Accenture at CheckThat! 2020: If you say so: Post-hoc fact-checking of claims using transformer-based models

Evan Williams[1][0000-0002-0534-9450], Paul Rodrigues[1 2][0000-0002-2151-636X], and Valerie Novak[2][0000-0001-8317-0993]

[1] Accenture, 800 N. Glebe Rd., Arlington, 22209, USA
e.m.williams@accenture.com
paul.rodrigues@accenture.com
[2] University of Maryland, College Park, MD, USA
[3] vnovak@umd.edu

Abstract. We introduce the strategies used by the Accenture Team for the CLEF2020 CheckThat! Lab, Task 1, on English and Arabic. This shared task evaluated whether a claim in social media text should be professionally fact checked. To a journalist, a statement presented as fact, which would be of interest to a large audience, requires professional fact-checking before dissemination. We utilized BERT and RoBERTa models to identify claims in social media text a professional fact-checker should review, and rank these in priority order for the fact-checker. For the English challenge, we fine-tuned a RoBERTa model and added an extra mean pooling layer and a dropout layer to enhance generalizability to unseen text. For the Arabic task, we fine-tuned Arabic-language BERT models and demonstrate the use of back-translation to amplify the minority class and balance the dataset. The work presented here was scored 1st place in the English track, and 1st, 2nd, 3rd, and 4th place in the Arabic track.

Keywords: fact checking, fact identification, Arabic, BERT, RoBERTa

1 Introduction

Natural Language Processing (NLP) has been driving Artificial Intelligence research since the 1950s, but recently increased in distinction due to the quantity of text that can be utilized as well as new techniques to extract even more value from text. In 2018, a surge of research produced deep learning architectures in NLP which beat state of the art on a multitude of tasks, such as sentiment analysis, question answering, and semantic similarity, in a variety of languages.

Since the innovation of ULMFit [12], numerous new architectures have been introduced, such as ELMo [17], BERT [9], ERNIE [26], RoBERTa [14], GPT-2 [18], GPT-3 [6], and others, yielding breakthrough innovations and increased performance, nearly month after month. These architectures require massive amounts of training data, which can be expensive to train on high-performance computing clusters [25]. However, they facilitate the practice of transfer learning. A base model trained on a large amount of general text data can then be fine-tuned, or customized for a specific problem and domain/genre, using text with far less annotated data than previous systems required. This use of transfer learning allows us to effectively craft custom cutting-edge models to solve a wide range of classification problems.

While these architectures are often utilized to improve NLP tasks, the application of transformer-based transfer learning approaches are less often demonstrated as components in decision-support systems which aid the workflow of subject matter experts. We do see these technologies being used in the medical field (e.g. [20]), and anticipate there will be many more applications coming. The CheckThat! Lab poses one such application, which could reduce information burden in the workflow of a journalist.

1.1 CheckThat! Lab

We participated in Task 1 of the 2020 CheckThat! challenge. [5] Organizers distributed collections of tweets in English and in Arabic for training, annotated for topic group, whether the tweet was a claim, and whether the tweet was check-worthy, along with Twitter provided meta-data. [24, 10] Participants in the challenge utilized this data to train a model that could receive a list of novel tweets, classify each for check-worthiness, and rank the group of tweets by how check-worthy they were. Evaluation of the model was performed on a second test dataset provided for each language. These test datasets were held back by the organizers until shortly before the competition end time. Organizers provided this dataset unlabeled, and participants provided the labels and ranking to the organizers. Organizers evaluated the ranking produced by participating groups to a withheld labeled and ranked list. Participants were permitted to submit one primary run and up to 3 contrasting runs. The official metric for Arabic was Precision @ 30 (P@30). Precision @ k is the number of relevant results in the top k claims in the ranked list. The official metric for English was Mean Average Precision (mAP), or the mean of the average precision scores for each of the claims.

Provided Data Tweets were collected by CheckThat! organizers using keyword watchlists, consisting of usernames, hashtags, or key words, designed around a variety of topic areas.

For English, one topic was provided related to COVID-19, and filtered for tweets that mentioned #COVID19, #CoronavirusOutbreak, #Coronavirus, #Corona, #CoronaAlert, #CoronaOutbreak, corona, and COVID-19. This topic was the same in train, test, and the evaluation set.

