Question Answering Using SimpleTransformers Library

Boğaziçi University

Mine Gazioğlu[1], Günay Eser[1], and İlker Kurtuluş[1]

[1]Department of Computational Science and Engineering

Abstract

Question Answering is focused on building systems that automatically answer questions posed by humans.[1] There are various subcategories of Question Answering Systems. In this paper we are focusing on a Question Answering system that is classified as a single-document open-domain which is also referenced as reading comprehension. Here the NLP system is given a paragraph (or a single document) that may or may not contain the answer to the question, based on the context, the system should answer the question.[2][3] As the single-document open domain system dataset we will use TQUAD, that is the Turkish Question-Answering dataset on Turkish and Islamic Science History within the scope of Teknofest 2018 Artificial Intelligence competition and as for QA task we will use Simple Transformers Library based on Transformers Library by HuggingFace for the Turkish Question Answering task.We created a test dataset based on Turkish Wikipedia and a use case for which we collected tweets manually.

Introduction

Ranging from automatic question answering systems to search engines, NLP systems have had a huge contribution to our ability to access knowledge stored in text.Here we use a Question Answering system that is classified as a single-document open-domain which is also referenced as reading comprehension.[2][3] There are, broadly categorizing, 2 types of Question Answering Systems :

- Closed domain question answering systems
- Open domain question answering systems

For closed domain systems, any question asked has to be about the specific domain or have a limited vocabulary. Open domain systems deal with questions about nearly anything, and can only rely on general ontologies and world knowledge.These systems concern less straight-forward questions. [2][3]Some of the most popular open domain datasets are SQUAD, WikiQA and TREC-QA [5][13][14]

Large and high-quality annotated corpora are usually scant for languages other than English. While most prominent Question Answering Datasets support solely English language there are some ongoing research and some datasets with multilingual options.One of the reasons why producing dataset in different languages is hard is that different languages express meaning in structurally different ways. E.g. TyDi QA is a multilingual dataset that contains many questions and answers in various languages. [4]There are also some work on translation of SQUAD into other languages using Neural Machine Translation. [6]

Our first approach was to make research on available datasets with Turkish-language instead of creating a crowdsourced dataset. A dataset called TQUAD was publicly available for use on github. [7] This dataset is most likely manually constructed considering how SQUAD(Stanford Question Answering Dataset)[5] was created. We use Simple Transformers library for Question Answering Task using a particular context for each question. Simple Transformers Library is based on the Transformers library by HuggingFace and it simplifies implementation of some NLP models such as question answering. It has support for many Transformer Models.[8] We manually created a test set and a use case. Test set using Turkish Wikipedia and the use case using the selected Twitter trending-topic/hashtag tweets. We do not have baseline scores to compare our results with since no pretrained turkish language models on huggingface published any evaluation of their model on TQUAD. We attribute this deficiency to lack of turkish question answering datasets publicly available. As a comparison of our final results we compared the scores of models with fine + hyperparameter tuning and the models with no tuning.

As for evaluation we used the Exact Match and F1-score metrics also used to evaluate SQUAD.Additionally we created a metric of our

own regarded as Extended match which in core measures how much similarity is captured between the ground truth and prediction. We did hyperparameter tuning and fine tuning before getting results. As for hyperparameters to be tuned, we only used epoch number and batch size as further search was computationally expensive and we did not have computational resources for that.

"?" model gave the best Exact match and F1-scores with epoch number ... and batch size ...For twitter use case we manually assigned a score from 1 to 5 showing how much the most frequent word sum up the content of the trending topic.The best/most related...

