SVM (support vector machines) and RF (random forest). We report four of the combination they tried including the one that, reportedly, achieved the highest F_1 score (LP-SVM).

We consider it as topic classification (like AG news in the verification experiments) and report (both macro and micro) F_1 scores as well.

3.3. DISEQuA (in Spanish)

News documents written in Spanish (spanning different topics and belonging to open domain text) were first col-

[1] https://www.w11st.co.kr

lected by Magnini et al. (2003), then keywords were extracted and finally keywords were converted to questions. Each question belongs to one of 6 possible categories.
Solorio et al. (2004) applies SVM on 4 different features (results of which are reported in table 5). They form the *Internet* features by picking representative words, use them to form classes-related queries which in turn are used to query Internet-based search engines. *Words* features stand for a simple bag-of-words. For *Prefix*, they use prefixes of lengths 4 and 5 since the average length of a word in Spanish is 4.75 (Solorio et al. (2004)).
We treat the problem as a question classification (like TREC-6 in the verification experiments), apply replicated ULMFiT and present the test error rate. We report the results by Solorio et al. (2004) as they were concerned with non-English question classification on the same data-set.

4. Experiments

4.1. Datasets and Tasks

The choice of the datasets in ULMFiT seems to follow the SotA text classification and transfer learning back then Johnson and Zhang (2017) and McCann et al. (2017).
We present results of replicated ULMFiT (RULMFit) on the six datasets presented in ULMFiT belonging to the three sub-topics of sentiment analysis, question classification and topic classification (verification experiments). Additionally, we experiment on three non-English datasets, spanning the three categories (extension experiments). Table 1 lists all the datasets we used.
Sentiment analysis For verification, we use the binary movie review IMDB Maas et al. (2011), the binary and the five-class version of the Yelp review dataset compiled by Zhang et al. (2015). For extension, we use 11st Korean dataset compiled by Zhang and LeCun (2017).
Question classification For verification, we use the six-class version of the small TREC dataset Voorhees and Tice (1999) dataset of open-domain, fact-based questions divided into broad semantic categories. For extension, we use the Spanish part of the multilingual question answering DISEQuA Magnini et al. (2003)
Topic classification For verification, we use the large-scale AG news and DBpedia ontology datasets created by Zhang et al. (2015). For extension, we use RTA news Arabic dataset introduced by Al-Salemi et al. (2019).

4.2. Experimental Setup

Pre-processing For the English datasets, we follow the pre-processing of Johnson and Zhang (2017) and McCann et al. (2017). For the non-English datasets, we follow the pre-processing of Zhang and LeCun (2017) for 11st Korean, Al-Salemi et al. (2019) for RTA news Arabic and Magnini et al. (2003) for DISEQuA Spanish.
Training Our goal is to replicate the environment of ULMFiT to the best of our computational abilities. To that end, for each one of the six English datasets of the verification experiments, we use a 10% of the training set as a validation set to adjust the hyperparameters, a batch size of 128 for the language model and 64 for the classifier, and a pre-trained model on WikiText dataset Merity et al. (2016). We use LSTM-AWD Merity et al. (2017) for both the language model and the classifier.
Both the language model and classifier LSTM-AWD have 3 layers, 1152 hidden activations per layer, embedding size of 400, bptt batch size of 80. The LSTM-AWD of the language model has dropouts of 0.02, 0.25, 0.2, 0.15 while that of the classifier has dropouts of 0.05, 0.4, 0.5, 0.3 for embedding, input, weight and hidden respectively. We used Adam optimizer with weight decay of 0.01. We use gradually decreasing learning rate starting at 0.02 and 0.1 for the language model and the classifier respectively. We train using a TITAN X (Pascal) with 12 GB of memory. This governed the batch size and number of epochs. Appendix A details the process of setting (as well as the final values used for) them for each one of the verification experiments.

