Research Article
Journal of Intelligent Systems: Theory and Applications 4(2) (2021) 65-75
DOI: 10.38016/jista.815823

A Turkish Question Answering System Based on Deep Learning Neural Networks

Cavide Balkı Gemirter [1*], Dionysis Goularas [2]

[1]Yeditepe University, Faculty of Engineering, Department of Computer Engineering, İstanbul/Turkey,

TEB ARF Teknoloji (Türk Ekonomi Bankası)

[2]Yeditepe University, Faculty of Engineering, Department of Computer Engineering, İstanbul/Turkey

cavidebalki.gemirter@std.yeditepe.edu.tr, goularas@cse.yeditepe.edu.tr

Abstract
In the domain of Natural Language Processing (NLP), despite the progress made for some common languages, difficulties persist for many others for the completion of particular NLP tasks. In this scope, the current study aims to explore these challenges by proposing a question answering (QA) system in the Turkish language. In particular, the system will generate the best answers in terms of content and length from questions that are based on a set of documents related to the banking sector. In order to achieve this goal, the system utilizes advanced artificial intelligence algorithms and large data sets. More specifically, BERT algorithm is used for the generation of the language model, followed by a fine-tuning procedure for performing a machine reading for question answering (MRQA) task. In this work, various experiments were conducted using original and translated data sets in an effort to solve the challenges that arise from morphologically complex languages as Turkish. Finally, the system achieved a performance that overall is applicable to a wider range than any other QA system in the Turkish language. The proposed methodology is not only proper to the Turkish language, but can also be adapted to any other language for performing various NLP tasks.

Keywords: machine reading comprehension, machine reading for question answering, deep learning, BERT.

Derin Öğrenme Sinir Ağlarına Dayalı Türkçe Soru Cevaplama Sistemi

Öz
Doğal Dil İşleme (NLP) alanında, yaygın diller için kaydedilen bazı ilerlemelere rağmen, diğer dillerde belli başlı NLP görevleri için zorluklar devam etmektedir. Bu kapsamda, mevcut çalışma Türkçe dilinde bir soru cevaplama (QA) sistemi önererek bu zorluklara çözüm araştırmayı amaçlamaktadır. Sistem, bankacılık sektöründen seçilen dokümanları kullanarak, sorulan sorulara içerik ve uzunluk açısından en iyi yanıtları üretecektir. Bu amaca ulaşmak için sistem, gelişmiş yapay zeka algoritmaları ve büyük veri kümeleri kullanır. Daha spesifik olarak, dil modelinin oluşturulması için BERT algoritması kullanılmış, ardından sistemin soru cevaplama (MRQA) becerisini arttırmak için bir iyileştirme (fine-tuning) uygulanmıştır. Bu çalışmada, Türkçe gibi morfolojik açıdan karmaşık dillerden kaynaklanan zorlukları çözmek için orijinal ve İngilizce'den çevrilmiş veri setleri kullanılarak çeşitli deneyler yapılmıştır. Son olarak, sistem, genel olarak Türkçe dilinde diğer tüm QA sistemlerinden genel olarak daha yeni bir yelpazede yüksek bir performans elde etmiştir. Önerilen metodoloji sadece Türk diline özgü olmayıp aynı zamanda çeşitli NLP görevlerini yerine getirmek için başka diğer dillerde de uyarlanabilir.

Anahtar Kelimeler: makine okuma anlama, soru cevaplama için makine okuma, derin öğrenme, BERT.

1. Introduction

In recent years, novel artificial intelligence algorithms proposed solutions in problems from various domains and outperformed previous methodologies and architectures. Even if some of the algorithmic ideas were not new, the dramatic increase of available data and process power, various parallelization techniques, cloud-based processing methodologies and the use of graphic processor units (GPU) allowed developing a series of different types of neural networks that demonstrated astonishing results. In the domain of Natural Language Processing (NLP), the use of deep

* Corresponding Author.
E-mail: cavidebalki.gemirter@std.yeditepe.edu.tr

Recieved : 25 October 2020
Revision : 03 March 2021
Accepted : 19 March 2021

neural networks outperformed almost all previous types of approaches. Their ability to create language models from large amounts of data is one of the main reasons for their success. Moreover, recent models managed to generate word representations that change according to their current context, thus giving a dynamic approach and increasing the system performance in various NLP tasks. These networks are configured and tested mainly for English and some other common languages because of the available data sets.

Despite some efforts to create multilingual models, there is still a lack of available data and little experience in the way that these algorithms should be trained and utilized for different languages. In particular, Turkish language has been proven to be challenging for Natural Language Processing because it is an agglutinative language with a derivational structure and morphologically rich. Consequently, the main motivation of this study is to explore the challenges of generating a Turkish language model and performing a particular NLP task, a question answering system (QA). Additionally, another motivation of this work is to explore the difficulties that could arise and propose adequate methods and guidelines for the generation of a language model and the completion of particular tasks in a language structurally different from English.

More specifically, the proposed system will be able to give the best and shorter answer to a question related to the banking sector. In order to achieve this goal, the system will be trained from a variety of data sets. In general, this task is called "Machine Reading for Question Answering" (MRQA) and it is essential for QA systems and search engines in general. To the best of our knowledge, in the Turkish language, a performant MRQA system doesn't exist, as most of the systems follow a semantic approach.

Today, in the domain of NLP the machine learning system that outperforms the state of the art is the Bidirectional Encoder Representatives from Transformers (BERT) (Devlin et al., 2018). Its success lies in the fact that in the generated language model words have a contextual, dynamic representation rather than a fixed one, resulting in a context-sensitive language model. Moreover, as its structure is agnostic and not configured for a particular task or domain, it can be fine-tuned with little effort and perform various NLP tasks. Although BERT proposed a multi-language model, its performance in the Turkish language is not satisfactory. In this context, the purpose of this study is first to create a model of Turkish language based on BERT and then use this model in order to generate a MRQA system oriented to the banking sector. Figure 1 presents a general overview of the study.

