Expert Systems With Applications 195 (2022) 116592

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Natural language based analysis of SQuAD: An analytical approach for BERT

Zekeriya Anil Guven [*,1], Murat Osman Unalir [2]

Department of Computer Engineering, Ege University, Izmir, Turkey

ARTICLE INFO

Keywords:
Natural language processing
BERT
Text analysis
Question answering
SQuAD

ABSTRACT

In recent years, deep learning models have been used in the implementation of question answering systems. In this study, the performance of the question answering system was evaluated from the perspective of natural language processing using SQuAD, which was developed to measure the performance of deep learning language models. In line with the evaluations, in order to increase the performance, 3 natural language based methods, namely RNP, that can be used with pre-trained BERT language models have been proposed and they have increased the performance of the question answering system in which the pre-trained BERT models are used by 1.1% to 2.4%. As a result of the application of RNP methods with sentence selection, an increase in accuracy between 6.6% and 8.76% was achieved in answer detection. Since these methods don't require any training process, it has been shown that they can be used in question answering systems to increase the performance of any deep learning model.

1. Introduction

Natural Language Processing (NLP) is a research and application area that explores how computers can be used to understand and process natural language text or speech. NLP researchers aim to gather information on how humans understand and use language. Thus, appropriate tools and techniques can be developed to enable computer systems to understand and manipulate natural languages to perform the desired tasks (Chowdhary, 2020). NLP areas can be divided into two broad subareas: core and application areas (Otter et al., 2018). Core areas examine key issues such as language modeling, which highlights quantitative relationships among words. Morphological processing, syntactic parsing and semantic processing are defined as core area's operation. Application areas include extraction of useful information like named entities and relationships, translation of text between languages, document summarization, question answering (QA) (McCann et al., 2018) as automatic by inference, sentiment analysis (Gokalp et al., 2020; Güven et al., 2020) and document classification and clustering.

Many machine learning (ML) methods are performed for NLP. The general purpose of ML is to recognize patterns in data that inform how unseen problems are treated. ML methods are given to the model according to the extracted features. ML tasks are generally divided into three broad categories, depending on the nature of the learning "signal" or "feedback" available for a learning system. These are supervised, unsupervised, and reinforcement learning (Ballı and Sağbaş, 2018). Supervised learning learns a function that maps an input to an output based on sample input–output pairs. Unsupervised learning is a class of learning problems where input data are obtained as in supervised learning, but without labels. The goal of learning here is to recover some fundamental and possibly not trivial structure in the dataset. Reinforcement learning is a field of ML where an agent takes actions in an environment with the aim of maximizing a reward (Carleo et al., 2019). Deep Learning (DL), which is a sub-area of machine learning, has driven the quick progress in artificial intelligence. It is also leading to surprising breakthroughs on long-standing problems in many areas such as NLP. Tools supported by deep learning play an increasing role in the making of movies and in understanding and communicating with humans. This kind of development is possible with deep learning frameworks such as Caffe (Jia et al., 2014), PyTorch (Paszke et al., 2019), TensorFlow (Abadi et al., 2016), etc. These frameworks have been effective in spreading ideas in this area (Guo et al., 2019).

When new data is given to the DL model, this model needs to be

* Corresponding author.
E-mail addresses: zekeriya.anil.guven@ege.edu.tr (Z.A. Guven), murat.osman.unalir@ege.edu.tr (M.O. Unalir).
[1] 0000-0002-7025-2815.
[2] 0000-0003-4531-0566.

https://doi.org/10.1016/j.eswa.2022.116592
Received 7 June 2021; Received in revised form 18 September 2021; Accepted 19 January 2022
Available online 31 January 2022

retrained. This operation constitutes problems in terms of cost and time. Marcus (2020) stated that when an irrelevant new sentence is added to any article in the Stanford Question Answering Dataset (SQuAD)[3], which consists of Wikipedia articles, questions and answers specific to the articles created by Stanford University, used in the DL language model, the model answered the question incorrectly, so this model could not detect it. In this paper, we propose an NLP-based QA System (NLP-QAS) to solve this problem. The NLP-QAS includes proposed 3 NLP methods (Remove and Compare, Searching with NER, Searching with POS tagging), namely RNP, for applying on sentence selection and pre-trained DL language models such as BERT. The RNP methods have been developed using textual techniques such as Part of Speech (POS) tagging and Named Entity Recognition (NER). The SQuAD has been chosen to use these methods. Firstly, the question terms are analyzed by searching the sentences in the related paragraph on SQuAD and the most appropriate sentence is selected. The RNP methods have been performed to the chosen sentence for answer detection. Then, these methods are used on pre-trained BERT models. It is thought that RNP methods can answer the questions whose answers can't be detected with pre-trained BERT models.

The main contributions of our study to the literature are as follows:

- An NLP-QAS has been proposed including sentence selection and NLP-based methods for answer detection.
- To the best of our knowledge, this study is the first one that extends the BERT language model's capability with RNP methods in QA systems.
- Experimental results show that the proposed NLP-QAS and RNP methods could outperform on questions that the BERT model couldn't answer.

The remainder of the paper is structured as follows. Literature research of the BERT model and SQuAD are explained under the heading "Related Work" in Section 2. Section 3 describes research's methodology containing dataset, BERT model, data preprocessing, NLP-QAS and RNP methods. The analysis of the article terms in SQuAD, sentence extraction containing answers by examining question, analysis of answer detection and analysis of BERT with RNP methods are described in Section 4. Finally, Section 5 indicates the conclusion of this paper and discusses possible future studies.

2. Related work

QA provides answers to questions posed in natural language. Since the same information can be expressed differently in natural language. Consequently, small variations in semantically equivalent questions can produce different answers (Dong et al., 2017). Hence, QA is a difficult NLP area. Enabling computers to automatically answer questions in natural language in any topic has been the focus of many studies in recent years. ML and DL methods are used in most studies. Zhou et al. (2018), propose a recurrent convolutional neural network (RCNN) for answer selection in community question answering (CQA), which is an online QA website driven by a community. It combines convolutional neural networks (CNN) with RNN to capture both the semantic matching between question and answer and the semantic correlations embedded in the sequence of answers. Results show that RCNN can improve over the baseline model. Martinez-Gil et al. (2019) proposed a new method for automatic answering of multiple choice questions This method is reinforced co-occurrence that aims to discover latent patterns on a large corpus of texts. As a result of an empirical evaluation applied on a dataset of legal questions, they showed the positive contribution of the proposed method Esposito et al. (2020) proposed a hybrid Query Expansion approach based on lexical sources and word embeddings to improve fetching related sentences from documents. First, they obtained the synonyms and hypernyms of the relevant terms in the question from MultiWordNet and contextualized them with the collection of documents used. Finally, with a semantic similarity metric built on top of Word2Vec, the resulting set is sorted and filtered based on the wording and the meaning of the question. Yeh and Chen (2020) proposed an alternative approach for the Q&A system called "QAInfomax", which aims to help models avoid getting stuck with superficial biases in the data during learning. For this, they maximized mutual information among passages, questions, and answers. Their proposed QAInfomax achieved state-of-the-art performance in the AdversarialSQuAD dataset without additional training data. Literature research of the BERT model and SQuAD used in the QA are given under separate subheadings.

2.1. SQuAD

Many language representation models have been used in the literature to measure the success of SQuAD. Devlin et al. (2019) proposed BERT model for NLP tasks. When Devlin et al. (2019) analyzed the success rate of BERT, they obtained %93.2F-measure for SQuAD 1.1 and %83.1F-measure for SQuAD 2.0. Zhang et al. (2019), added explicit syntactic constraints to the mechanism for better linguistic word representation. They adopt pre-training dependent syntactic parsing tree structure to generate corresponding nodes for each word in a sentence. Their aim was to design an effective neural network model that uses grammar as effectively as possible. To verify the effectiveness of the structure, the syntax driven network (SG-Net) has been applied to the pre-trained typical language model BERT. For the SQuAD 2.0 dataset, they obtained better results than the success of BERT (%83.1). They found the f-measure of the system to be %87.9. Zhang et al. (2019), to ensure natural language understanding, proposed developing a language representation model called Semantics-aware BERT (SemBERT), including explicit contextual semantics. SemBERT is a fine-tuned BERT model. Its structure consists of three components: the semantic role labeller, a set of encoders, and the semantic integration component. They used pre-trained BERT weights and followed the same fine-tuning procedure as BERT without any changes. The model was evaluated in 11 comparison datasets including natural language inference, question answering, semantic similarity and text classification. When the results for SQuAD 2.0 were examined, %87.9f-measure value was achieved. SemBERT has increased the BERT value in both exact match and f-measure. Since symbols such as "[MASK]" used by BERT during pre-training lacked real data at the post-processing time and caused fine-tuning inconsistency, Yang et al. (2019) proposed XLNet, a generalized Autoregressive (AR) pre-training method that solves this problem. XLNet has made the most of both AR language modeling and auto-encoder. The model enables learning two-way contexts by maximizing the expected probability on all permutations of the factorization. They suggested using two latent representation sets, content and query, in its structure. Content and token are encoded in content representation. In the query representation, contextual information and location are verified. They made comparisons in many areas to evaluate the model. For the SQuAD, they compared the BERT and RoBERTa model with XLNet. They achieved %95.1f-measure for SQUAD 1.1 and %90.6 for SQuAD 2.0. They have shown that XLNet gives the best performance of all other models.

2.2. BERT model

The BERT model is used in most areas such as QA, topic modelling and sentiment analysis. Peinelt et al. (2020) proposed tBERT, a simple architecture that combines subjects with BERT for semantic similarity prediction. They showed that tBERT provides improvements in multiple semantic similarity prediction datasets versus a finely tuned vanilla BERT. Sun et al. (2019) propose a new solution of aspect-based sentiment analysis by converting it to a sentence-pair classification task. They

[3] https://github.com/rajpurkar/SQuAD-explorer/tree/master/dataset

Table 1
The statistics of SQuAD versions (Rajpurkar vd., 2018).