5. Results

To evaluate the verification experiments, following Howard and Ruder (2018) we report error rate for the six English data sets. To evaluate the extension experiments, following the baselines (Al-Salemi et al. (2019), Magnini et al. (2003) and Zhang and LeCun (2017)) we report error rates for DISEQuA (Spanish) and 11st product review full (Korean) while we report F_1 (both micro and macro) of Russian portal news article (Arabic). We do 10-folds cross validation of only DISEQuA (Spanish). Furthermore, for the main verification experiments, we use the same splits of the six English datasets by the baselines, use 10% of the training portion as validation to train the models and blind-test on the test portion. For the other (ablation and extension) we report validation results.
Tables 2 and 3 show the verification experiments results while table 5 show the extension experiments results.
In addition to experiments of RULMFit (R for replicated), we also conduct experiments and report results for RULMFit-U (unidirectional language model and classifier) to study the value of using bidirectional language model and classifier and RULMFiT-NPT (for not pre-trained) to study the effect of pre-training. We conducted the two ablation experiments on three data-sets TREC6, IMDB and AG news representing the three sub-fields question classification, sentiment analysis and topic classification respectively. Table 4 presents the results. We do similar ablation analysis of the three of the extension data-sets and report results in table 6.

5.1. Verification

5.1.1. Main Experiments

Table 2 shows replication results of IMDB and TREC-6 along with original ULMFiT Howard and Ruder (2018) and models used by McCann et al. (2017). Table 3 shows the test error rates of the larger AG, DBPedia, Yelp-bi and Yelp-full used by Johnson and Zhang (2017).
Replication results shown in tables 2 and 3 do not exactly match ULMFiT's but they are in a close range relative to the baseline models. We attribute the slight differences to the different hardware, batch sizes (and hence learning rate) and number of epochs of RULMFiT (compared to ULMFiT). However, *all the test error rate values are smaller*

Dataset	Language	Type	# classes	# examples
TREC-6	English	Question	6	5.5k
IMDB	English	Sentiment	2	25k
Yelp (binary)	English	Sentiment	2	560k
Yelp (full)	English	Sentiment	5	650k
AG news	English	Topic	4	120k
DBPedia	English	Topic	14	560k
Russian news	Arabic	Topic	40	23k
11st	Korean	Sentiment	5	850k
DiseQuA	Spanish	Question	7	450

Table 1: Text classification datasets and tasks with number of classes and training examples

	Model	Test		Model	Test
IMDB	CoVE McCann et al. (2017)	8.2	TREC-6	CoVE McCann et al. (2017)	4.2
	oh-LSMT Johnson and Zhang (2016)	5.9		TBCNN Mou et al. (2015)	4.0
	Virtual Miyato et al. (2016)	5.9		LSTM-CNN Zhou et al. (2016)	3.9
	ULMFiT Howard and Ruder (2018)	4.6		ULMFiT Howard and Ruder (2018)	3.6
	RULMFiT	4.71		RULMFiT	3.2

Table 2: Test error rates (%) on two text classification datasets used by McCann et al. (2017)

than these by ULMFiT's baselines which indicates **the superiority of the replicated work over the baselines.**

5.1.2. Ablation Analysis

Table 4 shows validation error rates of the three representative datasets. Observing the table, *not pre-training gives higher error rates for all the three datasets* which implies that **pre-training indeed helps with the final classification accuracy**. However, for TREC6 (the smallest dataset) the impact is bigger than IMDB and AG news. This supports the claim that **the bigger the dataset the more knowledge is gained by fine-tuning the language model and less beneficial the pre-training is.**

Comparing bi-directional experiments results against the uni-directional (as shown in table 4), *bidirectionality gave lower error rates than a uni-directional model for the three datasets* which implies that, **from final classification accuracy stand point, using bi-directional model is better than using a uni-directional one. However, it comes at the cost of training and using an additional model**.

5.2. Extension

5.2.1. Main Experiments

As shown in table 5, results of the non-English languages are either in the range or worse than the baselines with varying degrees of differences for different languages. Spanish, being close to English, seems to have made use of the English pre-trained model and achieved an error rate that is en par with the baseline (SVM-Internet). On the other hand, Korean, being the farthest away from English, seems to make the least benefit of the English pre-trained model. Finally, in terms of closeness to the baseline, Arabic came in between Spanish and Korean. Arabic is different form English in the sense that it is handled right-to-left but in terms of vocab size, number of unique characters and morphology Giaber (2017).

5.2.2. Ablation Analysis

As shown in table 6 non pre-training experiments show worse results for all the datasets which implies that all the non-English datasets made use of the English pre-trained model. The Korean dataset's RULMFiT-NPT is the closest to RULMFiT which implies that it made the least use of English pre-trained model compared to the Spanish and Arabic ones. Using the bi-directional ULMFiT doesn't show a clear pattern of superiority over uni-directional one on non-English datasets (since RULMFiT is not better than RULMFiT-U).