2. Related Work

In Turkish questioning (QA) systems, most of the research is focused on improving the skills of search engines by introducing two modules: one that ameliorates the structure of a user's query with specific preprocessing steps, and another that generates a selected list of the most appropriate search results (Amasyalı and Diri, 2005; Biricik et al., 2013; Çelebi et al., 2011; Er and Cicekli, 2013).

One of the tasks of the first module is to detect the question type with the help of a predefined table. Using specialized libraries for Turkish language processing like Zemberek (Akın and Akın, 2007) or Treebank (Eryiğit and Oflazer, 2006; Oflazer et al., 2003), the module analyzes the sentence morphologically and generates the stems of the words. The module can also create simplified variations of the query or eliminate prepositions, conjunctions, stop words, and replicates the query with synonyms of terms using the thesaurus.

In general, these studies utilize rule-based approaches. Their success is limited, and most of them are suitable for factoid questions only. To the best of our knowledge, there is still no approach that utilizes neural networks for a Turkish QA system. Today, NLP domain approaches that are based on neural networks outperform all rule-based systems. Neural networks solutions with pre-trained language representations were available with ELMo (Peters et al., 2018) and Generative Pre-trained Transformer (GPT) (Radford and Salimans, 2018). ELMo presented a bidirectional architecture but it was difficult to be adapted to different tasks. On the other hand, GPT required minimal architectural changes but it was unidirectional. In 2018, Bidirectional Encoder Representations from Transformers (BERT) was published by Google. BERT managed to have a bidirectional architecture by requiring minimal architectural changes for performing various NLP tasks. Within this way, BERT and its recent variations managed to achieve remarkable results.

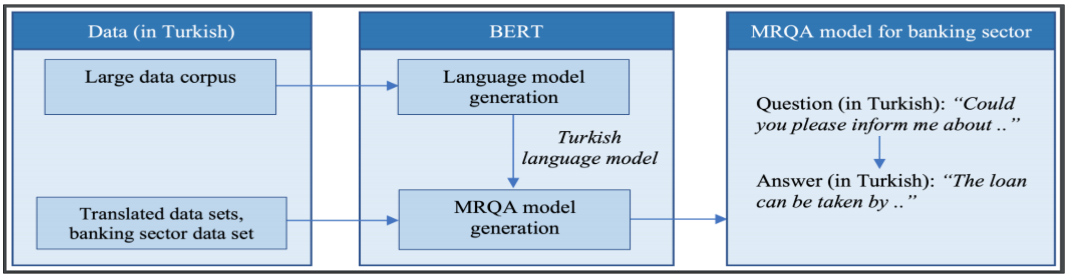

Data (in Turkish)
Large data corpus
Translated data sets, banking sector data set
BERT
Language model generation
Turkish language model
MRQA model generation
MRQA model for banking sector
Question (in Turkish): "Could you please inform me about .."
Answer (in Turkish): "The loan can be taken by .."

Figure 1. Diagrammatic representation of the study.

A recent competition for generalized MRQA tasks in English language (Fisch et al., 2019) included also solutions based on BERT architecture. In this contest, the presence of big multinational corporations demonstrated the increasing interest in the MRQA task worldwide. After its success in English, BERT has been implemented to other languages. Despite the fact that BERT proposes a multilingual model (mBERT), its performance is relatively low. For this reason, for a particular language a special effort has to be made for the generation of a new language model. In (Antoun et al., 2020) the authors generated a BERT model in the Arabic language named AraBERT and tested it in several NLP tasks, including QA. In tests conducted with the Arabic Reading Comprehension Dataset (ARCD), they utilized an English question answering datasets (SQuAD) translated to Arabic. When tested, AraBERT presented a similar performance to mBERT. There exists also a BERT model for the French language named CamemBERT (Martin et al., 2019), for the Korean language named KoBERT[1], and another for the Persian language named ParsBERT (Farahani et al., 2020) but these models do not report accuracy results on QA tasks. Finally, there is a model for the Chinese language trained by Google. This model was tested in the following Machine Reading Comprehension (MRC) datasets generated for Chinese: CMRC 2018, DRCD, CJRC (Cui et al., 2019).

Based on the above, it can be concluded that the existing QA systems in the Turkish language have limited success because of the approaches they utilize. Moreover, despite the progress made for systems in English language, there is still little progress for QA platforms in other languages. In the following section, the methodology for the creation of an MRQA system in the Turkish language is presented.

3. Methodology

3.1. Data sets

When a QA architecture incorporates neural networks, one of the challenges is to generate adequate data sets for the training procedure. In this work, special attention and effort were given in the choice and the generation of pertinent data sets. Two types were generated: one dedicated for a pre-training task for training the language model in Turkish and another for a fine-tuning task in order to perform a QA task for the banking domain. Table 1 presents the data sets used for training the language model. Here, the first data set is based on Wikipedia pages[2], the second on a news article collection in Turkish and the third on a corpus based on the specific domain of the final QA system prepared by the authors of this study. All sentences are in the Turkish language.

Table 2 presents the data sets utilized for the QA task or fine tuning the model. All documents were created based on the Stanford Question Answer Data Set (SQuAD) structure (Rajpurkar et al., 2016), published in 2016.

Table 1. Data sets for the fine-tuning task (QA system for the banking domain).

Name	Size	Content	Information
Wikipedia Corpus (Tr)	456.5 MB	4.5M sentences	Turkish Wikipedia dump 922335 pages (08/2019)
News Corpus (Tr)	2.5 GB	20M sentences	News articles collection in Turkish
Economy Corpus (Tr)	15.5 MB	270K sentences	Turkish economy blogs from Web

Table 2. Data sets for the fine-tuning task (QA system for the banking domain).