		SquAD 1.1	SquAD 2.0
Train	Total examples	87,599	130,319
	Negative examples	0	43,498
	Total articles	442	442
	Articles with negatives	0	285
Development	Total examples	10,570	11,873
	Negative examples	0	5945
	Total articles	48	35
	Articles with negatives	0	35
Test	Total examples	9533	8662
	Negative examples	0	4332
	Total articles	46	28
	Articles with negatives	0	28

fine-tuned the pre-trained BERT model and achieved on SentiHood and SemEval-2014 Task 4 datasets. Qu et al. (2019) proposed a conceptually simple but highly effective approach called history answer embedding. They adopt a rule-based method for history selection. They have ensured the seamless integration of conversation history into a conversational QA model based on BERT. Adhikari et al. (2019), have achieved state-of-the-art results for document classification by fine-tuning the BERT. They have also shown that BERT can be decomposed to a much simpler neural model that provides competitive accuracy at a much more modest computational cost. Al-Garadi et al. (2021), proposed a BERT-based model to improve classification performance in prescription medication (PM) abuse classification and compared the success of the proposed model to deep learning, BERT-like models using a public Twitter PM abuse dataset. They discussed empirical analysis of BERT-based models and their advantages and disadvantages for application in social media text classification in general and PM abuse detection in particular.

3. Methodology

3.1. Dataset description

The SQuAD is created by Stanford University for the QA task. SQuAD consists of two versions, 1.1 and 2.0. SQuAD 1.1 is a reading comprehension dataset that contains answers to each question in Wikipedia articles. It contains 23,215 paragraphs and 107,785 question–answer pairs for 536 articles obtained from Wikipedia. The answers aren't multiple choice, there is only one answer. The answers to the questions were extracted by the crowdworkers in their own words (Rajpurkar et al., 2016).

SQuAD 2.0 contains the same articles and question–answer pairs in SQuAD 1.1. In addition to this data, a total of 53,775 questions without answers for the same articles were added to this version (Table 1). The unanswered questions were created by using the opposite of words, making changes on the numbers, and performing negation processes on the questions whose answers were known. The purpose of using this dataset is not only to answer the questions, but also to avoid answering questions that have no answers (Rajpurkar et al., 2018). An example of an article in SQuAD 2.0 is shown in Fig. 1. The figure contains a paragraph for a sample article, questions and answers for this paragraph. The actual (ground truth answers) and predicted (prediction) answers for the questions are listed. The questions that are answered incorrectly (red bar) and correctly (green bar), and the question without an answer (<No answer >) are also shown.

Both datasets are divided into random training (80%), development (10%) and test (10%) sets. The test set is hidden as it is only used on the SQuAD benchmark platform. Statistics of both datasets are shown in Table 1. Table 1 indicates that some articles of the development and test set have been removed from SQuAD 2.0. In this study, SQuAD 2.0 version is analyzed and used for answer detection analysis. All operations in the NLP-QAS are only performed on the questions with answers.

3.2. BERT language model

The language model (LM) is the probability distribution over word strings. LM provides context for distinguishing similar words and sentences. It is divided into unidirectional and bidirectional LM. Unidirectional LM assigns a probability by factorization sequence when given the input sequence. Transformer (Vaswani et al., 2017) can be given as an example for unidirectional LM. Transformer relies solely on attentional mechanisms, giving up recurrence and convolutions completely. The transformer architecture scales with training data and model size, provides parallel training easier, and captures long-range sequence features (Wolf et al., 2019). Transformer is able to use a longer history by caching previous outputs and using relative position. Bidirectional LM assigns a probability to the array using the input sequence, position, and left–right context of the word. ELMo and BERT are examples of this model. ELMo suggests working with the forward feed and backpropagation LSTM to predict the probability. ELMO uses multiple LSTM layers (Petroni et al., 2020).

BERT is defined as a transformer-based bidirectional encoder representation. BERT produces multiple, contextual, bidirectional word representations. BERT proposes a new train goal with the masked language model (MLM) method. MLM randomly masks some of the tokens in the input. Its purpose is to predict only the masked word based on its context. There are two stages in the structure of BERT; pre-training and fine-tuning. During pre-training, the model is trained with unlabeled data on different pre-training tasks. In the fine-tuning phase, the BERT model is first started with pre-trained parameters. Then, all parameters are fine-tuned using labeled data from downstream tasks. BERT has two model sizes as BERT-Base (12 layers, 768 hidden dimensions and 12 attention heads with 110 M total number of parameters) and BERT-Large (24 layers, 1024 hidden dimensions and 16 attention heads with 340 M total number of parameters) (Devlin et al., 2019). Most language models have been generated using the BERT model. So most of the models are based on BERT. ALBERT (Lan et al., 2019), SemBERT (Zhang

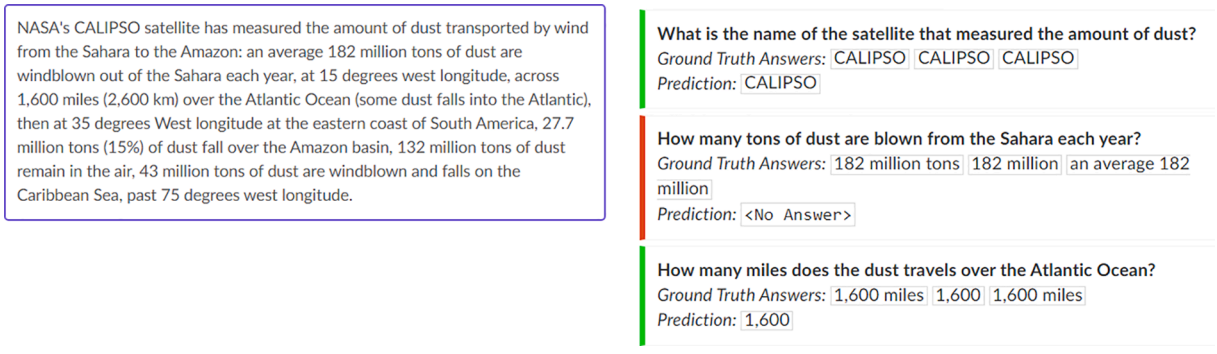

Fig. 1. An example for SQuAD 2.0 (SQuAD, 2021).

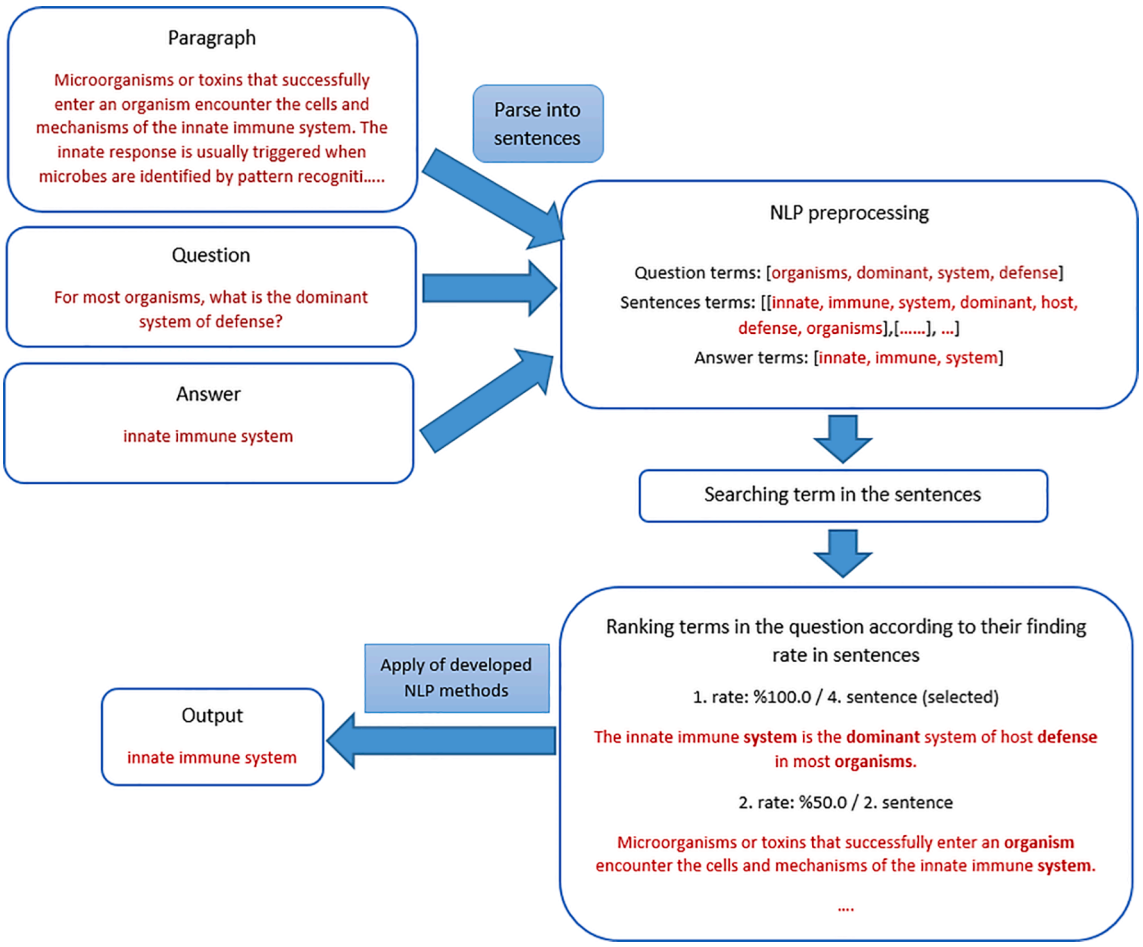

Fig. 2. Analysis structure of the NLP-QAS for SQuAD.

et al., 2019), RoBERTa (Liu et al., 2019), SG-Net (Zhang et al., 2019) can be given as extensions to these models.