5.3. Discussion

The verification part concludes that we have a model that is close to the original ULMFiT from point of view of bidirectionality, pre-training and trends in the six datasets.

Applying RULMFiT on the non-English datasets, results show potential of the technique as well as available rooms of improvements in terms of classification accuracy.

Factors playing roles in the results: the type of the subcategory of classification (sentiment analysis, topic and question classification), the sizes of the datasets (specially the training data) and finally, the similarity between the language in question and English. The latter matters because ULMFiT is pre-trained on English and uses tokenization, numericalization and representation that are based on English vocab size, number of unique characters and morphology.

To be able to draw a better conclusion of applying ULMFiT on non-English dataset would require more experiments. Due to the scope of this paper, we leave this for future work.

6. Conclusion

We replicated the results of Howard and Ruder (2018) which pushed the state of the art of English text generation

	AG	DBPedia	Yelp-bi	Yelp-full
Char-level CNN Zhang et al. (2015)	9.51	1.55	4.88	37.95
CNN Johnson and Zhang (2016)	6.57	0.84	2.90	32.39
DPCNN Johnson and Zhang (2017)	6.87	0.88	2.64	30.58
ULMFiT Howard and Ruder (2018)	5.01	0.80	2.16	29.98
RULMFiT	5.79	0.71	2.18	29.14

Table 3: Test error rates (%) on text classification datasets used by Johnson and Zhang (2017)

Experiment	TREC6	AG News	IMDB
RULMFiT-U	3.8	6.05	5.004
RUMLFiT	3.2	6.0	4.48
RULMFiT-NPT	7.8	6.38	4.94

Table 4: Validation error rates of the representative datasets for the ablation experiments

on the six datasets compared to the baselines. The original paper used transfer learning from a source problem of language modeling to a target problem of text classification where they first fine tune a pre-trained language model on a general corpus of text, save its encoder, load it to a classifier and fine tune it as well. By using discriminitive fine tuning, slanted learning rates and graduate unfreezing the method was successful achieving the objective. We followed the same settings (libraries, hyper-parameters, methodology) of the paper to the best of our computational abilities.

Given the definition of Cohen et al. (2018) of repeatability and reproducibility, we found experiments whose results are described in tables 2 and 3 of Howard and Ruder (2018) to be both repeatable and reproducible. Even though not all the parameters where detailed in the original paper and could not be found on Fast AI tutorials online, following the practices of Machine Learning, default values and practices mentioned by Howard & Ruder rendered the experiments both repeatable and reproducible. The three attached appendices shed more light on the reproduction attempt.

7. Future Directions

The transfer learning method builds upon a language model that is pre-trained on a specific data-set. Since the performance is impacted by the pre-training data-set and data representation steps (tokenization, numericalization and building the vocabularies) they are avenues for further investigation (specially for non-English).

Other transfer learning works are also worth replication. Radford et al (Radford et al. (2018)) showed transfer learning pushed the state of the art of several NLP tasks including text classification using Transformer Vaswani et al. (2017) as their model. Transformers were pre-trained in two directions as one model (BERT) by Devlin et al. (2018). Finally, the idea is extended by Lample and Conneau (2019) to pre-train language models across different languages (XLMs)

Appendix A: Lessons Learned

Since this work is an effort to reproduce the work by Howard and Ruder (2018), we present lessons learned from the reproduction attempts here.

As viewed by Cohen et al. (2018) **replicability (repeatability)** of an experiment is different from its **reprodcability**. The former being a property of the experiment (whether it is doable (or to what extent it is easy) to follow a set of steps/procedures to carry out the experiment again) while the later is more about the outcome of a replicable/repeatable experiment (assuming an experiment is already replicable/repeatable, whether the produced outcomes of the replicated/repeated experiment matches (or to what extent it does) the outcomes reported by the original experiment).

Resources that were necessary to be able to repeat experiments whose results are shown in tables 2 and 3 of Howard and Ruder (2018) (**repeatability** lessons): hardware resources, appropriate environment setup, programming skills and allocating time to run the experiments and following up on the long running ones. Among lessons that we learned to achieve **reproducibility**: careful alignment with proper splits, using blind testing, following the default values mentioned by the authors (e.g. default of 15 epochs for the Language Model # epochs) and early stoppage of training based on validation accuracy. In addition to helping with reproducibility, these practices help with cutting on training time and minimizing the chances of having to re-do a failed experiment that is time consuming.