For Arabic, the training data included three topics–Protests in Lebanon, Emirati cleric Wassim Youssef, as well as Turkey’s intervention in Syria. Testing data included topics such as Deal of the Century, The Houthis in Yemen, COVID-19, Feminists, Events in Libya, The group of resident non-citizens in Kuwait, Algeria, as well as Boycotting Countries & Promoting Rumors against Qatar. We note that the topics provided between train and test datasets differ, with no overlap.

The topic word lists were used by the organizers to collect posts on Twitter. Annotators were presented these posts and were asked to evaluate each for check-worthiness. Check-worthiness was defined as “a tweet that includes a claim that is of interest to a large audience (especially journalists), might have a harmful effect, etc.” [8] Tweets were assigned check-worthiness labels after review by two annotators as well as a review by a third expert annotator. Check-worthiness was evaluated on the following three criteria [4]:

- Do you think the claim in the tweet is of interest to the public?
- To what extent do you think the claim can negatively affect the reputation of an entity, country, etc.?
- Do you think journalists will be interested in covering the spread of the claim or the information discussed by the claim?

In examining the labeled training data, we confirmed nuanced differences between tweets that were check-worthy and tweets that were not. For example, the tweet below, which was taken from the English task development data, initially appears to be peddling a false COVID-19 claim. However, the rest of the tweet makes it clear that the author is joking, which is presumably why this tweet was not labeled as being check-worthy.

> ”ALERT The corona virus can be spread through money. If you have any money at home, put on some gloves, put all the money in to a plastic bag and put it outside the front door tonight. I’m collecting all the plastic bags tonight for safety. Think of your health.”

In contrast, the tweet below, which was labeled check-worthy, is spreading harmful COVID-19 misinformation which could dissuade people from getting tested.

> ”Coronavirus test in US is $3,000. Here in Tokyo it’s $50, $166 without State ins. In much of Europe it’s free Worse, in much of the US, it’s not even available, unreliable. And meanwhile #POTUS recently called Corona one big “hoax.” USA: 1st world $$$, 3rd world healthcare.”

We had concern that nuanced text like this may be difficult to discriminate and rank accurately.

For a journalist, the task of identifying noteworthy claims for the vetting process may be intuitive. Their knowledge of the material, background in academic training, and experience as a journalist inform their processes and decision-making. Our learner is not coached, trained, or experienced in this area beforehand. It receives the data and annotations provided by the annotators and learns the patterns of language to replicate their decision process.

2 Transformer Architectures and Pre-trained Models

2.1 BERT

Bidirectional Encoder Representations from Transformers (BERT) models have fundamentally changed the NLP landscape. The original BERT model's architecture consists of 12 transformers stacked on top of one another with a hidden size of 768 and 12 self-attention heads. [9] BERT models are trained by performing unsupervised tasks, namely masked token prediction (Masked LM) and prediction of future sentences (Next Sentence Prediction) on massive amounts of data. BERT utilizes a WordPiece tokenization scheme. [22], and was trained on Wikipedia and the BooksCorpus [30]. At the time of release, BERT was state-of-the-art in 11 NLP tasks.

Since initial release, many pre-trained BERT neural networks have been released. These can be focused on new languages, or differ in size. They can be either smaller and more efficient, or larger and more comprehensive, than the original release [27]. Any of these pre-trained models could serve as a base model for fine-tuning to new datasets and new tasks.

2.2 RoBERTa

RoBERTa, developed by Liu et al. [14], is an derivative of BERT which introduced modifications to the training process. The primary modifications are the provision of more training data, increasing pre-training steps with bigger batches over more data, removing Next Sentence Prediction, training on longer sequences, and dynamically changing the masking pattern applied to the training data [14]. While RoBERTa also requires sub-word tokenization, RoBERTa uses a Byte-Pair Encoding (BPE) instead of WordPiece. [23] The base-roberta model was pre-trained on 160GB of text extracted from BookCorpus, English Wikipedia, CC-News, OpenWebText, and Stories (a subset of CommonCrawl Data) [14].

At the time of release, the RoBERTa architecture achieved state-of-the-art results on publicly available benchmark datasets such as GLUE [28], RACE [13], and SQuAD [19]. Like BERT, RoBERTa models come in a variety of sizes, and choosing a model requires a trade-off between computational efficiency and model size.

While some new architectures have been released which exceed RoBERTa's performance, RoBERTa remains an accessible framework and continues to be one of the most highly ranked architectures on the SuperGLUE leaderboard.[4]

2.3 AraBERT

AraBERT is an Arabic model developed by Wissam Antoun, Fady Baly, and Hazem Hajj at the American University of Beirut [3]. The *aubmindlab/arabert*

[4] https://super.gluebenchmark.com/leaderboard

series of models were pre-trained on Arabic documents retrieved from the web, as well as two publicly available corpora: the 1.5 billion word Arabic Corpus, and the 1 billion word Open Source International Arabic News Corpus (OSIAN). No token count was provided for the web scraped documents. [3].