Name	Size	Content	Information
SQuAD (Tr)	24.42 MB	490 documents 20963 paragraphs 45872 questions 56117 answers	Q&A from paragraphs from Wikipedia articles. (Machine translation from English to Turkish)
NewsQA (Tr)	19.66 MB	8379 documents 8343 paragraphs 21270 questions 21270 answers	Q&A from articles from CNN news. (Machine translation from English to Turkish)
Banking Sector QA (Tr)	5 MB	679 documents 1637 paragraphs 17708 questions 17708 answers	Q&A from documents from the banking sector. (in Turkish)

[1] GitHub (2020). KoBERT GitHub Page [online]. Website https://github.com/SKTBrain/KoBERT [accessed 25 05 2020].

[2] Wikimedia (2020). Wikipedia Dump [online]. Website https://dumps.wikimedia.org/backup-index.html [accessed 25 05 2020].

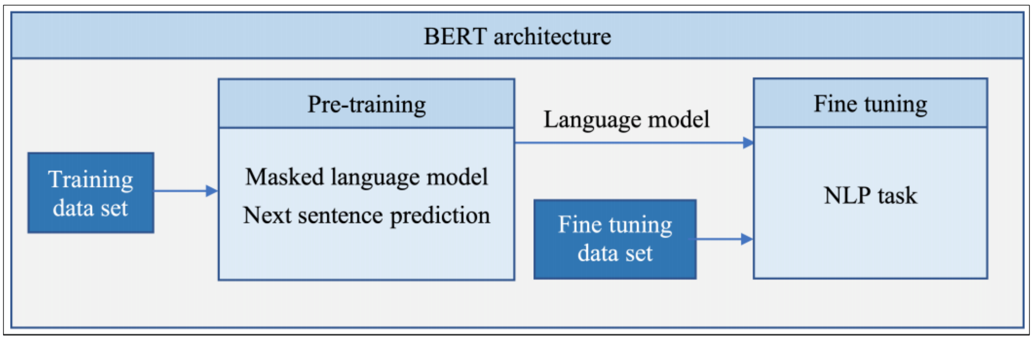


Figure 2. BERT architecture.

A SQuAD based data set includes a set of paragraphs accompanied with a set of questions and answers for each paragraph. The questions are related to the associated paragraph and the answers are generated from its text. The first data set is the original SQuAD data set translated automatically in Turkish. The second is the NewsQA data set (Trischler et al., 2016) translated also automatically in Turkish having also a similar structure with the SQuAD data set.

Finally, the third data set is created by a team working in a private Turkish bank and supervised by the authors of this study. This data set follows the SQuAD structure: a set of paragraphs with a set of related questions and answers for each paragraph.

3.2. BERT

In 2018 Google proposed the Bidirectional Encoder Representations from Transformers (BERT) neural network. First it generates a context-sensitive language model (pre-training task) and then it can perform a series of NLP tasks (fine tuning task). The language model is generated by applying two training procedures simultaneously. The first aims to predict a number of masked words from a sentence and the second aims to predict the following sentence. When the language model is generated, BERT can do a particular NLP task by using a supplementary data set. During the fine-tuning procedure, the weighs of the BERT network are slightly modified. Figure 2 presents the architecture of BERT. In this study, the BERT base model was utilized, having 110 million parameters, 12 transformer layers and 12 attention heads for each transformer layer (Vaswani et al., 2017).

3.2.1. Word Sense Disambiguation in BERT

Turkish is a morphologically rich language with a large number of suffixes and a variety of possible word positioning inside a sentence. In morphologically simpler languages such as English, POS tagging is a much more pertinent procedure. On the contrary, in agglutinating languages such as Turkish, the morphological disambiguation process is challenging. In this case, morphological disambiguation is crucial for finding the stems of the words. Otherwise, the neural network has difficulties in handling the suffixes and, as a result determining the connections between the words. BERT manages to overcome those problems by using:

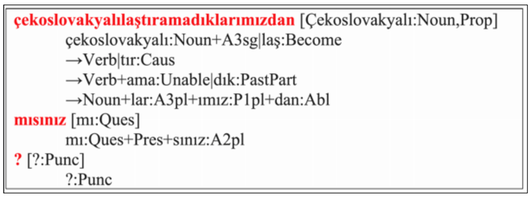
- A subword--based embedding system.
- A Masked Language Model (MLM) and next sentence prediction training.
- Bidirectional transformers.

Although BERT cannot solve the morphological disambiguation problem of Turkish, the overall architecture solves the word sense disambiguation problem, as demonstrated in (Wiedemann et al., 2019) which is enough for performing a number of NLP tasks like text classification, machine translation, and question answering.

3.2.2. Subword-based embedding system

Word embedding methods became popular because they convert an input text into a numerical representation that can be used for mathematical operations in neural networks. Today, new generation word embedding methods capture the contextualized meaning even in cases with polysemy and the resulted word vector can vary according to the context. For example, the vector of the word ‘bank' is different when it is used in a sentence with a finance context and when it describes a seat in a park.

Until modern NLP solutions, morphological disambiguation was performed with rule-based solutions, such as Zemberek for Turkish. ‘Çekoslovakyalılaştıramadıklarımızdan mısınız?’ is the most unusual example in Turkish. Zemberek’s output is shown in Figure 3.



```
çekoslovakyalılaştıramadıklarımızdan [Çekoslovakyalı:Noun,Prop]
        çekoslovakyalı:Noun+A3sg|laş:Become
        →Verb|tır:Caus
        →Verb+ama:Unable|dık:PastPart
        →Noun+lar:A3pl+ımız:P1pl+dan:Abl
mısınız [mı:Ques]
        mı:Ques+Pres+sınız:A2pl
? [?:Punc]
        ?:Punc
```

Figure 3. Morphological disambiguation result of 'Çekoslovakyalılaştıramadıklarımızdan mısınız?' using Zemberek.