The pre-trained BERT-Base and BERT-Large language models have been selected for the analysis in this study. While BERT-Base training was performed on 4 Cloud TPUs (16 TPU chips in total), BERT-Large training was performed on 16 Cloud TPUs (64 TPU chips in total) in Pod configuration. The sequence length was limited to 128 tokens for 90% of the steps and 512 for the remaining 10%. The optimizer used is Adam with a learning rate of 1e-4, $\beta 1=0.9$ and $\beta 2 =0.999$, a weight decay of 0.01, learning rate warmup for 10,000 steps and linear decay of the learning rate after (Devlin et al., 2019). The BERT-Large model has also been trained with a new technique, which is called Whole Word Masking. In this case, all of the tokens corresponding to a word are masked at once. The overall masking rate remains the same.

3.3. Data preprocessing

Specific preprocesses should be performed on the SQuAD for more successful answer detection. Preprocess prevents unnecessary data from being used. The performed preprocesses are as follows:

- Punctuation marks are removed.
- The texts are converted to lowercase. Thus, case complexity is resolved.
- Texts are parsed into tokens. Duplicate tokens are removed to create unique tokens. Thus, the data volume is decreased.
- Tokens are filtered through stopwords. Finally, unnecessary data is removed.
- Proper names such as place, person and time are found in the articles with the NER technique. After the determination of proper names, the terms that appear separately are combined and used as a single term ([Didier], [Drogba] - [Didier Drogba]).
- Lemmatization technique is applied on words. The same words with different forms were obtained in one form by lemmatization. Lemmatization is also analyzed together with POS tagging. Lemmatization is used optionally.

3.4. NLP-based QA system architecture

In order to answer detection, question terms are analyzed at the sentences of each paragraph. Before the analysis, the paragraphs are split into sentences. Afterwards, preprocessing techniques are performed to the question terms and sentences.

In this study, an NLP-QAS is proposed for answer detection. Firstly, sentence selection has been performed and the success of the RNP methods have been analyzed for answer detection in this sentence at the NLP-QAS. An example of the NLP-QAS operation is shown in Fig. 2. The question, paragraph and answer are taken as input, and operations

```
initialization SQuAD dataset;
reading titles, paragraphs, questions, answers lists;
for i in range(len(titles)):
    title ← titles[i];                                              // Article's title
    questionList ← questions[i];                                    // Title's question list
    for j in range(len(questionList)):                              // Reading the paragraph's questions
        for k in range(len(questionList[j])):
            paragraph ← paragraphs[j][k];                           // Reading paragraph
            question ← questionList[j][k];                          // Reading question
            answer ← answers[j][k];                                 // Reading answer
            if preprocess = TRUE:                                   // Preprocess is selected
                question ← PreProcess(question);                    // Preprocess function
                questionTerm ← create question term list according to question;
            else:
                question ← perform just punctuation to question;
                questionTerm ← create question term list according to question;
            if len(answer) ≠ 0:
                sentences ← SearchTerms(paragraph, question);       /* searching question
                 terms in sentences,sorting sentences according to QTP */
                if len(sentences) ≠ 0:
                    for l in range(len(sentences)):
                        if sentences[l] has answer:
                            AnswerDetection(sentences[l], answer, question);  /* Sending values
                             to the function in order to answer detection with
                             developed methods */
                            break;
                else:                                               // Examine sentences without QTP
                    sentences ← paragraph's all sentences;
                    for m in range(len(sentences)):
                        if sentences[m] has answer:
                            AnswerDetection(sentences[m], answer, question);
```

Fig. 3. Pseudocode of sentence selection.

```
runningMethod ← running method's number;
question,actualAnswer,sentence ← related terms for methods;
detectedAnswer ← 0;                                                 // Detected answer count
if runningMethod = 1:                                               // RC Method
    questionTerms,answerTerms,sentenceTerms ← Performed preprocess;
    stopwords ← load NLTK stopwords list;
    for i in length(questionTerms):                                 // Removing question terms from sentence
        if sentenceTerms has questionTerms[i]:
            Removing questionTerms[i] in the sentenceTerms
    for i in length(stopwords):                                     // Removing stopwords from term lists
        if sentenceTerms has stopwords[i]:
            Removing stopwords[i] in the sentenceTerms
        if answerTerms has stopwords[i]:
            Removing stopwords[i] term in the answerTerms
    if length(sentenceTerms) = length(answerTerms):                 // Lists' term count is equal
        sentence ← create sentence with sentenceTerms list;
        answer ← create answer with answerTerms list;
        if sentence = answer:                                       // sentence and answer are equal
            detectedAnswer ← detectedAnswer + 1 ;                   // Answer is detected
```

Fig. 4. Pseudocode of RC method.

```
if runningMethod = 2:                                                    // SNER Method
    anwerTag ← Determine NER tag according to question pronoun;
    if question has 'who' pronoun:                                       // who indicates person
        answerTag ← PERSON
    if question has 'when' pronoun:                                      // when indicates time
        answerTag ← DATE
    /* There are many answerTag values(GPE,QUANTITY,etc.)  according to
    question pronoun:where,how,etc.  So If condition count are more   */;
    if answerTag ≠ ∅:
        answer ← ∅;
        sentenceNER ← NER dictionary(key,value) of sentence;
        for i ← 0 in length(sentenceNER):
            if answerTag = sentenceNER.key: // Sentence's NER tag and answer tag is
             equal
                answer ← sentenceNER.value
        if actualAnswer = answer:                    // Answer and actual answer is equal
            detectedAnswer ← detectedAnswer + 1 ;                // Answer is detected
```

Fig. 5. Pseudocode of SNER method.

```
if runningMethod = 3:                                                    // SPOS Method
    posQuestion ← Split into POS tag(word,tag) of question ;
    if posTag.tag has WRB:                                  // Pos tag indicates wh- adverb
        if posQuestion.word = when:
            answerTag ← CD
        if posQuestion'.word = where:
            answerTag ← NN
    if posQuestion.tag has WP$:          // Pos tag indicates possessive wh-pronoun
        if posQuestion'.word = whose:
            answerTag ← NNP
    /* There are many answerTag values(IN,NN, etc.)  according to pos
    tag:NNP,WP,etc.  So If condition count are more           */;
    if answerTag ≠ ∅:
        answer ← ∅;
        sentence ← Removing question terms from sentence;
        posSentence ← Split into POS tag(word,tag) of sentence;
        index ← 0 ;
        for i in length(posSentence):
            if answerTag = posSentence[i].tag:
                answer ← posSentence[i].word + ' '
            if index + 1 ¡ length(posSentence) − 1:        // create answer as incremental
                for j ← index+1 in length(posSentence):
                    if answerTag = posSentence[j].tag:
                        answer ← posSentence[j].word + ' '
                    else:
                        break;
                break;
            index ← index + 1
        if actualAnswer = answer:                    // Answer and actual answer is equal
            detectedAnswer ← detectedAnswer + 1;                 // Answer is detected
```

Fig. 6. Pseudocode of SPOS method.

between question and answer is analyzed with RNP methods. Firstly, the question and the sentences of the related paragraph are preprocessed and parsed into tokens. Question terms are compared with the terms of each sentence. The rate of question terms found in that sentence is called question term percentage (QTP). The QTP of each sentence is sorted in the descending order. The sentence with the highest rate containing the answer is selected for answer detection. Finally, the answer is searched with the RNP methods on the chosen sentence and the answer found by the RNP methods is determined as the possible answer.

The pseudocode for the sentence selection of the NLP-QAS is indicated in Fig. 3. As stated in the pseudocode, operations are performed only on questions that contain answers. As shown in Fig. 3, for the

Table 2
An example for RNP methods.

METHOD	QUESTION	ANSWER	ANSWER DETECTION
RC	In what unit is the size of the input taken?	bits	Sentence: This is usually taken to be the size of the input in bits. Removing of question terms and stopwords New sentence: **bits**
SNER	In what country is Normandy located? Answer type: GPE (what country => GPE)	France	Sentence: The Normans (Norman: Nourmands; French: Normands; Latin: Normanni) were the people who in the 10th and 11th centuries gave their name to Normandy, a region in France. NER types: [('Normans', 'NORP'), ('French', 'NORP'), ('Latin', 'NORP'), ('Normanni', 'PERSON'), ('the 10th and 11th centuries', 'DATE'), ('Normandy', 'ORG'), **('France', 'GPE')**, ('a', 'DT'), ('fiefdom', 'NN'), (',', ','), ('established', 'VBN'), ('by', 'IN')
SPOS	When was the Duchy of Normandy founded? Answer tag: CD (when(WRB) => CD)	911	Question POS tags: [('when', 'WRB'), ('was', 'VBD'), ('the', 'DT'), ('duchy', 'NN'), ('of', 'IN'), ('normandy', 'NN'), ('founded', 'VBN'), ('?', '.')] Sentence: The Duchy of Normandy, which began in 911 as a fiefdom, was established by the treaty of Saint-Clair-sur-Epte between King Charles III of West Francia and the famed Viking ruler Rollo, and was situated in the former Frankish kingdom of Neustria. Removing question terms; New sentence POS tags: [('The', 'DT'), (',', ','), ('which', 'WDT'), ('began', 'VBD'), ('in', 'IN'), **('911', 'CD')**, ('as', 'IN'), ('a', 'DT'), ('fiefdom', 'NN'), (',', ','), ('established', 'VBN'), ('by', 'IN'), ('treaty', 'NN'), ('of', 'IN'), ...]

paragraphs in all articles, firstly, the question sentence is decomposed into terms according to the preprocess selection (line no. 11–16). Then, the question terms are searched within the sentences in the paragraph and the sentence containing the answer among the sentences ordered according to the QTP is selected for answer detection (line no. 18–23). However, if the sentence is not found according to QTP, the sentence containing the answer in the paragraph is selected for answer detection (line no. 24–28).

3.5. NLP-QAS methods

Natural language based methods are proposed for more successful answer detection in the NLP-QAS. The RNP methods are developed by NLP techniques like NER, POS, string processes.