Appendix B: Replicability Details

7.1. Challenges

Computational time and space usage are challenging to repeat these experiments. This is imposed by the relatively large sizes of the data-sets under question as well as having to fine-tune two different neural networks for each of the six data-sets (the language model and the classifier). Table 7 shows approximate sizes and running time of RULMFiT on the data-sets.

7.2. Operation Environment

Hardware-wise we used a server with Intel(R) Xeon(R) CPU (3.20GHz and 64 bits), 250 GB of memory and Nvidia Titan X GPU with 12 GB of memory. **Software**-wise, we used Ubuntu 18.04 system (with Linux Kernel version

	Model	Micro F_1	Macro F_1		Model	Error %		Model	Error %
Arabic	BR-RF	67.56	62.23	Spanish	SVM-Internet	31.14	Korean	Byte-large	32.56
	BR-SVM	69.89	65.88		SVM-Prefix4	23.03		Byte-small	32.43
	LP-RF	65.28	57.8		SVM-Prefix5	18.55		Randomized-large	32.73
	LP-SVM	73.04	69.79		SVM-Words	20.1		Randomized-small	32.69
	RULMFiT	62.0	54.0		RULMFiT	28.27		RULMFiT	50.62

Table 5: Validation results of non English data sets

Experiment	Arabic		Spanish	Korean
	Micro F_1	Macro F_1	Error %	Error %
RULMFiT-U	67.0	54.0	27.78	50.62
RUMLFiT	62.0	54.0	28.27	50.62
RULMFiT-NPT	51.0	45.0	31.544	50.85

Table 6: Ablation experiments validation results on the non-English data sets

4.15.0), Python 3.7 and utilized the following Python libraries: FastAI 1.0.60, sklearn 0.22.1 (particularly to calculate evaluation metrics) and numpy 1.18.0.

7.3. Hyper-parameters

A lot of parameters play factors towards the ability to replicate (and hence reproduce) results of Howard and Ruder (2018) as discussed in section 4.2.. Recall that the work under study has two main components: a language model and a classifier (section 2.). We discuss here the two factors: number of epochs and batch size. Tables 8 and 9 list the number of epochs and batch sizes used for the verification experiments for both the language models and the classifiers respectively. Out of our experience it was not always best to keep increasing the **number of epochs** (otherwise over-fitting will kick in). For big datasets (e.g. yelp full and yelp binary) we save the language model after every epoch, run for 15 epochs (the default reported by Howard and Ruder (2018)) and pick the one that gave the lowest validation loss). Howard and Ruder (2018) mentioned they fine tuned the language models with early stoppage. For the **batch size**, we would always try out the maximum possible value (to process the maximum amount of data at a given iteration and reduce training time Smith et al. (2017)), but Cuda runtime error won't allow it sometimes. In these cases, we decrease the batch size and make the experiment run take more time. It is a matter of hardware limitation.

Appendix C: Reproducibility Details

Given that an experiment is replicable/repeatable, authors of Cohen et al. (2018) define three dimensions of reprodcability: **value** (a numeric value (e.g. error rate, $F_!$, accuracy ..etc)), **finding** (a result of comparing two or more dependent variables on reproduced values (e.g. test error rate of certain classification algorithm is lower than that of another one on a certain data-set)) and **conclusion** (a more general induction made based on several reproduced findings that consistently agree on a pattern). All the instances of reproducability dimensions of the verification experiments as well as the verification ablation analysis are highlighted in text (sections 5.: values are underlined, *findings are italic* and **conclusions are bald**). Besides, we calculate the percentages of relative differences between the replicated results (RULMFiT) and the original work (ULMFiT) in figure 1. We did the same thing for ULMFiT's baselines and included it in the same chart.

8. Bibliographical References

References

B. Al-Salemi, M. Ayob, G. Kendall, and S. A. M. Noah. Multi-label arabic text categorization: A benchmark and baseline comparison of multi-label learning algorithms. *Information Processing & Management*, 56(1):212–227, 2019.

I. Allagui. Multiple mirrors of the arab digital gap. *Global Media Journal*, 8(14):N_A, 2009.

K. B. Cohen, J. Xia, P. Zweigenbaum, T. J. Callahan, O. Hargraves, F. Goss, N. Ide, A. Névéol, C. Grouin, and L. E. Hunter. Three dimensions of reproducibility in natural language processing. In *LREC... International Conference on Language Resources & Evaluation:[proceedings]. International Conference on Language Resources and Evaluation*, volume 2018, page 156. NIH Public Access, 2018.