2.4 ArabicBERT

ArabicBERT is an Arabic model developed by Ali Safaya, Moutasem Abdullatif, and Deniz Yuret KUIS of Koc University. [21] ArabicBERT was trained on Wikipedia, and the OSCAR corpus [16], which utilized web data from CommonCrawl. The corpus used to create the pre-trained model was, in total, 8.5 billion words.

3 Quantitative Analysis

3.1 Label Balance

The datasets for both the English and the Arabic Challenges were imbalanced. The English Task 1 datasets contained a development dataset of 150 tweets and a training dataset of 672 tweets containing 39% and 34% check-worthy tweets respectively. The Arabic Task 1 training dataset provided 1,500 labeled tweets, 458 of which (31%) were labeled check-worthy.

We will discuss provisions we make for the Arabic imbalance later in the paper.

3.2 Vocabulary Analysis

When utilizing pre-trained models, vocabulary used to create these models plays a critical role. The process of fine-tuning does not allow for the addition of additional vocabulary, so these systems fallback to subword units during tokenization. Because we were evaluating a corpus that contained emerging topics (such as COVID-19), and our pre-trained models were created at different points between 2018 and 2020, we wanted to understand what our pre-trained models contained. We hypothesized that the models with the greatest token overlap would perform the best.

English The token overlap between the English test dataset and RoBERTa's vocabulary file was roughly 850 tokens (54%), with RoBERTa containing about 50K items in its vocabulary. Many tokens missing from the RoBERTa vocabulary were related to the coronavirus topic, including several terms for COVID-19 as well as named entities, emoji, foreign languages in non-Latin script, misspellings and slang/internet chat language (LMAOOO). No analysis was performed on the BERT vocabulary file.

Arabic The three Arabic model vocabularies contained 64K WordPieces (*aubmindlab/bert-base-arabert*), 64K WordPieces (*aubmindlab/bert-base-arabertv01*) and 32K WordPieces (*asafaya/bert-base-arabic*). A rough tokenization and cleaning of the tweets in the test data set resulted in roughly 15K unique tokens. The overlap between the three Arabic model vocabulary and the Arabic test data set was roughly 8.5K tokens or 56% of the tokens in the test data (*aubmindlab/bert-base-arabertv01*), 5.5K tokens or 36% of the tokens in the test data (*asafaya/bert-base-arabic*) and 3.5K or 23% of the tokens in the the test data (*aubmindlab/bert-base-arabert*). Some categories of vocabulary found in the test data set, but missing from the top performing model, included English words or loan words in Arabic script, colloquial/slang, misspellings/missing spaces, named entities (names of people and places), emoji and tokens in Latin script. The *asafaya/bert-base-arabic* Arabic model vocabulary also included a lot of longer WordPieces that were unlikely to be found in data. Additionally, even though the test data set contained short vowels, none of the Arabic model vocabularies had any short vowels.

4 Approach and Results

The datasets provided for English and Arabic contained Twitter metadata fields, but we discard these. Our methodology only utilizes the message text of the Tweet as well as the check-worthy field containing a binary label where the positive class denoted a check-worthy claim.[5]

Competition rules required that tweets most likely to be check-worthy needed to appear at the top of each topic. To generate rankings, we took the positive and negative class scores, generated by a sequence classification head on top the pooled output of the neural network models (whether it be BERT, RoBERTa, AraBERT, or ArabicBERT), and passed those scores through a softmax function to normalize the classification outputs. We then subtracted the negative class probability from the positive class probability. This yielded interpretable, normalized scores between 1 and -1, where higher scores reflected our model's confidence that a tweet was check-worthy. We then sorted by the difference of probabilities to produce the ranked tweets submitted to the organizers of the conference.

4.1 English

Classification For our internal evaluations, we split the English training data provided into 80% training and 20% validation sets. We used the development set as was provided by the organizers.

We evaluated three baseline models. We fine-tuned the data over 2 epochs on the original English BERT model [9], a BERT model trained on COVID-19

[5] We tried concatenating the text field with the pre-labeled topicID field, but this did not improve the model's performance at all, so we chose to exclude topic labels from the model.