The RNP methods have been used for answer detection in the chosen sentence as a result of QTP. Three RNP methods have been proposed. These three methods are as follows:

1 Remove and compare (RC): The question terms are removed from the chosen sentence (line no. 7–9). Then stopwords from both chosen sentence and answer are removed (line no. 10–14). If term count in chosen sentence is equal to the term count in answer, the remaining terms in the chosen sentence are combined into sentences (line no. 15–17). This sentence is compared with the actual answer (line no. 18–19). If the sentence and actual answer are equal, the answer is detected as true. The pseudocode of this method is indicated in Fig. 4.

2 Searching with NER (SNER): The NER statements are used for answer detection. NER finds expressions such as person, place and time in a sentence. The entity types obtained from the NER are used in this method (Annotation Specifications, 2021). Firstly, question pronouns are sought in the question sentence. The most appropriate NER entity type (PERSON, DATE, etc.) is selected for answer detection according to the question pronoun (line no. 2–6). Then, it is checked whether there is this entity type in NER statements of the chosen sentence. If this sentence contains this entity type, the term of this entity type is selected as the possible answer (line no. 8–12). Finally, the possible and actual answers are compared and if both are equal, the possible answer is correct (line no. 13–14). The pseudocode of SNER method is indicated in Fig. 5.

3 Searching with POS tagging (SPOS): The POS tagging is used for the last method. Special tags belonging to the word are determined in POS tagging (Rachiele, 2018). Answer tag is determined according to the POS tag of the question pronoun (line no. 2–10). After the question terms are removed from the chosen sentence, the new sentence is parsed into POS tags (line no. 13–14). If there is a tag with the answer tag in this sentence, the term of this tag is selected as the possible answer. But in the following term, it is checked to see if it has the same tag. If there is a term with the same tag, the term is added to the possible answer (line no. 16–26). Then, possible and actual answers are compared. If they are both equal, the possible answer is correct (line no. 27–28). The pseudocode of the SPOS method is indicated in Fig. 6.

An example for RNP methods is shown in Table 2. Table shows that the answers are detected by these methods. In the Appendix, the extended version of Table 2 is given for each method in Table 19, Table 20 and Table 21 respectively.

4. Experiments

The analysis of the SQuAD and the performed NLP-QAS are explained under different headings in this section. Preprocessed terms, question distribution and the distribution of the question pronouns on SQuAD were analyzed in the first subheading. The second subheading describes analysis of sentence selection containing answers according to QTP. The analysis of answer detection was performed with RNP methods on the chosen sentences in the third subheading. Last subheading describes the analysis of BERT with RNP methods.

In this study, preprocessing techniques has been used optionally and has been analyzed for all stages.

4.1. Analysis of dataset

Firstly, the words in all articles have been analyzed for SQuAD. The term count that can be used for operations have been obtained by removing the stopwords in each article. The statistics containing term rates for the articles are shown in Fig. 7. When the stopwords are removed, the term count that can be used for the Dev_set has decreased to 86.3%. Thus, the data volume to be processed decreased by 13.7%. In addition, the term count that can be used for the Train_set is 86.5%. Since it is not appropriate to represent 442 articles (the training set: Train_set) in the figure, only the 35 articles in the development set (Dev_set) is shown in Fig. 7 for SquAD.

The question statistics in SQuAD are shown in Table 3. When the distribution of questions with and without answers is analyzed, the Dev_set has an uniform distribution, but the distribution of the Train_set hasn't. All operations have been performed on the questions containing answers.

The distribution of the question pronouns have been analyzed. It is aimed to detect the answer correctly by determining the answer type expressed by the questions (who: person, when: time, where: place, how many / much: quantity). Table 4 indicates the distributions of the

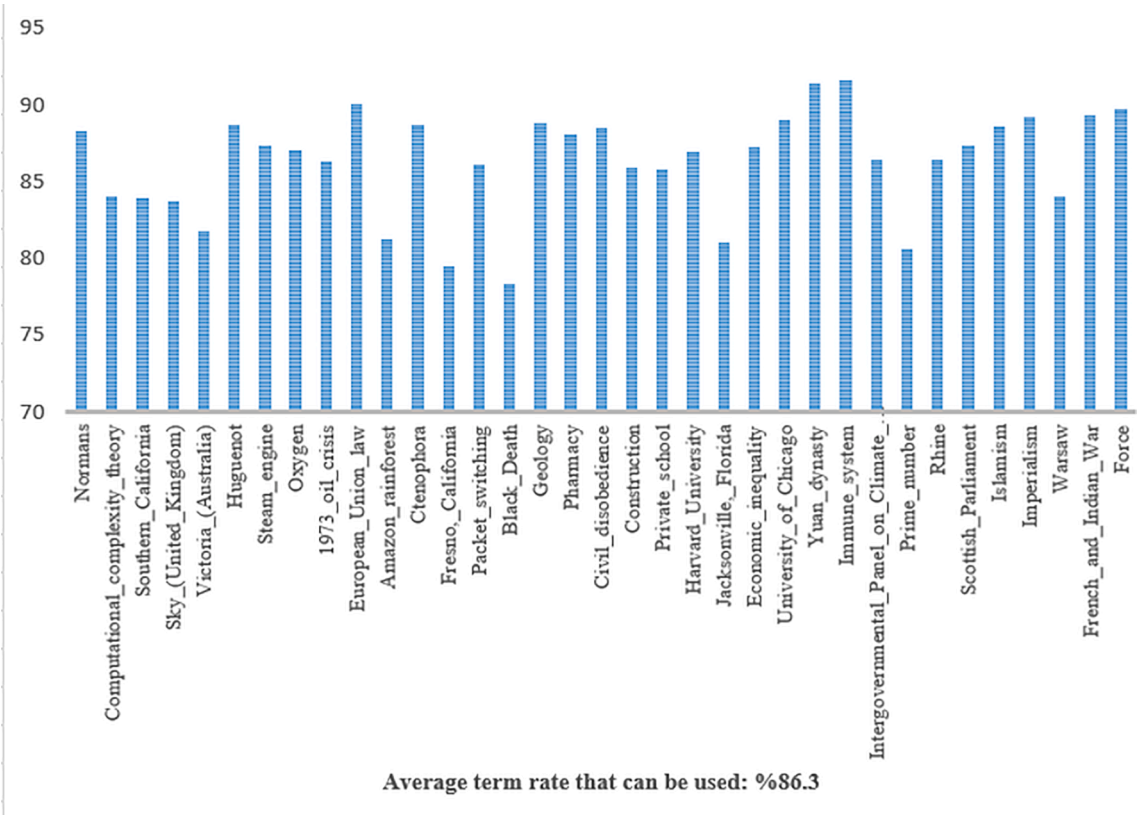

Fig. 7. Term rate (%) that can be used after preprocessing for each article in the Dev_set.

Table 3
Question statistics of dataset.

	Dev_set (%)	Train_set (%)
Total question count	11,873 (%100)	130,319 (%100)
Question count without answer	5945 (%50.08)	43,498 (%33.38)
Question count with answer	5928 (%49.92)	86,821 (%66.62)

Table 4
Distribution of question pronouns.

Question pronouns	Dev_set	Train_set	Total pronoun	Rate (%)
What	3561	49,123	52,684	56.80
Who	537	9813	10,350	11.16
How	641	9187	9828	10.59
When	470	6537	7007	7.55
Which	311	5812	6123	6.60
Where	250	3629	3879	4.18
Others	62	1515	1577	1.7
Why	96	1205	1301	1.4

question pronouns. 'What' is mostly used among the question pronouns. Afterwards, 'who' is used mostly. Since question pronouns such as "who, when, where" indicate specific terms such as person, place, and time. It is thought to be very useful for answer detection.

4.2. Analysis of sentence selection

The sentence selection according to QTP has been analyzed in this section. Firstly, sentence parsing libraries are examined for sentence selection. Because, it is very important to correctly parse paragraphs into sentences. Sent_tokenize[4] and punkt[5] methods of NLTK[6] libraries have been used for sentence parsing. In order to select the related sentence obtained by these methods, the previously explained operations have been performed in Fig. 2. The statistics of these methods for sentence selection are shown in Table 5. The punkt method, which gives a better result, has been selected for sentence parser in the later stages.

Table 5
The success of NLTK methods in sentence selection.

	Sent_tokenize **Preprocess = Y**		Punkt **Preprocess = N**	
	Dev_set	Train_set	Dev_set	Train_set
Sentence selection count	5668	81,312	5688	81,766
Total question count	5928	86,821	5928	86,821
Selection rate (%)	95.59	93.65	**95.93**	**94.17**

Then, it was understood that the dataset has been parsed incorrectly into sentences due to a problem such as not having punctuation marks at the end of some sentences for the sentence parsing. Therefore, a method, which is called solved dataset problem (SDP), has been developed to solve this problem. Problems such as starting a sentence with a lowercase letter after the punctuation mark, and no punctuation mark at the end of sentences have been solved with SDP. The positive effect of SDP to sentence selection is shown in Table 6.

Lemmatization technique was performed to all questions, answers and sentences in the entire dataset. Then, its effect has been analyzed for sentence selection. After lemmatization, paragraphs are parsed into

[4] https://www.nltk.org/api/nltk.tokenize.html
[5] https://www.nltk.org/_modules/nltk/tokenize/punkt.html
[6] https://www.nltk.org/

Table 6
The effect of the SDP method to sentence selection.

	Punkt				Punkt with SDP			
	Preprocess = Y		**Preprocess = N**		**Preprocess = Y**		**Preprocess = N**	
	Dev_set	Train_set	Dev_set	Train_set	Dev_set	Train_set	Dev_set	Train_set
Sentence selection count	5688	81,766	5878	82,955	5720	82,288	5879	85,956
Total question count	5928	86,821	5928	86,821	5928	86,821	5928	86,821
Selection rate (%)	95.93	94.17	99.145	98.99	**96.475**	**94.777**	**99.156**	**99.002**

Table 7
The effect of lemmatization on sentence selection for entire dataset.