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

J. M. Giaber. Differences in word formation between arabic and english: Implications for concision in terminology translation. *Al-Arabiyya*, pages 53–79, 2017.

J. Howard and S. Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

R. Johnson and T. Zhang. Supervised and semi-supervised text categorization using lstm for region embeddings. *arXiv preprint arXiv:1602.02373*, 2016.

Experiment	Data set	Size	Running time
RULMFiT and RULMFiT-U	Yelp full	524 MB	2 days
RULMFiT and RULMFiT-U	Yelp binary	437 MB	2 days
RULMFiT and RULMFiT-U	DBPedia	196 MB	1 day
RULMFiT and RULMFiT-U	IMDB	218 MB	7 hours
RULMFiT and RULMFiT-U	AGNews	32 MB	3 hours
RULMFiT and RULMFiT-U	TREC6	360 KB	10 min
RULMFiT-NPT	IMDB	218 MB	7 hours
RULMFiT-NPT	AGNews	32 MB	3 hours
RULMFiT-NPT	TREC6	360 KB	10 min

Table 7: Approximate sizes of data sets and running times of verification experiments

Experiment	Data set	Forward	Backward	Forward	Backward
		# epochs		batch size	
RULMFiT and RULMFiT-U	Yelp full	15	15	100	100
RULMFiT and RULMFiT-U	Yelp binary	15	19	100	100
RULMFiT and RULMFiT-U	DBPedia	15	15	128	128
RULMFiT and RULMFiT-U	IMDB	15	15	128	128
RULMFiT and RULMFiT-U	AGNews	15	15	128	128
RULMFiT and RULMFiT-U	TREC6	15	15	64	64
RULMFiT-NPT	IMDB	15	15	128	128
RULMFiT-NPT	AGNews	15	15	128	128
RULMFiT-NPT	TREC6	15	15	64	64

Table 8: Language model number of epochs and batch sizes of verification experiments

R. Johnson and T. Zhang. Deep pyramid convolutional neural networks for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 562–570, 2017.

G. Lample and A. Conneau. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*, 2019.

M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015.

A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 142–150. Association for Computational Linguistics, 2011.

B. Magnini, S. Romagnoli, A. Vallin, J. Herrera, A. Peñas, V. Peinado, F. Verdejo, and M. de Rijke. Creating the disequa corpus: a test set for multilingual question answering. In *Workshop of the Cross-Language Evaluation Forum for European Languages*, pages 487–500. Springer, 2003.

A. Mahendran and A. Vedaldi. Visualizing deep convolutional neural networks using natural pre-images. *International Journal of Computer Vision*, 120(3):233–255, 2016.

B. McCann, J. Bradbury, C. Xiong, and R. Socher. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6294–6305, 2017.

S. Merity, C. Xiong, J. Bradbury, and R. Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

S. Merity, N. S. Keskar, and R. Socher. Regularizing and optimizing lstm language models. *arXiv preprint arXiv:1708.02182*, 2017.

T. Miyato, A. M. Dai, and I. Goodfellow. Adversarial training methods for semi-supervised text classification. *arXiv preprint arXiv:1605.07725*, 2016.

L. Mou, H. Peng, G. Li, Y. Xu, L. Zhang, and Z. Jin. Discriminative neural sentence modeling by tree-based convolution. *arXiv preprint arXiv:1504.01106*, 2015.