Related Work

Question Answering Using crowd-sourcing have been actively pursued in the recent years.[5][9] Reading comprehension is the task to answer a natural language question given a paragraph or a single document. For open domain question answering, without specific context, answers should be extracted from a large collection e.g. web or a document. For extractive question answering, answer is a segment of the document or the paragraph. Example datasets for extractive Question Answering include SQUAD, NewsQA etc.[5][10] Transformer based pretraining methods include BERT, XLNet etc. These methods have helped to build Question Answering models.[11][12]

SQUAD

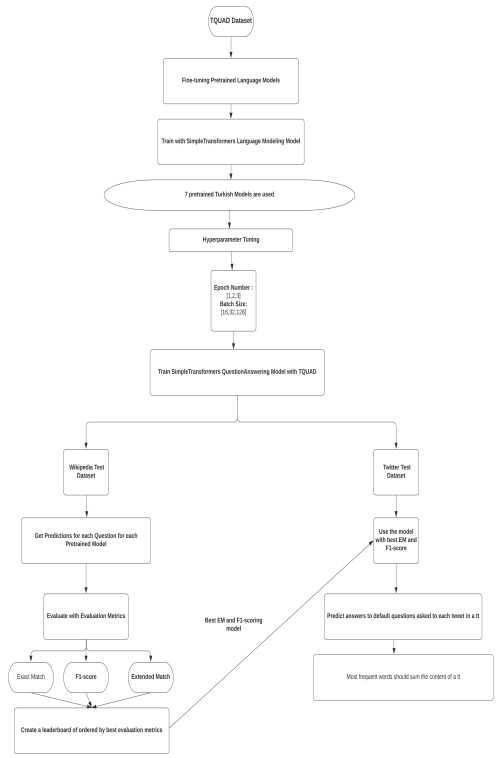
The most popular Question answering study, Stanford Question Answering Dataset (SQUAD) is a reading comprehension dataset, that comprises of Wikipedia articles, questions generated by crowdworkers and their answers which are a segment of text of the corresponding context passage. The questions can be either answerable or unanswerable. For SQUAD, the dataset was collected in three stages:

- Curating passages
- Crowdsourcing question-answers on those passages
- And obtaining additional answers

In summary they retrieved high-quality articles, extracted individual paragraphs from each article with some preprocessing like stripping away images,figures,tables and discarding paragraphs shorter than 500 characters.They partitioned the main dataset into train,development and test sets. For question-answer collection they employed crowdworkers to create up to 5 questions and answers on the content of the paragraph and are paid hourly. To make evaluation more robust and get human performance on SQUAD dataset they added at least 2 additional answers to each question in development and test sets.[5]

Recently Transformer based pretraining methods like BERT [9] and XLNet [39] have helped to build QA models outperforming Humans on reading comprehension on SQuAD .

Approach



Flowchart of progress.

Experiments

Data

Training Set

Question Answering Model is trained on 8308 questions and answers that belong to 681 topics and 2232 contexts on a dataset called TQUAD.This dataset is the Turkish Question Answering dataset on Turkish and Islamic Science History within the scope of Teknofest 2018 Artificial Intelligence competition. The dataset comprises of title,context,questions,answers,question id and an answer starting index. Title is topic names on the general broad topics that can contain multiple context paragraphs. Context paragraphs explain a concept. For one context paragraph there are multiple questions and answers. Each question,answer pair is represented by a unique id and an answer start index. The answer start index is used in a way that that the length of the answer

is added to start index when using Simple Transformers library for Question Answering Task.

Test Set

Wikipedia Test Set
We have prepared 131 test questions manually on 23 various topics from Turkish Wikipedia. The topics are:

- Amerika Birleşik Devletleri
- Ay
- Boğaziçi Üniversitesi
- Budizm
- Candan Erçetin
- Canlı video oyun
- Covid-19
- DNA
- ESP Gitarları
- Eyfel Kulesi
- Facebook
- Fil
- Fotoğraf
- Göl
- Heavy Metal
- Keman
- Keops Piramidi
- Michael Ende
- Penguen
- Sait Faik Abasıyanık
- Türkiye'de Karate
- Youtube
- Çanakkale Savaşı

In the test set multiple questions are asked given a context, each with a unique id and a unique answer. The topics are not chosen by a certain criteria, we tried to keep the context range as broad as possible. The contexts can be scientific, about historical events, people, social media platforms, objects, animals and so on.Although, we did not have a conspicuous criteria for the topics, the context excerpted from Wikipedia should not be too short and should be in accord with the content of the topic.The context should have variety of explanatory information on the context e.g date,location.The answers are created such that they are a segment of the context paragraph. Questions vary on a scale of straightforwardness. Some questions require direct excerpt of a segment from the context as answer. A few select questions are not as direct as majority.