WordPiece subword-based embedding (Wu et al., 2016) is a word segmentation algorithm that extracts subwords from a given data set. In the initial step, the WordPiece algorithm splits the corpus into characters, following by recursively combining them into subwords and calculating the loglikelihood of every candidate subword. After several passes over corpus, the algorithm generates a fixed-sized vocabulary by using the frequencies of combined subwords and picks the most frequent ones. The subword can be a word, a syllable, or a single character. Using the vocabulary file as a reference, most of the given text phrases can be tokenized, and the rest are marked as unknow tokens (UNK).

In summary, a desired subword vocabulary size is defined and after splitting words into characters, WordPiece generates the subwords progressively based on likelihood criteria, until a certain threshold is satisfied or the subword vocabulary size is reached. Although WordPiece is not the direct solution to the morphological disambiguation problem of the Turkish language, the result is satisfactory for the tokenization of the input sequences. The tokenization result of 'Çekoslovakyalılaştıramadıklarımızdan mısınız?' with WordPiece is presented in Figure 4. The symbols ## indicate that the sub word is a suffix. Consequently, by using these generated suffixes, the tokenizer can identify the suffixes in Turkish words and also cover many Out-Of-Vocabulary (OOV) words, thus giving an advantage over classical word embedding methods.

3.2.3. *Masked Language Model (MLM) and Next Sentence Prediction training (NSP)*

The training operation is based on two unsupervised tasks that are executed simultaneously: The Masked Language Model (MLM) and the next sentence prediction. The main logic behind the MLM is to try to learn the relationships of words in a language by randomly masking some tokens in the corpus and then try to predict the original ones with an attention bidirectional transformer network that handles the context both from left and right. After tokenizing the content with WordPiece tokenizer, the token embeddings are combined with positional embedding location for preventing long-distance mappings with unnecessary tokens in self-attentions. After Encoder & Decoder self-attention stacks, a Softmax classifier compares the predicted and original words and updates the weights of the network which builds the language model. In MLM, the system attempts to predict 15% of tokens in a sentence that are chosen randomly. During this procedure these tokens are replaced 80% by the token [MASK], 10% by a random word and 10% by the original word. In training, only 1.5% of all tokens are replaced with random words. Compared to traditional language models, as a result of masking process although the training time takes longer, the success of the results is satisfactory. Figure 5 shows examples of token replacements for the three cases.

Çek ##os ##lovak ##yalı ##laştır ##amadı ##k ##larımızdan mısınız ?

Figure 4. Tokenization result of 'Çekoslovakyalılaştıramadıklarımızdan mısınız?' using WordPiece.

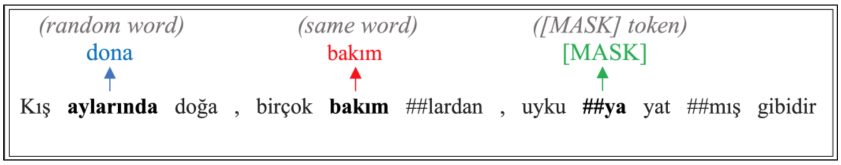

(random word)
dona
(same word)
bakım
([MASK] token)
[MASK]
Kış aylarında doğa , birçok bakım ##lardan , uyku ##ya yat ##mış gibidir

Figure 5. Token replacement examples: random word (blue), same word (red) and [MASK] token (green).

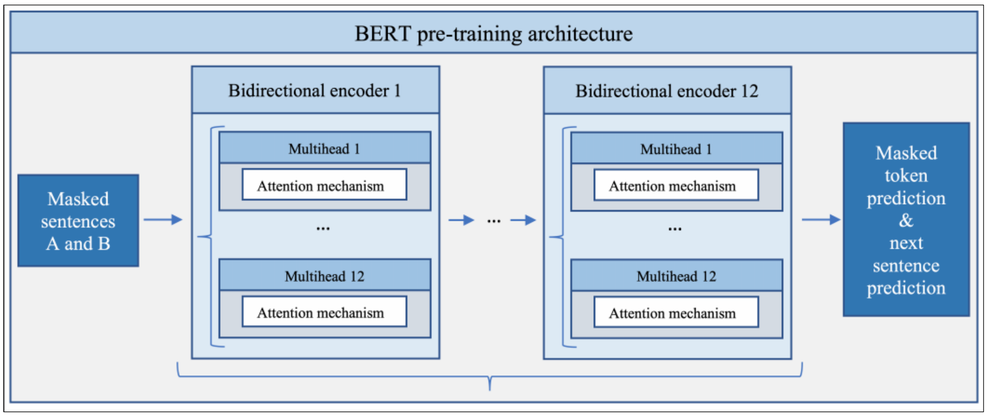


Figure 6. BERT pre-training architecture.

The percentage values are experimental. The reason of adding random words or keeping the original word is to reduce the consequences by the fact that the token [MASK] will not be present in the fine-tuning procedure. Next sentence prediction is a binarized task. In this task, the next sentence is replaced 50% by a random sentence. This procedure aims to predict the relation between sentences by Sigmoid classification and enlarge the contextual meaning that exists in isolated sentences.