	Lemmatization				Lemmatization with POS tagging			
	Preprocess = Y		**Preprocess = N**		**Preprocess = Y**		**Preprocess = N**	
	Dev_set	Train_set	Dev_set	Train_set	Dev_set	Train_set	Dev_set	Train_set
Sentence selection count	4536	63,004	4660	65,713	4509	62,862	4624	65,470
Total question count	5928	86,821	5928	86,821	5928	86,821	5928	86,821
Selection rate (%)	76.5	72.56	78.59	75.68	76.04	72.40	77.98	75.4

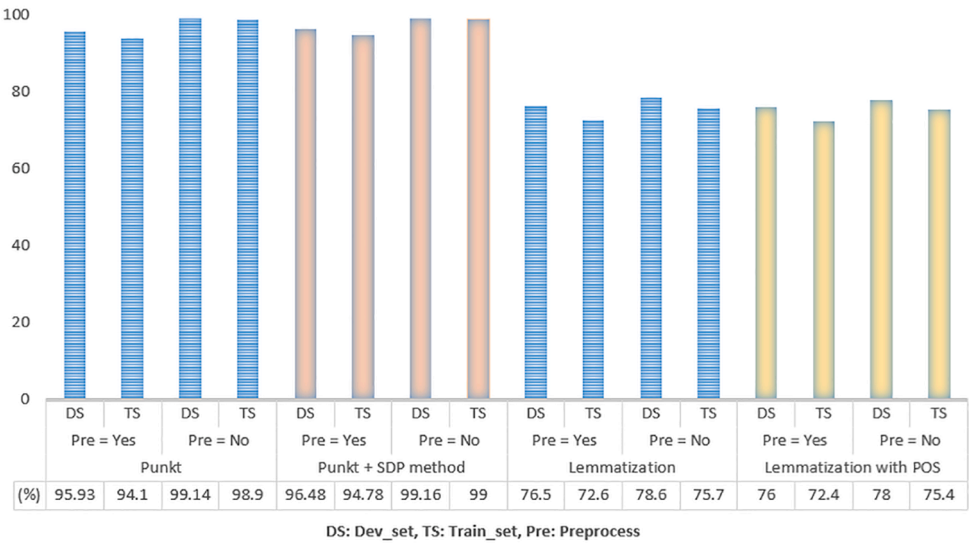

Fig. 8. The success of performed methods for sentence selection.

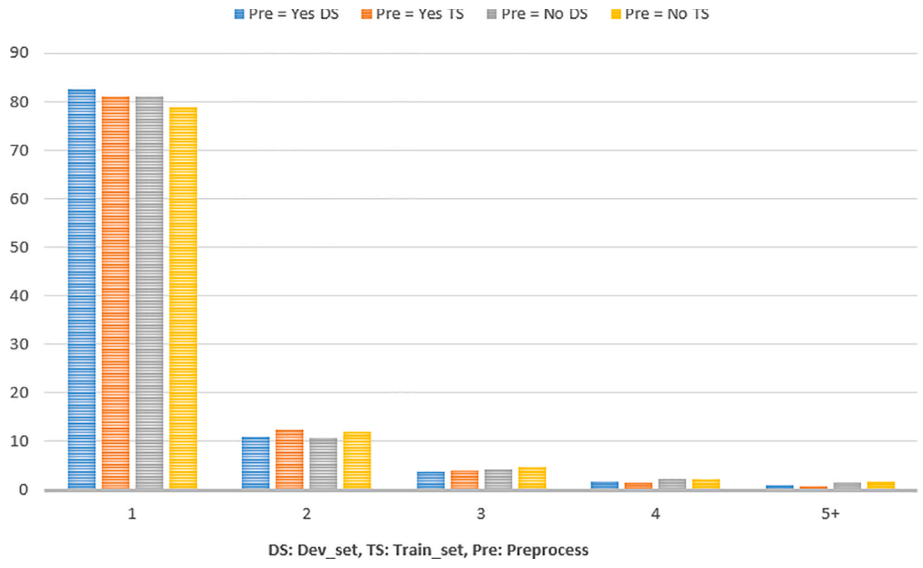

Fig. 9. Statistics on the rank of the chosen sentence according to QTP.

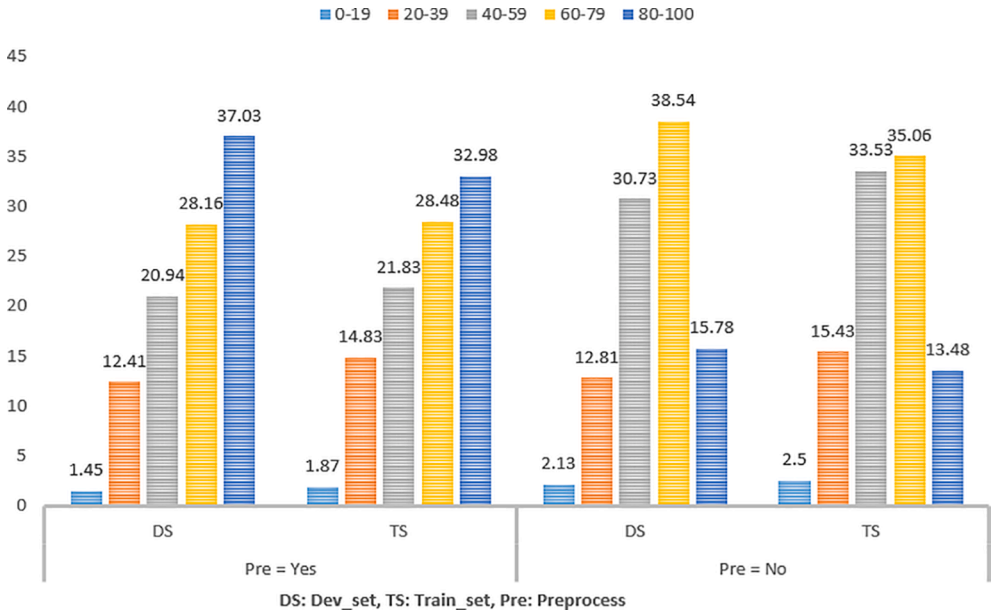

Fig. 10. QTP range statistics for the chosen sentence.

Table 8
Statistics of answer detection on chosen sentences according to QTP.

	Punkt				Punkt with SDP			
	Preprocess = Y		**Preprocess = N**		**Preprocess = Y**		**Preprocess = N**	
	Dev_set	Train_set	Dev_set	Train_set	Dev_set	Train_set	Dev_set	Train_set
RC	116	1604	118	1713	117	1606	118	1713
SNER	354	6539	365	6850	503	9577	518	9989
SPOS	246	4014	253	4150	249	4027	253	4137
Total answer detection	716	12,157	736	12,713	869	15,210	889	15,839
Total related sentence	5688	81,766	5878	85,955	5720	82,288	5879	85,956
Acc (%)	12.59	14.87	12.52	14.79	**15.19**	**18.48**	**15.12**	**18.42**

Table 9
The effect of lemmatization for only sentences for which the answer can't be detected.

	Lemmatization				Lemmatization with POS tagging			
	Preprocess = Y		**Preprocess = N**		**Preprocess = Y**		**Preprocess = N**	
	Dev_set	Train_set	Dev_set	Train_set	Dev_set	Train_set	Dev_set	Train_set
RC	122	1709	123	1819	134	1899	136	2015
SNER	515	9769	532	10,190	519	9869	535	10,299
SPOS	289	4574	294	4695	300	4650	304	4775
Total answer detection	926	16,052	949	16,704	953	16,418	975	17,089
Total related sentence	5720	82,288	5879	85,956	5720	82,288	5879	85,956
Acc (%)	16.18	19.5	16.14	19.42	**16.66**	**19.95**	**16.58**	**19.88**

Table 10
The effect of lemmatization after sentence selection and before the answer detection.

	Lemmatization				Lemmatization with POS tagging			
	Preprocess = Y		**Preprocess = N**		**Preprocess = Y**		**Preprocess = N**	
	Dev_set	Train_set	Dev_set	Train_set	Dev_set	Train_set	Dev_set	Train_set
RC	91	1205	91	1286	105	1441	106	1525
SNER	506	9536	523	9942	502	9479	518	9887
SPOS	258	4164	264	4276	261	4080	265	4185
Total answer detection	855	14,905	878	16,704	868	15,000	889	15,597
Total related sentence	5720	82,288	5879	85,956	5720	82,288	5879	85,956
Acc (%)	14.95	18.11	14.93	18.03	**15.17**	**18.22**	**15.12**	**18.14**

Table 11
The effect of lemmatization for the entire dataset on answer detection.

	Lemmatization				Lemmatization with POS tagging			
	Preprocess = Y		**Preprocess = N**		**Preprocess = Y**		**Preprocess = N**	
	Dev_set	Train_set	Dev_set	Train_set	Dev_set	Train_set	Dev_set	Train_set
RC	82	1072	83	1101	90	1284	91	1317
SNER	393	7162	405	7484	385	7067	398	7368
SPOS	225	3324	229	3442	233	3188	232	3291
Total answer detection	700	11,508	717	12,027	708	11,539	721	11,976
Total related sentence	4536	63,004	4660	65,713	4509	62,862	4624	65,470
Acc (%)	15.43	18.34	15.38	18.3	**15.70**	**18.35**	**15.59**	**18.3**

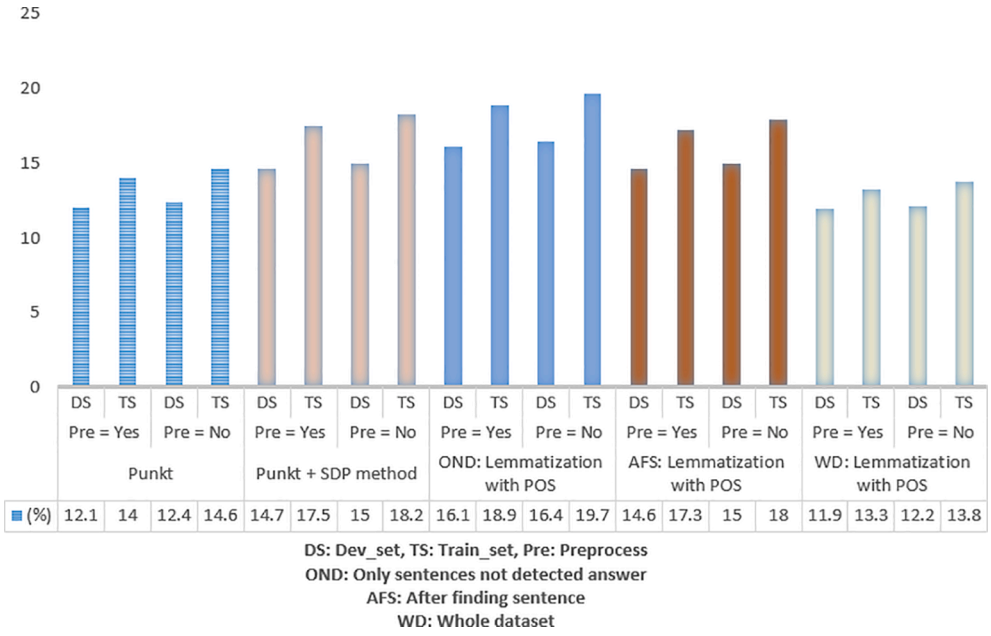

Fig. 11. Statistics of all methods for answer detection.