Twitter Data (Use Case)

When collecting twitter data, there were some certain qualifications we sought both on the trend-topic/hashtag and the tweets about the topic. Some hashtags can be bot-made and some hashtags are insignificant with the information it contains. Therefore, we tried to maintain informative hashtags/trending-topics in the election process. Our main purpose is to choose hashtags that is questionable and get answers to "What is going on?". (Note that this is not the question we asked in the testing process.)

We delicately picked the trending-topics/hashtags according to their affairs in the agenda. Some of the filters we applied when choosing the hashtags/trending-topics:

- There needed to be a clear and understandable problem about the topic mentioned in the hashtag.
- There needed to be an affair occurred recently about the topic.
- There needed to be a request or demand about a topic and the grounds should be clear.

We selected our hashtags/trending-topics according to these qualifications of the topics they are about. Next, we proceeded towards tweet collecting process.

When collecting tweets that contain the hashtag/trending-topic we have chosen. We did not use the Official API of Twitter. Instead, we picked the tweets manually which are suitable for our purpose. The conditions of this suitability are quite natural and aim to keep the dataset clean. The main reason we picked tweets manually instead of via API is to avoid preprocessing process through our work. We ensured that each tweet we collected to has qualifications like:

- It must be tweeted about the topic, (Some tweets might contain the hashtag and can mention about something completely different.)
- It can consist of emotion, opinion or judgement,
- It could has a piece of information about the topic,
- It explains a reason or give grounds for the demand/request.
- It must not be too short.
- It must not have too much typos, since model needs to recognize the words. (We allowed some extent of typos in tweets for reflecting the real twitter case.)
- It should preserve Turkish language grammar structure to some extent as well.

We also tried not to pick tweets that contain very similar structure and meaning. Considering these conditions, we collected tweets for 4 trending-topics/hashtags from Twitter:

- demet evgar
- serdar hocanın yanındayız
- YKSyi erteleyin
- kıdem tazminatıma dokunma

Methodology

We have made research on available datasets with Turkish-language.We could either create a crowdsourced dataset or use an existing one. As explained in Related Works creating a crowdsourced dataset needs financial resources and a certain span of time. For these reasons were not able to create our own training dataset.We also note a scarcity of Turkish reading comprehension datasets.Here we use TQUAD, that is the Turkish Question Answering dataset on Turkish and Islamic Science History within the scope of Teknofest 2018 Artificial Intelligence competition.[7]We use Simple Transformers library for Question Answering Task.Simple Transformers Library is based on the Transformers library by HuggingFace. It has support for many Transformer Models.[8]

As for evaluation,we used the script used to test SQUAD. The metrics include exact match and F1-score. Additionally we created a metric called Extended Match which gives an overall measure of similarity between ground truth and predictions. We did hyperparameter and fine tuning before getting results. As for hyperparameter tuning we only tuned epoch number and batch size as further search was computationally expensive and we did not have computational resources for that.

To evaluate our results after training with 7 pretrained language models we created a test set and a use case. Test set is created using Turkish Wikipedia and the use case using selected Twitter trending topics tweets manually. As we could not find any other model performances that were obtained using TQUAD we do not yet have baseline scores to compare our scores with. However, we compare the results of hyperparameter and fine tuned models with the ones that have not been tuned.

To be able to train with TQUAD and make predictions on the test set the input to QuestionAnswering from SimpleTransformers should be of a certain format as such we sculpted train and test data into the necessary input format. The input subsets for training data are context, question id, start index, question and answer. The question start index plus the length of the answer is used to determine the location of the answer in the context paragraph. For test data the input for prediction should include context paragraph, question and question id.