Twitter data [15], and the original English RoBERTa model [14]. We assumed that the COVID-19 Twitter model would generate the highest accuracy given its deep contextual knowledge of both Twitter data and COVID-19, but of the three models, RoBERTa generated the highest precision and recall for both the positive and negative class. We chose to eliminate the previous two models and focus on optimizing RoBERTa.[6]

In our internal evaluations, we noticed the model overfitting quickly. To help prevent this, we added an extra mean pooling layer and dropout layer to the model. Our pooling layer takes the weights from the last layer, which were overfitting, and averages them with weights from the second-to-last layer. This reduces overfitting by smoothing out some of the weights originally calculated in the final layer. Dropout is a regularization technique that reduces overfitting by randomly omitting (or zeroing out) hidden units from the network during each training step at a probability specified by the user [11]. By adding these two layers to the end of our RoBERTa model, we were able to improve accuracy on our test set and reduce overfitting.

After a grid search, we fine-tuned with 2 epochs, a batch size of 32, and Adam optimization with a learning rate of 1.5e-5. The RoBERTa model was fine-tuned using the Keras API to TensorFlow.

This output was then fed through a softmax function, and the difference between the positive and negative class likelihoods were used to rank tweets within each pre-labeled topic category.

Results Results of our fine-tuned RoBERTa model can be found in Table 1 as **RoBERTa**. This submission placed first place among all competing teams with a mAP of 0.8064. Our contribution narrowly beat out the second place results, which likely utilized a similar model. We did not submit our BERT model or COVID Twitter models for formal evaluation.

Table 1. Accenture results from CheckThat! Task1 English.

Entry	**mAP**	**RR**	**R-P**	**P@1**	**P@3**	**P@5**	**P@10**	**P@20**	**P@30**
RoBERTa	0.8064	1.0000	0.7167	1.0000	1.0000	1.0000	1.0000	0.9500	0.7400

4.2 Arabic

Classification For our internal evaluations, we split the Arabic training data provided into 70% training, 20% validation, and 10% held-out sets. We evaluated four baseline Arabic BERT models retrieved from Huggingface, without any parameter tuning. [29]. These models were *Hate-speech-CNERG/ dehatebert-mono-arabic* [2], *asafaya/bert-base-arabic* [21], *aubmindlab/bert-base-arabert* [3],

[6] In hindsight, these two should have been contributed for formal evaluation.

and *aubmindlab/bert-base-arabertv01* [3]. Out of four, we found three to have promise, *aubmindlab/bert-base-arabertv01*, *aubmindlab/bert-base-arabert*, and *asafaya/bert-base-arabic*.

Classes were imbalanced in the Arabic training dataset with 30% of tweets labeled as part of the check-worthy class. In order to address the imbalanced classes, we chose to upsample the positive class using machine translation via Amazon Web Services (AWS) Translate.

Tweets from the positive class in the training and development sets were translated to English and then back to Arabic (ar→en→ar), appended to our training dataset, and assigned a label of check-worthy. This improved both precision and recall for check-worthy tweets, but slightly harmed the precision and recall for tweets that were not check-worthy. As the goal is to surface and rank the positive class at various levels of precision, a reduction in the F1-score of the negative class was acceptable for improving the F1-score of the positive class.

After a grid search, our final models were fine-tuned with 2 epochs, a learning rate of 2e-05, Adam optimization, and a batch size of 32. We used a Huggingface BERT sequence classification function[29] and, like with English, added a linear layer on top of the pooled output.

This output was then fed through a softmax function, and the difference between the positive and negative class likelihoods were used to rank tweets within each pre-labeled topic category.

Results Results for our Arabic evaluations can be found in Table 2. Our official submission to the competition was **AraBERT v0.1 Upsampled** and was evaluated in 1st place with a P@30 of 0.7000. Our comparative models **AraBERT v1.0 Upsampled**[7], **AraBERT v0.1 Unmodified**, and **ArabicBERT-Base Upsampled** were evaluated in 2nd, 3rd, and 4th place with P@30 scores of .6750, .6694, and .6639 respectively.

The benefit of back-translation to upsample the minority class can be seen by comparing **AraBERT v0.1 Upsampled** (P@30 of 0.7000) with **AraBERT v0.1 Unmodified** (P@30 of of 0.6694). These were the same model architectures, with identical hyperparameters, but one had upsampled data, and the other did not.