3.2.4. Bidirectional transformers

Figure 6 presents the pre-training architecture of BERT. The input consists of unlabeled sentences pairs with masked tokens and the output layer predicts the masked tokens and the next sentence. BERT is using a stack of bidirectional transformer encoders and decoders (Figure 7). The transformer network has encoder and decoders stacks that contains self-attention mechanisms and Feed-Forward networks. This multi-layer architecture aims to generate word vectors that will be adaptable to the context of a sentence. In order to achieve this goal, the encoder architecture applies a word vector encoding that comprises three steps: the first is based on WordPiece encoding, the second on the position of a word in the sentence and the third on a mechanism of comparison of sentence words between them. This mechanism is described as *attention* because a word has an *attention* to the other words of its sentence.

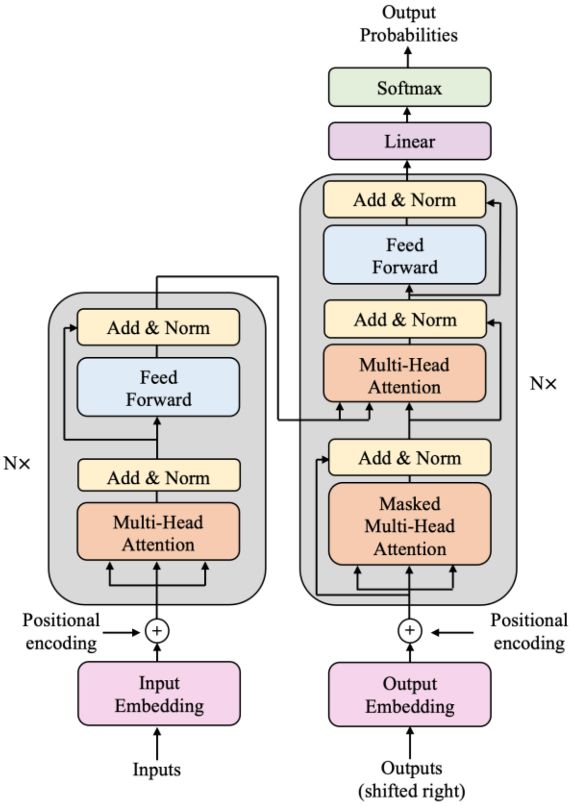


Figure 7. The transformer architecture (Vaswani et al., 2017).

Unlike a single context final state provided by a RNN model, the attention-based model has multiple hidden states, which considers all dependencies between every word in the input sequence. Although the connections are usually with previous or next words in the lower layers of the transformers, the semantic relations become more visible in the higher-level layers. In a traditional Encoder & Decoder architecture, if the input sequence is long, after a while. the model begins to forget some parts of the context Attention tries to solve this problem by focusing on finding the most critical input sections when summarizing the sequence. In an attention-based Encoder & Decoder architecture, there are weight matrices that keep the semantical relation densities of the words and highlight the significant parts of the sequence resulting in performance improvement of Decoder. In the transformer network, the output of every encoder is the input of the next encoder, like a chain. The output of the last encoder is the input of all the decoders. After converting the decoder's output to a logit vector at the top of the decoder stack, the Softmax layer calculates the probabilities, and the transformer picks the top-rated candidate. In summary, the *attention* is based on vector similarity measurements between vectors that are generated from the input vectors and a number of weight matrices.

The attention mechanism is applied many times with a multi-head approach and the resulted vectors are concatenated. The attention heads evaluate the same input of a given layer from different viewpoints. Every multi-head also includes a normalization process and an additional dense neural network. As the formation of a word vector takes into account the other words of the sentence, the system is able to generate context-sensitive word vectors.

3.2.5. Training procedure

The training procedures comprise the generation of a language model in the Turkish Language, a preprocessing of fine-tuning data sets and a fine-tuning procedure for training the QA system for the banking domain.

Pre-training for language model generation: The first step was to create a vocabulary of around 32.000 words. Among them around 30.000 was created from words existing in the data sets of Table 1 and another 2.000 words belonging to the finance sector were selected and added by the authors. WordPiece was utilized for token embedding. In BERT architecture additional embeddings are added based on the position of a word in a sentence and the sentence number. Two training procedures were done simultaneously: one for predicting the next sentence and another that is based on attempting to predict masked words in a sentence. During the training process, the three data sets of Table 1 were used together. Finally, a language model in the Turkish language is generated.

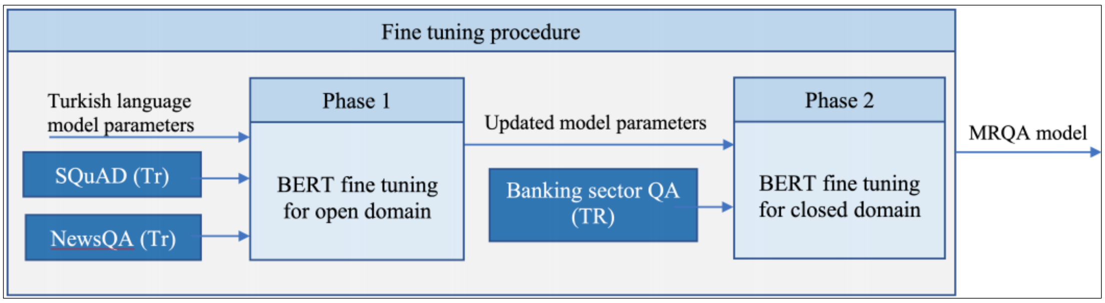


Figure 8. The two phases of the fine-tuning procedure.

Preprocessing of fine-tuning data sets: The automatic translation of data set documents in Table 2 resulted in some inconsistencies related to the SQuAD format. A pre-processing operation was applied aiming to fix problems related to an incorrect answer in a given paragraph or the absence of a start point in a sentence. The preprocessing required the development of some special procedures and in some cases a manual intervention.