Pre-trained Models

dbmdz/bert-base-turkish-* Current version of the model is trained on a filtered and sentence segmented version of the Turkish OSCAR corpus, a recent Wikipedia dump, various OPUS corpora and a special corpus provided by Kemal Oflazer. The final training corpus has a size of 35GB and 44,04,976,662 tokens. Detailed flow of training can be found here: https://github.com/stefan-it/turkish-bert/blob/master/CHEATSHEET.md

dbmdz/distilbert-base-turkish-* This model is trained a distilled version of BERTurk, that uses knowledge-distillation from BERTurk (teacher model).

dbmdz/electra-base-turkish-* The ELECTRA base model was trained with the official implementation. They used the same datasets as for BERTurk(for evaluation). They evaluated their model with PoS and NER tasks. https://github.com/stefan-it/turkish-bert/blob/master/electra/README.md

savasy/bert-base-turkish-squad-* This model is fine tuned with dbmdz/bert-base-turkish and trained on TQUAD dataset. Owner of this model did not published any evaluation results as well.

Unfortunately we could not find any information or background about the model lserinol/bert-turkish-question-answering, therefore it's training source remain unknown for us.

Hyperparameter Optimization and Fine-Tuning

Before training question answering model on TQUAD dataset we performed hyperparameter optimization and fine tuning of model weights with respect to the TQUAD data. As for pre-trained models we collected the pre-trained models from: `https://huggingface.co/models?search=turkish`

These pre-trained models include BERT, Electra, Distilbert language models. We fine-tuned these pre-trained language models in accord with TQUAD data which we use as training data. For fine-tuning process we use Simple Transformer's LanguageModeling model to train with TQUAD then used the tuned weights, vocabulary and model configurations from Language model for the Question Answering Model. Hyperparameter optimization is carried out for 2 hyperparameters: epoch number and batch size. We have provided options 1, 2, 3 for epoch number and 16, 32, 128 for batch size and got predictions for their combinations of use. A total of 45 models are run and we recorded the predictions made by each of these models and then chose the best performing hyperparameters and the best performing pre-trained model with SQUAD's evaluation script that contains exact match and F1-scores. We also checked the results for the metric of our creation, Extended Match, however the best score for Extended Match disagreed with the best score of Exact Match and F1-score. Best Exact Match and F1-scores are congruent. We then made predictions on the test data with the best performing hyperparameter combination and pretrained model.

Predictions

Predictions are made on Wikipedia Test Set

with 45 models that is trained with TQUAD dataset with hyperparameter tuning and fine-tuning. Predictions from the QuestionAnswering Model come as multiple sentences for each question.This means we have many predictions for any question. These predictions are ordered with the most probable answer at the top and least probable according to the model at the bottom. We selected the most probable answer which is the top one in cases where it is not an empty string. But in some cases the model would return an empty string as the most probable answer, which means it is decided that the question does not have an answer in the context paragraph. For these cases, we searched for the next non-empty most probable prediction for the answer. For SQUAD dataset, there are answerable and non-answerable questions. However, in our case we had given a unique answer to all the questions and therefore did not endorse empty strings as predictions.

As for evaluation, we used SQUAD's evaluation script that contains Exact Match score and F1-Score used to measure performance. As well as the extended measure metric that measures how similar ground truth and predictions are.

Experimental Details and Evaluation Method

Before explaining evaluation process, an explanation of the evaluation metrics we will use:

Exact Match: This evaluation metric measures whether the ground truth matches the predicted answer exactly. This is a strict binary metric that demands absolute match of predictions and actual answers. Exact match score for one question is either 1 or 0 ; 1 denoting complete match between prediction and answer while 0 is mismatch.The final result is determined by the sum of exact matches over total number of observations.Exact match is computed for each prediction then averaged over the test set to get a final score.

F1-Score: In technical terms it is the harmonic mean of precision and recall. This is a less strict metric compared to Exact Match. To elucidate the concept with an example; let us say the ground truth be "Most mammals on earth are cows" and the prediction be "Most mammals are cows". Here we need to compute precision and recall first to get to F1. First we need to find the number of common words between ground truth and predictions. Common word number is 4. Then precision is common word count over total number of words in the predicted answer and recall is common word count over total number of words in the ground truth. F1 score is then harmonic mean of these 2. F1 score is averaged over the dataset to get a final score.