Table 12
The acc of BLU and BLC models for answer detection.

	BBU	**BBC**	**BLU**	**BLC**
True answer detection	4570	4319	4915	4958
Total question count	5928	5928	5928	5928
Acc (%)	**77.09**	**72.857**	**82.911**	**83.637**

Table 13
The acc of RNP methods on BERT models.

	BBU	**BBC**	**BLU**	**BLC**
RC	21	30	18	16
SNER	36	41	18	14
SPOS	50	70	35	34
Total answer detection	107	141	71	64
Questions that BERT can't answer	1358	1609	1013	970
Acc (%)	7.88	8.76	7.01	6.60

Table 14
The acc of each RNP method separately.

	BBU (%)	**BBC (%)**	**BLU (%)**	**BLC (%)**
RC	21 (19.62)	30 (21.27)	18 (25.35)	16 (25)
SNER	36 (33.64)	42 (29.79)	18 (25.35)	14 (21.875)
SPOS	55 (51.40)	84 (59.57)	36 (50.71)	36 (56.25)
Total answer detection	107	141	71	64

Table 15
The effects of using pre-trained BERT models and RNP methods together on answer detection.

	BBU	**BBC**	**BLU**	**BLC**
True answer detection of pre-trained models	4570	4319	4915	4958
True answer detection of RNP methods	107	141	71	64
Total answer detection count	4677	4460	4986	5022
Total questioncount	5928	5928	5928	5928
Acc (%)	**78.89**	**75.23**	**84.11**	**84.71**

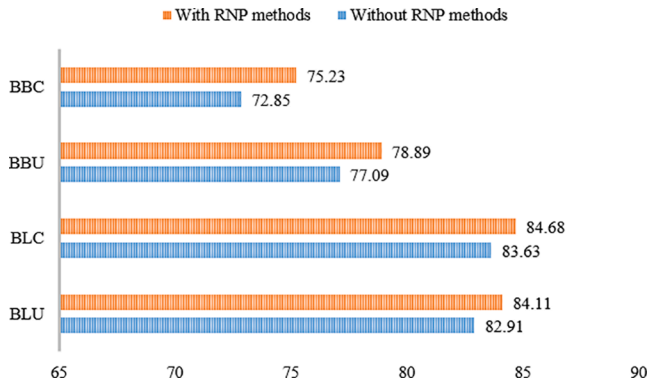

Fig. 12. Statistics of the pre-trained BERT models for answer detection.

Table 16
Distribution of question pronouns whose answers can't be detected.

Question pronouns	BBU (%)	BBC (%)	BLU (%)	BLC (%)	Total pronoun
What	856 (24.03)	1023 (28.72)	623 (17.49)	606 (17.01)	3561
Who	93 (17.31)	103 (19.18)	66 (12.29)	57 (10.61)	537
How	145 (22.62)	166 (25.89)	128 (19.96)	105 (16.38)	641
When	56 (11.91)	77 (16.38)	39 (8.29)	44 (9.36)	470
Which	56 (18.00)	81 (26.04)	45 (14.47)	43 (13.82)	311
Where	74 (29.6)	78 (31.20)	47 (18.80)	46 (18.40)	250
Others	29 (46.77)	36 (58.06)	23 (37.09)	24 (38.70)	62
Why	49 (51.04)	45 (46.87)	42 (43.75)	45 (46.87)	96

sentences with "Punkt with SDP". POS tagging can be used in lemmatization. The effect of lemmatization is shown in Table 7. Table shows that the lemmatization was less successful in sentence selection compared to the previous Table 6.

To summarize all the performed operations for sentence selection, the chart regarding the selection rate of sentences is shown in Fig. 8. The figure shows that the most successful method of sentence selection is "Punkt with SDP". When the lemmatization has performed in the entire dataset, the selection rate of sentences has decreased approximately 20%.

Next, the rank of the chosen sentence according to QTP has been analyzed. The statistics on the rank of the chosen sentence is shown in Fig. 9. Figure shows that these sentences are in the first rank with a high rate. Approximately 80% of the sentences have been detected in the first rank. The inference made from this is that most of the sentences contain a high rate of question terms. This shows that there are many sentences that can be useful for the answer detection.

Finally, the created chart for the QTP values of the chosen sentences is shown in Fig. 10. There are approximately 65% question terms in the range of 60%-100% with preprocessing. QTP in the range of 80%-100% is very low without preprocessing. The reason for being in 15% is that stopwords aren't removed and question pronouns are searched in related sentences.

4.3. Analysis of answer detection

Answer detection has been analyzed with RNP methods according to the chosen sentences. The accuracy rate (acc) has been determined as the criteria for success. Proposed RC, SNER and SPOS methods have been used for answer detection. In the sequential operation, RC has been selected as the first because it works faster. SNER, which detects more answers, has been selected as the second method. The acc of these methods on answer detection for the chosen sentences is shown in Table 8. Table shows that while RNP methods using the punkt method for sentence parsing detected 12% −14% answers, RNP methods provided approximately 3% acc increase with the effect of SDP. In the next stages, operations were performed on the "Punkt with SDP" method.

Among the selected sentences, the lemmatization has been performed for the related sentences, questions and answers in which only the answer couldn't be detected. The lemmatization process was also analyzed as it can be used in conjunction with POS tagging. The acc of these operations are shown in Table 9. Compared to the success of Table 8, the acc rate has increased by approximately 1.5%.

In addition, the lemmatization technique has been performed to the question, answer and sentence after sentence selection and before the answer detection. Table 10 indicates the acc of RNP methods for this technique. It has been observed that the acc has decreased slightly compared to Table 9.

Finally, the lemmatization technique has been performed for the entire dataset before the sentence selection and answer detection. The effect of this technique for answer detection is shown in Table 11. Table shows that this technique has had a negative effect on answer detection as well as sentence selection. The count of answer detection is very low compared to Table 9. The reason for this is that the sentence structure has changed due to lemmatization.

The acc rate of the RNP methods for answer detection is shown in Fig. 11. It is more successful for each method to use lemmatization with POS tagging. Therefore, only this method for lemmatization is shown in this figure. The acc rates are the total rate at which the RNP methods for answer detection. Figure shows that the most successful method is lemmatization performed only for sentences for which the answer can't be detected. The reason why this method is most successful is that, in addition to the previously detected answers, the sentences are lemmatized only for the undetectable answers. The most unsuccessful method has been to apply lemmatization after sentence selection and before the answer detection.

5. Analysis of BERT with RNP methods

The BERT language model has been tested for SQuAD. Thus, this model is used to test RNP methods. Since the BERT model has been trained with Train_set, the model has been tested with Dev_set. Test set of SQuAD isn't used for the test, because it's hidden.

The BERT-Base and BERT-Large models were trained as uncased or cased. Uncased means that the text has been lowercased before tokenization. The uncased model also strips out any accent markers. Cased means that the true case and accent markers are preserved (Devlin, 2021). After pre-training, these models were fine-tuned on the SQuAD. Finally, pre-trained models of these models were created for the SQuAD test (learning rate: 3e-5, epoch number: 2, sequence length: 384, document stride: 128).

BERT-Base-Uncased[7] (BBU), BERT-Base-Cased[8] (BBC), BERT-Large-Uncased[9] (BLU) and BERT-Large-Cased[10] (BLC) pre-trained models have been tested with Dev_set. The acc rate of these models for answer detection are shown in Table 12. While both BERT-Large models have achieved over 82.9%, BERT-Base models have obtained below 77.1% acc.

The answers of 1013 questions for BLU, 970 questions for BLC, 1358 questions for BBU and 1609 questions for BBC couldn't be detected for these pre-trained models (Table 12). The RNP methods have been used sequentially for answer detection of these questions. RNP methods have been performed to the chosen sentences according to QTP. The acc of these methods on BERT models are shown in Table 13. Table shows that these methods have detected answers that the pre-trained models couldn't detect.

RNP methods have been performed one by one to analyze the acc of each method. The acc of each method is shown in Table 14. Table shows that the most successful method was SPOS, and the most unsuccessful was RC for each method. SPOS method detected over 50% of the

[7] https://huggingface.co/twmkn9/bert-base-uncased-squad2
[8] https://huggingface.co/deepset/bert-base-cased-squad2
[9] https://huggingface.co/bert-large-uncased-whole-word-masking-finetuned-squad
[10] https://huggingface.co/bert-large-cased-whole-word-masking-finetuned-squad

Table 17
Distribution of question pronouns for acc of RNP methods.