Comments: Preprocessing Once we had Arabic model performance baselines, we experimented with various preprocessing techniques. We assumed that these steps would reduce noise and help the Arabic BERT models better map words to tokens in its vocabulary. We performed internal evaluations involving variations of removing diacritics, stopwords, urls, punctuation, and also of splitting

[7] This is a rapidly evolving area of NLP. At the time of the challenge, documentation was not yet published for AraBERT v1.0. We did not realize v1.0 required running Farasa [1] as a preprocessing step for tokenization before utilization. We expect an Upsampled v1.0 to beat an Upsampled v0.1 when utilizing the necessary Arabic segmenter.

Table 2. Accenture results from CheckThat! Task1 Arabic

Entry	**P@5**	**P@10**	**P@15**	**P@20**	**P@25**	**P@30**	**AP**
AraBERT v0.1 Upsampled	0.7333	0.7167	0.7167	0.6875	0.6933	0.7000	0.6232
AraBERT v1.0 Upsampled	0.6667	0.7417	0.7333	0.7125	0.6900	0.6750	0.5967
AraBERT v0.1 Unmodified	0.6833	0.7083	0.7111	0.6833	0.6833	0.6694	0.6035
ArabicBERT-Base Upsampled	0.6000	0.6917	0.6944	0.6833	0.6667	0.6639	0.5947

underscores. We tested each of these preprocessing functions alone, as well as in combination with other preprocessing functions. We saw no increase in precision or recall from these steps. In fact, many combinations of these functions brought down our overall accuracy. We ultimately chose to forego all preprocessing.

Comments: Machine Translation Back-translation provides the model with alternative ways to express similar concepts. This makes the model more robust to vocabulary not present in the training data.

We evaluated three strategies to augment the corpus using translation data.

- adding back-translated data (ar→en→ar)
- adding the English translation (ar→en)
- adding both the English and back-translated Arabic text (ar→en; ar→en→ar).

We found the back-translated Arabic (without English) (ar→en→ar) had the provided the largest increase in accuracy on our internal evaluations.

English was chosen as an intermediary language due solely to the fact that AWS has strong English NLP support. Future research may explore which intermediary language translations can offer the largest performance boosts. While we may have benefited from exploring intermediary language alternatives [8], we had to leave this for future work due to constraints in both time and budget.

We recognize that this translation approach resulted in label leakage into the hold-out and validation sets, resulting in overfitting on our internal evaluations. However by expanding the contextual vocabulary of the model, we had the intuition this would yield increased performance on the unseen test set.

Of all of the preprocessing and tuning steps we tried on our internal evaluations, none resulted in a larger accuracy boost than adding this back-translated data.

5 Future Work

New pre-trained neural network models are being released at a rapid pace. The trend is that they are getting larger–trained with more parameters, on larger quantities of text. Additionally, their baseline capabilities are expanding. Work like that which is presented here can be easily updated to take advantage of these new models as they become available. The workflow a year from now will

[8] as well as from up-sampling the English training set

be the same, but performance will improve. Today, BERT and similar pre-trained models have become the new baseline. These systems yield fantastic results, with little training data required for fine-tuning.

As larger models are created and released, the models become more difficult to understand. Classification and ranking is helpful to support SMEs performing their work, but full decision support systems cannot be black boxes, and need to be able to explain why they made the suggestions they did. We are working on improving the explainability of these models to provide better support to decision makers.

6 Conclusions

This paper introduced work by Accenture on using BERT and RoBERTa models to classify and rank unsubstantiated claims in social media for professional fact-checking. We demonstrate 5 models. We submitted one model to the English track, and placed 1st with a mAP of .8064. We submitted 4 models to the Arabic track, yielding 1st (P@30=.7000), 2nd (P@30=.6750), 3rd (P@30=.6694), and 4th (P@30=.6639) place.

References

1. Abdelali, A., Darwish, K., Durrani, N., Mubarak, H.: Farasa: A fast and furious segmenter for arabic. In: Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Demonstrations. pp. 11–16 (2016)
2. Aluru, S.S., Mathew, B., Saha, P., Mukherjee, A.: Deep learning models for multilingual hate speech detection. arXiv preprint arXiv:2004.06465 (2020)
3. Antoun, W., Baly, F., Hazem, H.: AraBERT: Transformer-based model for arabic language understanding. In: Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection. pp. 9–15 (2020), `https://arxiv.org/pdf/2003.00104v2.pdf`
4. Arampatzis, A., Kanoulas, E., Tsikrika, T., Vrochidis, S., Joho, H., Lioma, C., Eickhoff, C., Névéol, A., Cappellato, L., Ferro, N. (eds.): Experimental IR Meets Multilinguality, Multimodality, and Interaction Proceedings of the Eleventh International Conference of the CLEF Association (CLEF 2020). LNCS (12260), Springer (2020)
5. Barrón-Cedeño, A., Elsayed, T., Nakov, P., Da San Martino, G., Hasanain, M., Suwaileh, R., Haouari, F., Babulkov, N., Hamdan, B., Nikolov, A., Shaar, S., Sheikh Ali, Z.: Overview of CheckThat! 2020: Automatic identification and verification of claims in social media. In: Arampatzis et al. [4]
6. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners (2020)