Extended Match: This metric is our creation to assess how similar ground truth and predictions are. This metric is used to mimic how human evaluation would work. As for all other metrics to minimize human error(as questions and answers are hand-written for test set questions) we remove punctuations, make all sentences lower-case and remove extra spaces inter-sentence.Then we remove Turkish stopwords ,adding more stopwords manually checking what wording differences between ground truth and predictions decrease similarity between them.If the predictions is not an empty string and it is a subsentence of the ground truth we consider the sentence similar. This is a binary metric that assigns 1 when similarity occurs and 0 if the prediction is not a subsentence of the ground truth.
Important to note here that some preprocessing of both predicted and actual answers are made before evaluating them. As formation of a dataset is a manual process, there might be punctuation, spacing or casing mistakes when answers are manually added. SQUAD's evaluation script also takes care of this issue.

Results

The best performing model in terms of Exact Match Score and F1-Score was fine-tuned dbmdz/bert-base-turkish-cased, a model trained on a filtered and sentence segmented version of the Turkish OSCAR corpus,with epoch number 3 and batch size 16.Runner up model was the model trained with fine-tuned savasy/bert-base-turkish-squad with epoch number 1 and batch size 16. Out of all models dbmdz/distilbert-base-turkish-cased performed the worst.The best extended match score belongs to lserinol/bert-turkish-question-answering. In the second table presented results for all 7 pretrained models without fine-tuning or hyper-parameter tuning is presented. Best exact match score belongs to 2 models: dbmdz/bert-base-turkish-cased and lserinol/bert-turkish-question-answering. Best F1 and extended match scores belong to lserinol/bert-turkish-question-answering. We see an increase in all 3 scores when we perform hyperparameter and finetuning. dbmdz/distilbert-base-turkish-cased performed worst in both cases.

Twitter Use Case

We mentioned about the twitter data we collected in the section Twitter Data. As we obtained the best model from wikipedia test evaluation process (See Table 1), we wanted to try this model on a twitter use case we think of. Our aim here is to ask model some generic questions and find why the corresponding trending-topic is trending. As we mentioned, each question answering data instances consist of a context and a question and

model tries to find answer to the question from the given context.
Our methodology here is as follows:

- Take one trending topic.
- Consider each tweet about corresponding trending topic as a context.
- Build Questions as **Trending Topic + One of the Generic** Questions.
  For example, if the considered trending topic is "Demet Evgar ("Demet Evgar'a ne oldu")
- Our generic questions are:

  'a ne oldu?

  'e ne oldu?

  ne zaman?

  nerede?

  neden?

  ne için?

  kimdir?

  kim?
- With given context (one tweet) and question (trending topic + question), collect answers for each tweet.
- From these answers, find most frequent words.
- Those most frequent words would tell about why the trending topic is trending.

With this approach, we asked our best model **dbmdz/bert-base-turkish-cased** (with Epoch 3 and Batch Size 16) these questions for each trending topic and get results.

Our trending topics were, as we mentioned:

- demet evgar
- serdar hocanın yanındayız
- YKSyi erteleyin
- kıdem tazminatıma dokunma

For Demet Evgar's case (Famous Turkish actress), the real reason for Demet Evgar to became trending was her photograph showing her face after a bee sting.

When we look at the most frequent words we obtained from our model:

- 2020 - 15 times
- arı (bee) - 14 times
- sana (to you) - 8 times
- hayat (life) - 7 times
- yaşında (on the age) - 7 times
- 40 - 7 times
- bana (to me) - 5 times
- yapmaz (it won't) - 5 times
- demet - 5 times
- hastanede (in the hospital) - 5 times

We can see that we can at least have a clue of what happened and why Demet Evgar got to the trending topics by looking at these frequent words obtained from the answers. You can check other results we obtained in Figure 3.