Fine tuning for QA task: Fine tuning is applied in two phases. In the first phase, the fine tuning of the neural network which is already trained for a Turkish language model is done by using the SQuAD (Tr) and NewsQA (Tr) data sets. In this phase, the two data sets are combined together. The aim is to increase the QA skill of the system in general, or in other words, in an open domain. In the second phase, the model is trained with the updated parameters resulted from the previous fine tuning, by using the Banking Sector QA (Tr) data set. The second phase aims to increase the ability of the system to answer questions from a closed domain, the banking sector. Figure 8 shows an overview of the training phases together with the data sets they utilize.

3.3. System parameters

In the Pre-training task the following parameters are important to configure:

Maximum sequence length: Configures the maximum length of a sequence after the WordPiece tokenization. High values are necessary to learn positional embeddings in long sequences.

Maximum predictions per sequence and masked LM probability: when multiplied between them they define together the number of masked tokens in a sentence.

Do lower case and do whole word mask: Convert the sentences to lowercase and mask the tokens of a whole word instead of masking individual tokens, respectively. When these parameters are applied, the accuracy in general increases, especially for Asian languages. In the current study the vocabulary size is large enough to have most of the suffixed words in Turkish language. Consequently, the selection of an individual token masking or whole word masking doesn't alternate significantly the accuracy results.

Finally, for the fine-tuning task the important parameters to configure are the following ones:

Document stride parameter: This parameter allows to create training sentences examples that overlap between them. Within this way the next sentence training example will start in a given position in the previous sentence. This superposition is performed in a token level.

Maximum query length: The maximum number of tokens of a question. If a question is longer this number the rest will be truncated.

Maximum answer length parameter: The maximum length of an answer taken from the paragraph than belongs the related question. This parameter is character-based rather than token-based as it is the rest of the parameters.

3.4. Evaluation metrics

Similar to other machine reading comprehension and SQuAD studies, the exact match (EM) and F-Score will be used as the evaluation metrics. EM takes in account the predicted answer only if it is the same as the real answer and F-Score counts the predicted answers that they have an overlap with the real one.

4. Results

In this section, the results of the study are presented. In the beginning, the parameters that are achieving the best performance for answering questions for a specific domain, the banking sector are showed, followed by the errors types in errors and answers. Then, an evaluation of the study's model is performed as follows: first, the current model is compared with existing Turkish QA systems. Then its performance is evaluated in comparison with other BERT language models. Finally, the model is tested with other models in the Turkish language generated with BERT.

4.1. Neural network parameters evaluation

In order to better evaluate the accuracy of the system, different training parameters were tested. After experimenting with different values and considering the average and maximum length of paragraphs, questions, and answers in the data sets, it was observed that the following parameters are giving the best results for fine tuning: maximum sequence length = 512, document stride = 256, maximum question length = 64, and maximum answer length = 64.

From the results it can be seen that a long sequence length (512) with a half overlapping (stride set to 256) has an important impact on the accuracy results.

Table 3 presents different combinations and accuracy results in terms of EM and F-Score.

4.2. Error types in questions & answers

In order to evaluate the success of the system in a real-world environment, the team that prepared the Banking Sector QA data set was asked to think in general rather than focusing on a particular question type or formulation. The examination of the results, revealed that the system responds correctly to the majority of the factoid questions, where the answer to the question is a single fact. Since there is a distinct difference between the exact match (EM) and F-Score metrics, a 3.2% of wrong answers which have zero EM (572 out of 17708 answers) was identified. Table 4 presents the type of errors and Table 5 gives examples for most of these errors. The errors were categorized based on the following reasons:

1) Multiple possible answers: Questions that have multiple possible answers cause 30% of the errors, as seen in the first row of the table, both the answer and the prediction are logically correct replies to the question.

2) Questions requiring interpretation: For some answers, it is necessary to interpret the entire paragraph. For the moment, deep neural networks don't have this capability.

3) Conditional answer: In some questions, the answer varies according to the circumstances, as seen in the third example, the answer is ‘possible’ only for the customers of a brand.

4) Questions requiring a list of elements: These questions require an answer, which is a set of elements, or a list.

5) Answers with syntax variations: Since Turkish is an agglutinating language, generating one unique correct answer in terms of syntax is a difficult task. Hence, sometimes a correct answer given by the system can be enlarged or reduced when compared to the predicted answer, thus leading to a zero EM.

6) Incorrect question: There are also incorrectly prepared questions and answers in the data set. Some of them have many spelling mistakes, some of them are logically incorrect.

7) Incorrect answers: 25% of the questions are not answered correctly, generally for the type of questions that the system has not during the training phase or for very long questions and answers which are truncated during training.

4.3. Comparison with other Turkish QA systems

In this study the model of the current study was compared with Turkish QA systems in open and closed domain. In open domain, the system should be able to answer generic questions and in a closed domain, the system answers questions from a specific domain.

Table 3. Different parameter combinations and results for Banking Sector QA (Tr) data set.

Maximum Sequence Length (token)	Document Stride (token)	Maximum Query Length (token)	Maximum Answer Length (character)	EM	F-Score
512	256	64	64	54,09	79,01
512	256	64	30	52,72	78,69
512	128	64	64	52,00	77,46
512	512	128	64	51,94	75,66
512	384	256	30	49,33	74,39
384	64	64	30	47,56	73,42
256	64	64	30	46,01	71,97
128	64	64	30	44,38	70,11

Table 4. Description of wrong answers with zero EM (3.2%).

Error ID	Description	Counts
1	Multiple possible answers	180
2	Questions requiring interpretation	85
3	Conditional answers	80
4	Questions requiring a list of elements	5
5	Answers with syntax variations	34
6	Incorrect question	49
7	Incorrect answers	139
	Total	**572**

Table 5. Examples for error types 1-5.