	BBU				BBC				BLU				BLC			
	RC	SNER	SPOS	Total	RC	SNER	SPOS	Total	RC	SNER	SPOS	Total	RC	SNER	SPOS	Total
What	13	9	31	53	22	11	45	73	13	5	21	39	12	3	19	34
Who	3	10	7	18	6	12	11	23	2	3	4	8	2	3	4	8
How	2	3	8	13	1	3	16	20	3	2	7	12	2	1	8	11
When	0	4	8	9	0	7	11	14	0	0	3	3	0	3	4	6
Which	1	4	0	5	0	2	0	2	0	4	0	4	0	1	0	1
Where	1	6	1	8	1	7	1	9	0	4	1	5	0	3	1	4
Why	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Total	21	36	55	107	30	42	84	141	18	18	36	71	16	14	36	64

Table 18
Comparison of the answers of the pre-trained BERT model and RNP methods with the examples.

	RNP Methods	Pre-trained BERT model
Example 1	RC Sentence: College sports are also popular in southern California. Question:What other kind of sport is popular in southern California? Prediction: College Answer: College	Paragraph: College sports are also popular in southern California. The UCLA Bruins and the USC Trojans both field teams in NCAA Division I in the Pac-12 Conference, and there is a longtime rivalry between the schools. Question:What other kind of sport is popular in southern California? Prediction:College sports Answer: College
Example 2	SNER Sentence: In 1932 the German encyclopedia Knaurs Lexikon stated the length as 1,320 km (820 miles), presumably a typographical error. Question:Who stated a change of the length of the Rhine? Prediction: Knaurs Lexikon Answer: Knaurs Lexikon	Paragraph: Until 1932 the generally accepted length of the Rhine was 1,230 km (764 miles). In 1932 the German encyclopedia Knaurs Lexikon stated the length as 1,320 km (820 miles), presumably a typographical error. After this number was placed into the authoritative Brockhaus Enzyklopädie, it became generally accepted and found its way into numerous textbooks and official publications. The error was discovered in 2010, and the Dutch Rijkswaterstaat confirms the length at 1,232 km (766 miles). Question:Who stated a change of the length of the Rhine? Prediction: <No Answer> Answer: Knaurs Lexikon
Example 3	SPOS Sentence: Major events also play a big part in tourism in Victoria, particularly cultural tourism and sports tourism. Question:What part do events in Victoria's economy play? Prediction: tourism Answer: tourism	Paragraph: Major events also play a big part in tourism in Victoria, particularly cultural tourism and sports tourism. Most of these events are centred on Melbourne, but others occur in regional cities, such as the V8 Supercars and Australian Motorcycle Grand Prix at Phillip Island, the Grand Annual Steeplechase at Warrnambool and the Australian International Airshow at Geelong and numerous local festivals such as the popular Port Fairy Folk Festival, Queenscliff Music Festival, Bells Beach SurfClassic and the Bright Autumn Festival. Question:What part do events in Victoria's economy play? Prediction: <No Answer> Answer: tourism

detected answers.

Using pre-trained BERT models and RNP methods together has increased the acc. The effects of RNP methods for pre-trained BERT models are shown in Table 15. The acc has increased due to combined use. Table indicates that NLP techniques aren't used enough in the BERT models.

As a result, the acc of pre-trained BERT models with or without RNP methods are shown Fig. 12. The chart shows that the acc has increased by approximately 1.1% to 2.4% with RNP methods. In other words, the use of RNP methods has a positive effect for the pre-trained BERT models.

After analyzing the success of the RNP methods on the BERT models, question pronouns that these models could not answer have been analyzed. Analysis results are shown in Table 16. Analysis shows that these models can hardly answer open-ended questions such as why, what. Models are successful for questions that express something like time, person.

Finally, question pronouns belonging to the answers detected by the RNP methods have been analyzed. Which pronouns are detected more successfully by these methods are shown in Table 17. Table shows that the SPOS method is the most successful method in questions involving 'what' pronouns. The SNER method is very successful in questions involving "who, where" pronouns. The RC method is the most unsuccessful for other question pronouns, except 'what'.

6. Discussion

SQuAD is a benchmark platform for Q&A systems. It provides a reading comprehension dataset for testing the performance of Q&A systems. Generally, they are based on deep learning models based on BERT, ELMo, etc. (SQuAD, 2021). The motivation of this study is to focus on the questions that BERT models don't answer. We developed 3 natural language based methods, namely RNP, that increases the performance of BERT based Q&A systems.

In Table 18, we outlined the basic differences between the proposed RNP methods and the pre-trained BERT model. As can be seen from Example 1, the proposed RC method is applied for the questions that can't be answered with the pre-trained BERT model, however, the pre-

trained BERT model doesn't find the answer correctly. In examples 2 and 3, the answer to the question couldn't be found by the pre-trained BERT model. SNER and SPOS methods are used to find the answer to these questions by not requiring any training phase. Lastly, RNP methods focus on the sentence where the answer is in, whereas pre-trained BERT models focus on the paragraph. Since RNP methods don't require any training phase, the proposed methods can easily be applied to increase the accuracy of the pre-trained BERT models.

7. Conclusion and future work

In this study, The NLP-QAS that can be used with BERT models was proposed to answer the questions. This system includes two stages: sentence selection and answer detection. Firstly, an analysis was performed with proposed NLP-QAS on SQuAD. Many variations such as punkt, punkt with SDP and lemmatization were used for sentence selection. Among the variations, punkt with SDP was the most successful method in sentence selection. In the answer detection stage, RNP methods named RC, SPOS and SNER were proposed. RNP methods try to detect the answer on the chosen sentence as a result of sentence selection. The most successful result was obtained by applying lemmatization only for the sentences for which the answer couldn't be detected.

In the second analysis, it has been shown that the questions can be answered by applying the RNP methods in conjunction with BERT models. The questions related to the answers that couldn't be detected by the original BERT models were analyzed. As a result of the application of RNP methods with sentence selection, an increase in accuracy between 6.6% and 8.76% was achieved in answer detection with NLP-QAS. When success in all BERT models was analyzed, the acc rate of answer detection by RNP methods increased by approximately 1.1% to 2.4%. It is shown that the NLP based extension to BERT models for the SQuAD increases the overall performance of Q&A system.

As a next study, we aim to explore the effect of RNP methods on the derivative language models of BERT like ALBERT, RoBERTa, SemBERT and SG-NET. We can also apply NLP techniques like lemmatization and stemming to the original SQuAD, then train the BERT language model and measure the effect of NLP techniques.

As can be seen from Table 15 in subsection 4.4., there are still questions that cannot be answered in SQuAD. In the future study, we will work on an ontology based Q&A system, O-QAS, as an additional question answering method to find the answers to the unanswered questions in SQuAD. In O-QAS, we firstly parse the sentences in the related paragraphs of articles into < Subject,Predicate,Object > triples. Then, RC method of NLP-QAS is applied on the triples. Lastly, the remaining part of triples will be selected as the possible answer.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

9. Ethics approval

Ethical approval is not required as data on any human or animal subjects are not obtained in our study. The dataset was downloaded as open source.

CRediT authorship contribution statement

Zekeriya Anil Guven: Conceptualization, Methodology, Software, Visualization, Resources, Investigation, Validation, Data curation, Writing – original draft, Writing – review & editing. **Murat Osman Unalir:** Project administration, Conceptualization, Methodology, Writing – original draft, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix

The sample questions answered by the RNP methods for the BERT models and the performed operations are shown in the tables.

Table 19
The RC method example for BERT model.

Question Pronoun	RC Method Examples
What	Question :What other kind of sport is popular in southern California? Sentence: College *sports are also popular in southern California*. Remove stopwords and question terms: College. Answer: College
Who	Question: Who is viewed as the first modern geologist? Sentence: James Hutton *is often viewed as the first modern geologist*. Remove stopwords and question terms: James Hutton Answer: James Hutton
How	Question: How do cestids swim? Sentence: *Cestids can swim* by undulating their bodies as well as by the beating of their comb-rows. Remove stopwords and question terms: by undulating their bodies as well as by the beating of their comb-rows. Answer:by undulating their bodies as well as by the beating of their comb-rows.
Which	Question: Which findings suggested that the region was densely populated? Sentence: *However, recent* anthropological *findings have suggested that the region was actually densely populated*. Remove stopwords and question terms:anthropological Answer:anthropological
Where	Question: Where might committees meet outside of Parliament? Sentence:*Committees can also meet at other* locations *throughout* Scotland. Remove stopwords and question terms: locations Scotland. Answer: locations Scotland
Why	Question: Why were the 2011 Special Reports issued? Sentence:*Both Special Reports were* requested *by* governments. Remove stopwords and question terms: requested governments. Answer:requested governments

Table 20
The SNER method example for BERT model.

Question Pronoun	SNER Method Examples
What	Question : What country did the Normans invade in 1169? Sentence:The Normans settled mostly in an area in the east of Ireland, later known as the Pale, and also built many fine castles and settlements, including Trim Castle and Dublin Castle. Answer Tag: What country = GPE Sentence NER tags: *Normans,* **Ireland**, *Pale, Trim Castle, Dublin Castle* ('NORP', 'GPE', 'ORG', 'PERSON', 'PERSON') Answer:Ireland
Who	Question: Who translated this version of the scriptures? Sentence: Around 1294, a French version of the Scriptures was prepared by the Roman Catholic priest, Guyard de Moulin. Answer Tag: Who = PERSON Sentence NER tags:*1294, French, Roman Catholic,***Guyard de Moulin** ('DATE', 'NORP', 'NORP', 'PERSON') Answer:Guyard de Moulin
How	Question: How much gold did Victoria produce in the years of 1851–1860? Sentence: Victoria produced in the decade 1851–1860 20 million ounces of gold, one third of the world's output[citation needed] Answer Tag: How much = QUANTITY or MONEY Sentence NER tags: *Victoria, the decade,* **20 million ounces***, one third* ('GPE', 'DATE', 'QUANTITY', 'CARDINAL') Answer: 20 million ounces
When	Question: When did Mongke Khan become Great Khan? Sentence: Möngke Khan succeeded Ögedei's son, Güyük, as Great Khan in 1251. Answer Tag: When = DATE Sentence NER tags: *Möngke Khan, Ögedei, Güyük, Great Khan,***1251** ('PERSON', 'GPE', 'GPE', 'PERSON', 'DATE') Answer: 1251
Which	Question: Which year was the case Commission v Italy that dealt with cocoa products? Sentence: In a 2003 case, Commission v Italy Italian law required that cocoa products that included other vegetable fats could not be labelled as "chocolate". Answer Tag: Which year = DATE Sentence NER tags: **2003***, Italy, Italian* ('DATE', 'GPE', 'NORP') Answer: 2003
Where	Question:Where does the Rhine make a distinctive turn to the north? Sentence: The river makes a distinctive turn to the north near Chur. Answer Tag: Where = GPE Sentence NER tags: **Chur** ('GPE') Answer: Chur

Table 21
The SPOS method example for BERT model.