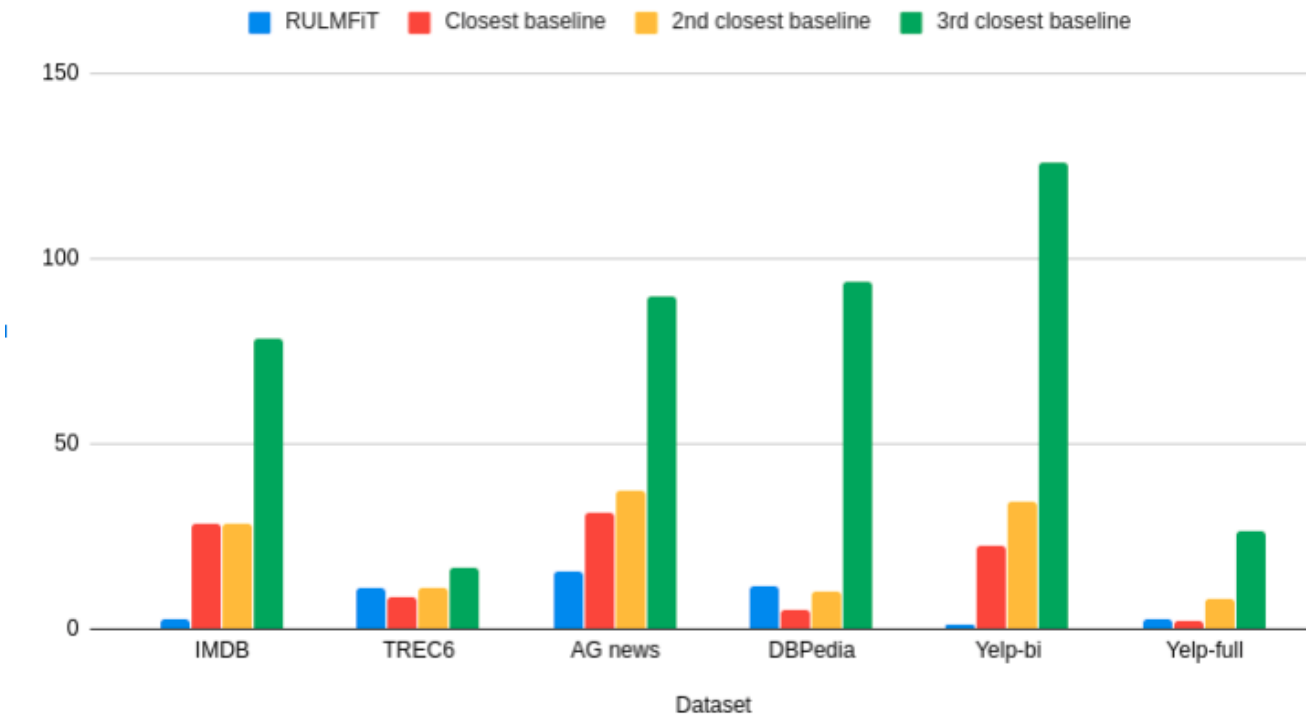
A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

S. Ruder, M. E. Peters, S. Swayamdipta, and T. Wolf. Transfer learning in natural language processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18, 2019.

Experiment	Data set	Forward	Backward	Forward	Backward
		# epochs		batch size	
RULMFiT and RULMFiT-U	Yelp full	2	2	48	48
RULMFiT and RULMFiT-U	Yelp binary	2	2	48	48
RULMFiT and RULMFiT-U	DBPedia	3	3	48	48
RULMFiT and RULMFiT-U	IMDB	5	5	32	32
RULMFiT and RULMFiT-U	AGNews	5	5	64	64
RULMFiT and RULMFiT-U	TREC6	50	50	64	64
RULMFiT-NPT	IMDB	5	5	32	32
RULMFiT-NPT	AGNews	5	5	64	64
RULMFiT-NPT	TREC6	50	50	64	64

Table 9: Classifier number of epochs and batch sizes of verification experiments

Figure 1: Relative differences (%) off Howard and Ruder (2018)



W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller. Evaluating the visualization of what a deep neural network has learned. *IEEE transactions on neural networks and learning systems*, 28(11):2660–2673, 2016.

A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014.

L. N. Smith. Cyclical learning rates for training neural networks. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 464–472. IEEE, 2017.

L. N. Smith and N. Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, volume 11006, page 1100612. International Society for Optics and Photonics, 2019.

S. L. Smith, P.-J. Kindermans, C. Ying, and Q. V. Le. Don't decay the learning rate, increase the batch size. *arXiv preprint arXiv:1711.00489*, 2017.

T. Solorio, M. Pérez-Coutiño, M. Montes-y Gémez, L. Villasenor-Pineda, and A. López-López. A language independent method for question classification. In *Proceedings of the 20th international conference on Computational Linguistics*, page 1374. Association for Computational Linguistics, 2004.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is

all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

E. M. Voorhees and D. M. Tice. The trec-8 question answering track evaluation. In *TREC*, volume 1999, page 82. Citeseer, 1999.

J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.

X. Zhang and Y. LeCun. Which encoding is the best for text classification in chinese, english, japanese and korean? *arXiv preprint arXiv:1708.02657*, 2017.

X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.

P. Zhou, Z. Qi, S. Zheng, J. Xu, H. Bao, and B. Xu. Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. *arXiv preprint arXiv:1611.06639*, 2016.