Analysis

The leading models such as dbmdz/bert-base-turkish-cased in final leaderboard was trained on corpora of size 35GB with 44,04,976,662 tokens. Larger corpora size can set a ground for the performance of this pretrained model. Runner up savasy/bert-base-turkish-squad was trained on TQUAD however, there is a development set available in TQUAD repository and we suspect the owner of the model could have merged train and development sets to train the model and the owner did not include the details. We do not have insights into how lserinol/bert-turkish-question-answering,another model of good performance, was created. The worst performing model,dbmdz/distilbert-base-turkish-cased,which has prediction accuracy 0 percent. Is created such that it is trained on 7GB of the Oscar Corpus. It is supposed to be a fast and small model that has 40 percent less parameters than bert-base-uncased. As we can see the training time is significantly short compared to other pretrained models. However, it does not preserve performance in our case.

As for the twitter case, due to time constraints, we could not manage to collect vast amount of twitter data for the use case, however, we wanted to see that the model on the top of our leaderboard can be used for such work as well.

Future Work

The TQUAD question answering dataset with 8300 questions, surely not enough for perfecting language models. The more question data from many different context, would surely provide a base for models to be improved significantly. Our main pupose here was to build a leaderboard with existing turkish language models on huggingface and experiment a use case, which we did for twitter data, and see possible capabilities of models. There are some things we could not manage during our work and keep as future work such as:

- We could have used development set with training set to get a better traning result.

- We could have collect more wikipedia test data than 131 for evaluation.
- Due to our computational constraints, we could not include Electra Based turkish models like **erayyildiz/electra-turkish-cased**, that could also be included in a future work.

Question answering datasets are very limited at the time of this paper, and they are very expensive to build as we learned from SQUAD [5] paper. We believe that as quality and size of NLP researches grow in Turkish language, there would be more datasets available on Question Answering. For twitter use case, we used such an approach that, we consider tweets as context and ask questions to tweets. Surely there are some other work that could be done, but we keep them as future works, such as:

- We could have also combined each tweet on particular trending topic and consider all as one context and then ask questions as well.
- Instead of picking tweets manually, we could use twitter API and collect tweets in an automated way and then preprocess the tweets using tukishlp module of python. It has the features to correct typos, misplaced space characters and etc.
- We could have built a custom evaluation method for evaluating twitter question answers to measure the performance using certain twitter accounts that explains each trending topic.

Conclusion

We present resulting scores for Question Answering task trained on TQUAD dataset. These scores might serve as a baseline for future work using TQUAD. A few notes on what could be the follow up should be stated. As mentioned earlier in the paper we have not used development set provided in github repository of TQUAD. Development and training sets can be combined, development set can be used for hyperparameter tuning. These could help further improve our scores. Wikipedia test data can be further expanded. We can add more than one answer for each question like the procedure in SQUAD. We would select the best resulting scores per question among multiple answers. We can also benefit from human evaluation of predictions. We designed Extended match to mimic human evaluation of predictions. We should not assume exact human-like evaluation from Extended Match. Human evaluation would yield more concrete results. For example, best Extended Match score on fine-tuned and hyperparameter tuned model lserinol/bert-turkish-question-answering is 48.62. We assumed for the test dataset, predictions were similar to ground truth 48 percent of the time. With human evaluation we might get a lower score. With better computational resources, a wide variety of hyperparameters can be tuned. Fine-tuning can be performed on train + development dataset(combined). Some pretrained models were not included as the size of the models were too big for our computational resources to handle. After all, we exhibited that even with 48.62 Extended Score we can use these models on some real life cases, twitter in our case. The best model can give information about a trending topic's reason for becoming a trending topic to some extent. We expect that with accelerating research speed on Turkish NLP, there will be significant advancements on questions answering and reading comprehension.

References

[1] Philipp Cimiano; Christina Unger; John McCrae (1 March 2014). Ontology-Based Interpretation of Natural Language. Morgan Claypool Publishers. ISBN 978-1-60845-990-2.

[2] Mervin, R., 2013. An Overview Of Question Answering System.