Error ID	Question	Real answer	Predicted answer
1	Kampanya için gerekli şartlar nelerdir?	Bankanın müşteri yada kredi kartına sahibi olmak	Kampanyadan yararlanmak isteyen müşteriler, üyeliklerini 31/12/2019 tarihine kadar aktifleştirmelidir.
2	Tarihi geçmiş belgeler için müşteri ne yapmalıdır?	Şubeleri ile görüşmeleri gerekmektedir.	güncellenmelidir
3	Parmak izi ile girişi tüm müşteriler kullanabilir mi?	sadece X marka telefonu olanlar	kullanabilir
4	İşlemler ne zaman fona dönüşür?	09:15, 11:15, 13:15, 15:15	günlük
5	Kampanya kuponları hangi sitelerde geçerlidir?	www.x.com	www.x.com'da

Table 6. Comparison of Turkish QA Systems. (1) Data sets translated to Turkish. (2) Mean Reciprocal Rank, considers the rank of the first correct answer in the list of possible answers. (3) Who, Where, When and What. (4) Author, capital, birth date, death date, language of country, birth place, death place. (*) Phase 1: Fine tuning the model using merged SQuAD (Tr) and NewsQA (Tr) data sets. (**) Phase 2: Fine tuning the model, which is already trained with phase 1, using Banking Sector QA (Tr) data set.

Study	Data sets	Domain	Metric	Results
BayBilmiş	TREC-9[1] and TREC-10[1]	Open	MRR[2]	0,313
Automatic QA for Turkish with Pattern Matching	Only Specific Questions[3]	Closed	Precision	0,79 (Average)
A Factoid QA System Using Answer Pattern Matching	Only Specific Factoid Questions[4]	Closed	MRR[2]	0,73
Current Study	SQuAD (Tr)[1] and NewsQA (Tr)[1*]	Open	EM	55,26
			F-Score	67,07
	Banking Sector QA (Tr)[**]	Closed	EM	54,09
			F-Score	79,01

Table 6 presents the results of this comparison in terms of Exact Match and F-Score. The performance of the system was measured in open and closed domain by taking in account its accuracy in the first phase of the fine-tuning procedure with the SQuAD (Tr) and NewsQA (Tr) data sets and its accuracy in the second phase where the Banking Sector QA (Tr) data set was additionally utilized for training.

The studies in Table 6 proposed solutions for certain types of questions, and the accomplished success rates have been achieved in particular data sets. Because of different evaluation metrics and test data sets, it is difficult to directly compare the results of the current study with the results of the previous QA systems in Turkish. But in general, based on the evaluation metrics (Mean reciprocal rank), and the type of data sets (question specific data sets) of these studies, it can be considered that the current study proposes a solution that is applicable to a wider range of QA's. The proposed method covers all types of questions, and there is no restriction in terms of question types or text for the evaluation data sets.

4.4. Comparison with other BERT language models

In order to evaluate the selected data sets and training procedure the current model was compared with models that used data sets in other languages. Table 7 presents the comparison with BERT models in the Arabic and Chinese Language in terms of EM and F-Score. Existing BERT models in the Persian, Korean and French language didn't present results for QA tasks.

Table 7. Comparison of BERT models with other languages. (1) Data sets translated to Turkish. (2) Multilanguage model published by Google. (3) Arabic Reading Comprehension Dataset, which was previously translated from SQuAD to Arabic. (4) Chinese specific model published by Google. (*) Phase 1: Fine tuning the model using merged SQuAD (Tr) and NewsQA (Tr) data sets.

Language	Model	Data sets	Domain	EM	F-Score
Arabic	mBERT	ARCD[3]	Open	34,2	61,3
	AraBERT	ARCD[3]		30,6	62,7
Chinese	BERT-Chinese[4]	CMRC		18,6	43,3
		DRCD		82,2	89,2
		CJRC		55,1	75,2
Turkish	Current Study	SQuAD (Tr) [1]		57,60	68,34
		NewsQA (Tr) [1]		48,01	59,86
		SQuAD (Tr)[1] and NewsQA (Tr)[1*]		55,26	67,07

Table 8. Comparison of Turkish base models. (1) Data sets translated to Turkish. (*) Phase 1: Fine tuning the model using merged SQuAD (Tr) and NewsQA (Tr) data sets. (**) Phase 2: Fine tuning the model, which is already trained with phase 1, using Banking Sector QA (Tr) data set.

Data sets	Model	Domain	EM	F-Score
SQuAD (Tr)^1 and NewsQA (Tr)^{1*}	BERTurk	Open	57,43	69,36
	Current Study		55,26	67,07
	mBERT		54,52	65,74
Banking Sector QA (Tr)**	BERTurk	Closed	55,89	80,87
	Current Study		54,09	79,01
	mBERT		50,74	77,03

4.5. Comparison with other BERT Turkish models

The results show that the performance of this system is, most of the time, better than other existing models. The current Turkish language model was compared with two others, Google's multilingual version of BERT model, mBERT and another model in the Turkish language entitled BERTurk (Schweter, 2020). In Table 8, it can be observed that this model is better than mBERT and slightly inferior but still comparable with the BERTurk model (around 2% difference). The selection of larger and various data sets for the training procedure is one of the main reasons that generate those differences.

5. Discussion

In this work, the evaluation of a QA system was done in open and closed domains, based on a deep neural network that generates a model language with a context-sensitive vocabulary encoding. The results revealed the following findings:

In Turkish QA systems, deep neural network methodologies can cover a wider domain compared to other approaches. The results revealed that almost all other methods based on a semantic or rule-based approach are successful in specific types of QAs. This finding is expected and in general it is applicable to other fields apart from the NLP domain.

Language training data sets play a key role. The language model plays a significant role in the success of a QA system and the data sets used for generating have a major contribution. BERTurk performed slightly better because it used a wider training corpus. Special effort should be given by providing adequate data sets.