Question Pronoun	SPOS Method Examples
What	Question :What kind of destruction did the 1994 earthquake cause the most of in US history? Sentence: It caused the most property damage of any earthquake in U.S. history, estimated at over $20 billion. Answer Tag: What = 'NN': Sentence POS tags: ('Itd', 'NNP'), **('property', 'NN'), ('damage', 'NN')**, ('any', 'DT'), ('U.S.', 'NNP'), (',', ','), ('estimated', 'VBN'), ('at', 'IN'), ('over', 'IN'), ('$', '$'), ('20′, 'CD'), ('billion', 'CD'), ('.', '.') Answer: property damage
Who	Question: Who sets the legislative agenda in Victoria? Sentence: The Premier is the public face of government and, with cabinet, sets the legislative and political agenda. Answer Tag: Who= 'NNP': Sentence POS tags: ('The', 'DT'), **('Premier', 'NNP')**, ('is', 'VBZ'), ('public', 'JJ'), ('face', 'NN'), ('of', 'IN'), ('government', 'NN'), ('and', 'CC'), (',', ','), ('with', 'IN'), ('caet', 'NN'), (',', ','), ('the', 'DT'), ('and', 'CC'), … Answer: Premier
How	Question: How many provinces did the Ottoman empire contain in the 17th century? Sentence: At the beginning of the 17th century the empire contained 32 provinces and numerous vassal states. Answer Tag: How many = 'CD': Sentence POS tags: ('At', 'IN'), ('bening', 'NN'), ('of', 'IN'), ('theed', 'NN'), **('32′, 'CD')**, ('and', 'CC'), ('numerous', 'JJ'), ('vassal', 'NN'), ('states', 'NNS'), ('.', '.') Answer: 32
When	Question: When was the colony destroyed? Sentence: A September 1565 French naval attack against the new Spanish colony at St. Augustine failed when its ships were hit by a hurricane on their way to the Spanish encampment at Fort Matanzas. Answer Tag: When = 'CD': Sentence POS tags: ('A', 'DT'), ('September', 'NNP'), **('1565′, 'CD')**, ('French', 'NNP'), ('naval', 'JJ'), ('attack', 'NN'), ('against', 'IN'), ('new', 'JJ'), ('Spanish', 'JJ'), ('at', 'IN'), ('St.', 'NNP'), ('Augustine', 'NNP'), ('failed', 'VBD'), ('when', 'WRB'), … Answer: 1565
Where	Question: Where did Korea border Kublai's territory? Sentence: Kublai secured the northeast border in 1259 by installing the hostage prince Wonjong as the ruler of Korea, making it a Mongol tributary state. Answer Tag: Where= 'NN': Sentence POS tags: ('Kublai', 'NNP'), ('secured', 'VBD'), ('the', 'DT'), **('northeast', 'NN')**, ('in', 'IN'), ('1259′, 'CD'), ('by', 'IN'), ('installing', 'VBG'), ('the', 'DT'), ('hostage', 'NN'), ('prince', 'NN'), ('Wonjong', 'NNP'), ('as', 'IN'), ('the', 'DT'), … Answer: northeast

References

Annotation Specifications. (n.d.). Retrieved January 21, 2021, from https://spacy.io/api/annotation.

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., & Zheng, X. (2016). TensorFlow: A system for large-scale machine learning. Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016.

Adhikari, A., Ram, A., Tang, R., & Lin, J. (2019). DocBERT: BERT for document classification. In arXiv.

Al-Garadi, M. A., Yang, Y. C., Cai, H., Ruan, Y., O'Connor, K., Graciela, G. H., … Sarker, A. (2021). Text classification models for the automatic detection of nonmedical prescription medication use from social media. *BMC Medical Informatics and Decision Making, 21*(1), 1–13. https://doi.org/10.1186/s12911-021-01394-0

Ballı, S., & Sağbaş, E. A. (2018). Diagnosis of transportation modes on mobile phone using logistic regression classification. *IET Software*. https://doi.org/10.1049/iet-sen.2017.0035

Carleo, G., Cirac, I., Cranmer, K., Daudet, L., Schuld, M., Tishby, N., … Zdeborová, L. (2019). Machine learning and the physical sciences. *Reviews of Modern Physics*. https://doi.org/10.1103/RevModPhys.91.045002

Chowdhary, K. R. (2020). Natural Language Processing. In Fundamentals of Artificial Intelligence (pp. 603–649). Springer India. Doi: 10.1007/978-81-322-3972-7_19.

Devlin, J. (n.d.). GitHub - TensorFlow code and pre-trained models for BERT. Retrieved January 22, 2021, from https://github.com/google-research/bert.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference.

Dong, L., Mallinson, J., Reddy, S., & Lapata, M. (2017). Learning to paraphrase for question answering. EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings. Doi: 10.18653/v1/d17-1091.

Esposito, M., Damiano, E., Minutolo, A., De Pietro, G., & Fujita, H. (2020). Hybrid query expansion using lexical resources and word embeddings for sentence retrieval in question answering. *Information Sciences*. https://doi.org/10.1016/j.ins.2019.12.002

Gokalp, O., Tasci, E., & Ugur, A. (2020). A novel wrapper feature selection algorithm based on iterated greedy metaheuristic for sentiment classification. *Expert Systems with Applications*. https://doi.org/10.1016/j.eswa.2020.113176

Guo, J., He, H., He, T., Lausen, L., Li, M., Lin, H., … Zhu, Y. (2019). GluonCV and GluonNLP: Deep Learning in Computer Vision and Natural Language Processing. *ArXiv, 21*, 1–7. http://arxiv.org/abs/1907.04433.

Güven, Z. A., Diri, B., & Cąkaloglu, T. (2020). Comparison of n-stage Latent Dirichlet Allocation versus other topic modeling methods for emotion analysis. Journal of the Faculty of Engineering and Architecture of Gazi University. https://doi.org/10.17341/gazimmfd.556104.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., … Darrell, T. (2014). *Caffe.*. https://doi.org/10.1145/2647868.2654889

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. In arXiv.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. In arXiv.

Marcus, G. (2020). The next decade in AI: Four steps towards robust artificial intelligence. In arXiv.

Martinez-Gil, J., Freudenthaler, B., & Tjoa, A. M. (2019). Multiple Choice Question Answering in the Legal Domain Using Reinforced Co-occurrence. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). https://doi.org/10.1007/978-3-030-27615-7_10.

McCann, B., Keskar, N. S., Xiong, C., & Socher, R. (2018). The natural language decathlon: Multitask learning as question answering. In arXiv.

Otter, D. W., Medina, J. R., & Kalita, J. K. (2018). A survey of the usages of deep learning in natural language processing. *In arXiv*. https://doi.org/10.1109/tnnls.2020.2979670

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., & Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In arXiv.

Peinelt, N., Nguyen, D., & Liakata, M. (2020). tBERT: Topic Models and BERT Joining Forces for Semantic Similarity Detection. Doi: 10.18653/v1/2020.acl-main.630.

Petroni, F., Rocktäschel, T., Lewis, P., Bakhtin, A., Wu, Y., Miller, A. H., & Riedel, S. (2020). Language models as knowledge bases? EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference. Doi: 10.18653/v1/d19-1250.

Qu, C., Yang, L., Qiu, M., Bruce Croft, W., Zhang, Y., & Iyyer, M. (2019). BERT with history answer embedding for conversational question answering. SIGIR 2019 - Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. Doi: 10.1145/3331184.3331341.

Rachiele, G. (2018). Tokenization and Parts of Speech(POS) Tagging in Python's NLTK library. https://medium.com/@gianpaul.r/tokenization-and-parts-of-speech-pos-tagging-in-pythons-nltk-library-2d30f70af13b.

Rajpurkar, P., Jia, R., & Liang, P. (2018). Know what you don't know: Unanswerable questions for SQuAD. ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers). Doi: 10.18653/v1/p18-21.

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*.

SQuAD. (n.d.). Retrieved January 21, 2021, from https://rajpurkar.github.io/SQuAD-explorer/.

Sun, C., Huang, L., & Qiu, X. (2019). Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. *NAACL HLT 2019–2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*.

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.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., & Brew, J. (2019). Transformers: State-of-the-art natural language processing. In arXiv. Doi: 10.18653/v1/2020.emnlp-demos.6.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized autoregressive pretraining for language understanding. Advances in Neural Information Processing Systems.

Yeh, Y. T., & Chen, Y. N. (2020). QainfoMax: Learning robust question answering system by mutual information maximization. *EMNLP-IJCNLP 2019–2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*.

Zhang, Z., Wu, Y., Zhao, H., Li, Z., Zhang, S., Zhou, X., & Zhou, X. (2019). Semantics-aware BERT for language understanding. *In arXiv*. https://doi.org/10.1609/aaai.v34i05.6510

Zhang, Z., Wu, Y., Zhou, J., Duan, S., Zhao, H., & Wang, R. (2019). SG-Net: Syntax-guided machine reading comprehension. *In arXiv*. https://doi.org/10.1609/aaai.v34i05.6511

Zhou, X., Hu, B., Chen, Q., & Wang, X. (2018). Recurrent convolutional neural network for answer selection in community question answering. *Neurocomputing*. https://doi.org/10.1016/j.neucom.2016.07.082