[3] A. Chandra Obula Reddy "A Survey on Types of Question Answering System." IOSR Journal of Computer Engineering (IOSR-JCE) 19.6 (2017): 19-23

[4] Google AI Blog, 2020. TyDi QA: A Multilingual Question Answering Benchmark. Available at: <https://ai.googleblog.com/2020/02/tydi-qa-multilingual-question-answering.html> [Accessed 13 July 2020]

[5] Rajpurkar, P., Zhang, J., Lopyrev, K. and Liang, P., 2016. Squad: 100,000+ Questions For Machine Comprehension Of Text. [online] arXiv.org. Available at: <https://arxiv.org/abs/1606.05250> [Accessed 13 July 2020

[6] Carrino, Casimiro Costa-jussà, Marta Fonollosa, José. (2019). Automatic Spanish Translation of the SQuAD Dataset for Multilingual Question Answering.

[7] Peker, Mehmet Ali,TQuad/turkish-nlp-qa-dataset ,(2018), Github Repository, https://github.com/TQuad/turkish-nlp-qa-dataset

[8] Rajapakse, Thiliana,Simple Transformers ,(2019), Github Repository, github.com/ ThilinaRajapakse/simpletransformers

[9] Rajpurkar, Pranav Jia, Robin Liang, Percy. (2018). Know What You Don't Know: Unanswerable Questions for SQuAD. 784-789. 10.18653/v1/P18-2124.

[10] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. CoRR, abs/1611.09830,

Pretrained Turkish BERT Models	Exact Match	F1-score	Extended Match	Epoch	Batch Size	Training Time
dbmdz/bert-base-turkish-cased	32.11	45.68	47.71	3	16	0:08:12
savasy/bert-base-turkish-squad	31.19	44.99	46.79	1	16	0:05:05
dbmdz/bert-base-turkish-cased	27.52	42.39	44.04	3	32	0:06:57
lserinol/bert-turkish-question-answering	27.52	43.73	44.95	3	16	0:08:11
lserinol/bert-turkish-question-answering	26.61	43.29	46.79	2	32	0:05:52
dbmdz/bert-base-turkish-cased	26.61	42.44	45.87	2	16	0:06:37
savasy/bert-base-turkish-squad	25.69	44.11	43.12	1	32	0:04:41
dbmdz/bert-base-turkish-cased	25.69	39.2	42.2	1	16	0:05:02
lserinol/bert-turkish-question-answering	24.77	43.78	45.87	2	16	0:06:38
savasy/bert-base-turkish-squad	24.77	41.77	39.45	3	16	0:08:08
savasy/bert-base-turkish-squad	24.77	41.94	42.2	2	16	0:06:40
savasy/bert-base-turkish-squad	24.77	41.46	42.2	3	32	0:07:00
lserinol/bert-turkish-question-answering	24.77	43.9	48.62	1	16	0:05:06
dbmdz/bert-base-turkish-cased	24.77	40.29	44.04	2	32	0:05:55
savasy/bert-base-turkish-squad	23.85	42.31	41.28	2	32	0:05:51
lserinol/bert-turkish-question-answering	23.85	43.82	45.87	3	32	0:07:01
lserinol/bert-turkish-question-answering	22.94	44.26	45.87	1	32	0:04:42
dbmdz/bert-base-turkish-cased	22.94	41.16	44.04	1	32	0:04:44
savasy/bert-turkish-uncased-qnli	22.02	39.17	37.61	3	16	0:08:09
dbmdz/bert-base-turkish-uncased	22.02	40.56	37.61	3	16	0:08:05
dbmdz/bert-base-turkish-uncased	22.02	37.87	40.37	2	16	0:06:36
savasy/bert-base-turkish-sentiment-cased	21.1	39.93	40.37	2	32	0:05:53
savasy/bert-base-turkish-sentiment-cased	21.1	38.73	44.95	2	16	0:06:38
dbmdz/bert-base-turkish-uncased	20.18	38.62	39.45	3	32	0:07:00
dbmdz/bert-base-turkish-uncased	20.18	35.65	36.7	1	32	0:04:42
savasy/bert-base-turkish-sentiment-cased	19.27	38.17	40.37	3	32	0:07:02
savasy/bert-turkish-uncased-qnli	19.27	36.29	36.7	2	32	0:05:53
dbmdz/bert-base-turkish-uncased	19.27	36.3	36.7	2	32	0:05:53
savasy/bert-turkish-uncased-qnli	18.35	36.43	38.53	3	32	0:07:02
savasy/bert-base-turkish-sentiment-cased	18.35	38.64	36.7	3	16	0:08:08
savasy/bert-turkish-uncased-qnli	18.35	36.45	37.61	1	16	0:05:08
dbmdz/bert-base-turkish-uncased	18.35	32.95	35.78	1	16	0:05:11
savasy/bert-turkish-uncased-qnli	16.51	33.84	33.03	2	16	0:06:42
savasy/bert-base-turkish-sentiment-cased	16.51	31.7	36.7	1	32	0:04:44
dbmdz/distilbert-base-turkish-cased	15.6	32.11	34.86	3	16	0:05:01
savasy/bert-turkish-uncased-qnli	15.6	31.6	32.11	1	32	0:04:43
dbmdz/distilbert-base-turkish-cased	14.68	28.6	33.94	2	16	0:04:10
savasy/bert-base-turkish-sentiment-cased	13.76	29.89	32.11	1	16	0:05:10
dbmdz/distilbert-base-turkish-cased	11.93	23.97	26.61	3	32	0:04:23
dbmdz/distilbert-base-turkish-cased	11.01	19.71	22.02	2	32	0:03:45
dbmdz/distilbert-base-turkish-cased	9.17	16.74	21.1	1	16	0:03:18
dbmdz/distilbert-base-turkish-cased	7.34	17.65	20.18	3	128	0:03:51
dbmdz/distilbert-base-turkish-cased	0.92	2.73	3.67	1	32	0:03:06
dbmdz/distilbert-base-turkish-cased	0	0	1.83	1	128	0:02:56
dbmdz/distilbert-base-turkish-cased	0	0.17	1.83	2	128	0:03:23