Translated data sets are adequate for fine tuning. The automatic translation of English data sets, even if they need a preprocessing step, gave high accuracy results during the fine tuning. According to authors' experience, this method can be applied to other languages and also other variations of deep neural networks based on bi-directional transformers.

The above finding let conclude that the proposed methodology can be successfully used in QA tasks for any language and any domain. Based on this experience, QA tasks can be carried out in an open or any closed domain, provided that existing data sets in English will be translated with a preprocessing procedure and an adequate data set will be generated for a particular domain. Generating a data set in SQuAD format for a closed domain even if it is time-consuming and requires human resources, it is still necessary for obtaining a closed domain QA system with high accuracy. Today, researchers are focused on increasing the accuracy of generic QA tasks in order to reduce the performance gap between open and closed domains. As a result, new variations of neural networks using bi-directional transformers like BERT are proposed, and new data sets are emerging for training purposes.

6. Conclusion

This study presented a QA system in the Turkish language for the banking domain. This approach required the use of various and large data sets. Even if BERT cannot solve the morphological disambiguation problem of Turkish, the overall architecture is sufficient for solving the word sense disambiguation problem. For QA tasks in open domain, a translation and preprocessing step of some data sets was necessary and for answering questions in the banking domain, the generation of a new data set was required. The experiments showed that the accuracy of the network can significantly vary according to the choice of the training parameters. To the best of our knowledge, this study is the first that proposes a framework in the Turkish language for a QA task in open and also in closed domain using deep neural networks. Additionally, the proposed methodology is applicable to any language and to any domain for performing QA tasks.

Acknowledgements

We would like to thank Prof. Dr. Kemal Oflazer for sharing the News Corpus (Tr), described in Section 3.1 and used in the pre-training task and the QA tasks for open and closed domain were generated using BERT neural network. We also thank TEB - Turk Ekonomi Bankasi and TEBArf for their support in generating the Banking QA data set and funding the TPU servers used in the study.

References

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint 2018, arXiv: 1810.04805.

Çelebi, E., Günel, B., Şen, B., 2011. Automatic question answering for Turkish with pattern parsing. INISTA 2011 - 2011 International Symposium on INnovations in Intelligent SysTems and Applications, pp. 389–393. https://doi.org/10.1109/INISTA.2011.5946098

Amasyalı, M.F., Diri, B., 2005. Bir soru cevaplama sistemi: Baybilmiş. Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi, 1(1).

Er, N.P., Cicekli, I., 2013. A factoid question answering system using answer pattern matching. Proceedings of the Sixth International Joint Conference on Natural Language Processing, pp. 854–858.

Biricik, G., Solmaz, S., Özdemir, E., Amasyalı, M.F., 2013. A Turkish Automatic Question Answering System with Question Multiplexing: Ben Bilirim. International Journal of Research in Information Technology (IJRIT) 1(6), 46–51.

Akın, A.A., Akın, M.D., 2007. Zemberek, an open source nlp framework for turkic languages. Structure, 10, 1–5.

Oflazer, K., Say, B., Hakkani-Tür, D.Z., Tür, G., 2003. Building a Turkish Treebank. Springer, pp. 261–277. https://doi.org/10.1007/978-94-010-0201-1_15

Eryiğit, G., Oflazer, K., 2006. Statistical dependency parsing for turkish, in: 11th Conference of the European Chapter of the Association for Computational Linguistics.

Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L., 2018. Deep contextualized word representations. NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, pp. 2227–2237. https://doi.org/10.18653/v1/n18-1202

Radford, A., Salimans, T., 2018. Improving Language Understanding by Generative Pre-Training. OpenAI, 1–12.

Fisch, A., Talmor, A., Jia, R., Seo, M., Choi, E., Chen, D., 2019. MRQA 2019 Shared Task: Evaluating Generalization in Reading Comprehension, pp. 1–13. https://doi.org/10.18653/v1/d19-5801

Antoun, W., Baly, F., Hajj, H., 2020. AraBERT: Transformer-based Model for Arabic Language Understanding. arXiv preprint 2020, arXiv:2003.00104.

Martin, L., Muller, B., Suárez, P.J.O., Dupont, Y., Romary, L., de la Clergerie, É.V., Seddah, D., Sagot, B., 2019. CamemBERT: a Tasty French Language Model. arXiv preprint 2019, arXiv:1911.03894.

Farahani, Mehrdad, Gharachorloo, M., Farahani, Marzieh, Manthouri, M., 2020. ParsBERT: Transformer-based Model for Persian Language Understanding. arXiv preprint 2020, arXiv:2005.12515.

Cui, Y., Che, W., Liu, T., Qin, B., Yang, Z., Wang, S., Hu, G., 2019. Pre-Training with Whole Word Masking for Chinese BERT. arXiv preprint 2019, arXiv:1906.08101.

Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P., 2016. SQuad: 100,000+ questions for machine comprehension of text. EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings, pp. 2383–2392.

Trischler, A., Wang, T., Yuan, X., Harris, J., Sordoni, A., Bachman, P., Suleman, K., 2016. Newsqa: A machine comprehension dataset. arXiv preprint 2016, arXiv:1611.09830.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Advances in Neural Information Processing Systems, 5999–6009.

Wiedemann, G., Remus, S., Chawla, A., Biemann, C., 2019. Does BERT Make Any Sense? Interpretable Word Sense Disambiguation with Contextualized Embeddings. arXiv preprint 2019, arXiv:1909.10430.

Wu, Y., Schuster, M., Chen, Z., Le, Q. v., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, Ł., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint 2016, arXiv: 1609.08144.

Schweter, S., 2020. BERTurk - BERT models for Turkish. https://doi.org/10.5281/zenodo.3770924