7. Cappellato, L., Eickhoff, C., Ferro, N., Névéol, A. (eds.): Working Notes of CLEF 2020—Conference and Labs of the Evaluation Forum (2020)
8. Committee, O.: Tasks 1 & 5: Check-worthiness, `https://sites.google.com/view/clef2020-checkthat/tasks/tasks-1-5-check-worthiness`
9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
10. Hasanain, M., Haouari, F., Suwaileh, R., Ali, Z., Hamdan, B., Elsayed, T., Barrón-Cedeño, A., Da San Martino, G., Nakov, P.: Overview of CheckThat! 2020 Arabic: Automatic identification and verification of claims in social media. In: Cappellato et al. [7]
11. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 (2012)
12. Howard, J., Ruder, S.: Fine-tuned language models for text classification. CoRR **abs/1801.06146** (2018), `http://arxiv.org/abs/1801.06146`
13. Lai, G., Xie, Q., Liu, H., Yang, Y., Hovy, E.: Race: Large-scale reading comprehension dataset from examinations (2017)
14. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: RoBERTa: A robustly optimized BERT pretraining approach. CoRR **abs/1907.11692** (2019), `http://arxiv.org/abs/1907.11692`
15. Müller, M., Salathé, M., Kummervold, P.E.: Covid-twitter-bert: A natural language processing model to analyse COVID-19 content on twitter. arXiv preprint arXiv:2005.07503 (2020)
16. Ortiz Suárez, P.J., Romary, L., Sagot, B.: A monolingual approach to contextualized word embeddings for mid-resource languages. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.acl-main.156, `http://dx.doi.org/10.18653/v1/2020.acl-main.156`
17. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Proc. of NAACL (2018)
18. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. Tech. rep., OpenAI, San Francisco, CA, USA (2019)
19. Rajpurkar, P., Jia, R., Liang, P.: Know what you don't know: Unanswerable questions for squad (2018)
20. Rasmy, L., Xiang, Y., Xie, Z., Tao, C., Zhi, D.: Med-bert: pre-trained contextualized embeddings on large-scale structured electronic health records for disease prediction (2020)
21. Safaya, A., Abdullatif, M., Yuret, D.: Kuisail at semeval-2020 task 12: Bert-cnn for offensive speech identification in social media. In: Proceedings of the International Workshop on Semantic Evaluation (SemEval) (2020)
22. Schuster, M., Nakajima, K.: Japanese and Korean voice search. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 5149–5152. IEEE (2012)
23. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units (2015)
24. Shaar, S., Nikolov, A., Babulkov, N., Alam, F., Barrón-Cedeño, A., Elsayed, T., Hasanain, M., Suwaileh, R., Haouari, F., Da San Martino, G., Nakov, P.: Overview of CheckThat! 2020 English: Automatic identification and verification of claims in social media. In: Cappellato et al. [7]

25. Sharir, O., Peleg, B., Shoham, Y.: The cost of training NLP models: A concise overview. arXiv preprint arXiv:2004.08900v1 (2020)
26. Sun, Y., Wang, S., Li, Y., Feng, S., Chen, X., Zhang, H., Tian, X., Zhu, D., Tian, H., Wu, H.: Ernie: Enhanced representation through knowledge integration. arXiv preprint arXiv:1904.09223 (2019)
27. Turc, I., Chang, M.W., Lee, K., Toutanova, K.: Well-read students learn better: On the importance of pre-training compact models (2019)
28. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R.: Glue: A multi-task benchmark and analysis platform for natural language understanding (2018)
29. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Brew, J.: Huggingface's transformers: State-of-the-art natural language processing. ArXiv **abs/1910.03771** (2019)
30. Zhu, Y., Kiros, R., Zemel, R.S., Salakhutdinov, R., Urtasun, R., Torralba, A., Fidler, S.: Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. CoRR **abs/1506.06724** (2015), http://arxiv.org/abs/1506.06724