Table 1: Fine tuned models' performances for each hyperparameter combination.

2016.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional

[12] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In NeurIPS. 2019

[13] Yang, Yi Yih, Wen-tau Meek, Christopher. (2015). WikiQA: A Challenge Dataset for Open-Domain Question Answering. 2013-2018. 10.18653/v1/D15-1237.

[14] Liu, Donglei Niu, Zhendong Zhang, Chunxia Zhang, Jiadi. (2019). Multi-Scale Deformable CNN for Answer Selection. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2953219.

Pretrained Turkish Language Models	Exact Match	F1-score	Extended Match
dbmdz/bert-base-turkish-cased	22.9	41.85	45.04
lserinol/bert-turkish-question-answering	22.9	45.16	47.33
savasy/bert-base-turkish-squad	21.37	44.07	44.27
dbmdz/bert-base-turkish-uncased	19.08	37.7	38.93
savasy/bert-turkish-uncased-qnli	17.56	39.31	38.93
savasy/bert-base-turkish-sentiment-cased	16.79	36.7	36.64
dbmdz/distilbert-base-turkish-cased	0.76	8.14	2.29

Table 2: Model performances without fine-tuning and hyperparameter tuning

Serdar Hocanın Yanındayız	
serdarhocayanındayız	47
sonuna	7
kadar	7
gelen	5
turizimden	5
kuzuların	5
sizi	5
düşünüyor,	5
seviyor	5
halkı	5
seviyor	5
halkı	5
YKSErtelensin	
sınav	14
ikamet	11
yakın	11
sınavı	11
öğrencinin	10
bir	8
yks	8
olsun	8
35.000	7
empty	7
KıdemTazminatımaDokunma	
işçinin	35
emeklilik	10
bir	8
tamamlayıcı	8
ben	8
işten	7
umudu	7
geleceği	7
alın	7
kolaylastirma	7

Table 3: Most frequent words in answers for other trending topics.

Flowchart of progress.