T.R.
EGE UNIVERSITY
Graduate School of Natural and Applied Sciences

AN ONTOLOGY BASED APPROACH FOR QUESTION ANSWERING SYSTEMS THAT USING MACHINE LEARNING

Zekeriya Anıl GÜVEN

Supervisor: Prof. Dr. Murat Osman ÜNALIR

Department of Computer Engineering

For the degree of Doctor of Philosophy

İzmir
2022

EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

KABUL VE ONAY SAYFASI

Zekeriya Anıl Güven tarafından Doktora tezi olarak sunulan "Makine Öğrenmesi Kullanan Soru Cevaplama Sistemleri için Ontoloji Tabanlı Bir Yaklaşım" başlıklı bu çalışma EÜ Lisansüstü Eğitim ve Öğretim Yönetmeliği ile EÜ Fen Bilimleri Enstitüsü Eğitim ve Öğretim Yönergesi'nin ilgili hükümleri uyarınca tarafımızdan değerlendirilerek savunmaya değer bulunmuş ve 26/09/2022 tarihinde yapılan tez savunma sınavında aday oybirliği/oyçokluğu ile başarılı bulunmuştur.

Jüri üyeleri	**İmza**
Jüri Başkanı :	
Raportör Üye :	
Üye :	
Üye :	
Üye :	

EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ

ETİK KURALLARA UYGUNLUK BEYANI

EÜ Lisansüstü Eğitim ve Öğretim Yönetmeliğinin ilgili hükümleri uyarınca Doktora Tezi olarak sunduğum "Makine Öğrenmesi Kullanan Soru Cevaplama Sistemleri için Ontoloji Tabanlı Bir Yaklaşım" başlıklı bu tezin kendi çalışmam olduğunu, sunduğum tüm sonuç, doküman, bilgi ve belgeleri bizzat ve bu tez çalışması kapsamında elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara atıf yaptığımı ve bunları kaynaklar listesinde usulüne uygun olarak verdiğimi, tez çalışması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını, bu tezin herhangi bir bölümünü bu üniversite veya diğer bir üniversitede başka bir tez çalışması içinde sunmadığımı, bu tezin planlanmasından yazımına kadar bütün safhalarda bilimsel etik kurallarına uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul edeceğimi beyan ederim.

26/09/2022

Zekeriya Anıl GÜVEN

ABSTRACT

AN ONTOLOGY BASED APPROACH FOR QUESTION ANSWERING SYSTEMS THAT USING MACHINE LEARNING

GÜVEN, Zekeriya Anıl

Ph.D. in Department of Computer Engineering

Supervisor: Prof. Dr. Murat Osman ÜNALIR

26/09/2022, 120 pages

Nowadays, due to the increase in textual data, the processing and analysis of these data have become more difficult. Natural language processing is a field that has been developed to solve this problem, and has been applied to many topics such as sentiment analysis, question answering, spam detection. Question answering, which aims to answer questions in natural language, is the main topic of this thesis. For question answering, SQuAD, which was created by Stanford University and consisted of single-answer rather than multiple-choice questions, is used as the dataset. There is a benchmarking platform for SQuAD, and many language models based on machine learning or deep learning are used for this platform. In deep learning-based models, retraining the model when new data is wanted to be added to the model creates a problem in terms of time and cost. Because of these problems, it is aimed to propose two extensions, namely natural language-based and triple-based, which can increase the success of these models on the language models used for SQuAD, and these models do not need retraining.

In the first extension, a natural language-based method is proposed by making use of natural language processing methods. By using string operations, Named Entity Recognition and Part of Speech tagging methods, remove&compare, search with Named Entity Recognition and Part of Speech tagging methods, namely RNP, have been developed. In order to use these methods, firstly, the related sentence in the paragraph is selected. RNP

methods are analyzed on the selected sentence. This analysis is applied to the whole dataset and BERT language models. When the questions on the whole SQuAD were examined, RNP methods determined the correct answers at a rate of approximately 19.9%. As a result of the analysis applied to the questions that the BERT models answered incorrectly, RNP methods increased the accuracy value of BERT models between 1.1% and 2.4% as an extension.

The triple-based extension is inspired by the ontology approach. This method, it is aimed to determine the answer correctly with triple extraction by making use of the subject-predicate-object triples of ontology. First, the related sentence selection process is performed according to question terms. A candidate answer is sought among these triples by extracting the triples on the selected sentence. The search process is carried out by analyzing the question terms. This extension is implemented for all questions that have an answer and no answer. Questions answered incorrectly by the BERT, ALBERT, ELECTRA, RoBERTa, and SpanBERT language models are analyzed. As a result of the analysis, the triple-based extension increased the accuracy of the language models between 3.3% and 7.5%.

These extensions show that they can answer questions that language models answer incorrectly and increase the accuracy value. Also, both extensions do not need any retraining as intended. Only when the paragraph and question are given to both extensions as input, it can analyze the questions independently of the language model and dataset.

Keywords: Natural Language Processing, BERT, Data Analysis, Question Answering, Language Model, SQuAD, Information Extraction.

ÖZET

MAKİNE ÖĞRENMESİ KULLANAN SORU CEVAPLAMA SİSTEMLERİ İÇİN ONTOLOJİ TABANLI BİR YAKLAŞIM

GÜVEN, Zekeriya Anıl

Doktora Tezi, Bilgisayar Mühendisliği Anabilim Dalı
Tez Danışmanı: Prof. Dr. Murat Osman ÜNALIR
26/09/2022, 120 sayfa

Günümüzde metinsel verilerin artmasından dolayı bu verilerin işlenmesi ve analizi daha da zorlaşmaktadır. Doğal dil işleme bu soruna çözüm için geliştirilmiş bir alandır, ve duygu analizi, soru cevaplama, spam tespiti gibi birçok konu için uygulanmaktadır. Doğal dildeki soruları yanıtlamayı amaçlayan soru cevaplama, bu tezin ana konusudur. Soru cevaplama için veri seti olarak Stanford Üniversitesi'nin oluşturduğu, çoktan seçmeli değil tek cevabı olan sorulardan oluşan SQuAD kullanılmaktadır. SQuAD için kıyaslama platformu bulunmaktadır, ve bu platform için makine öğrenmesi veya derin öğrenme tabanlı birçok dil modeli kullanılmaktadır. Derin öğrenme tabanlı modellerde, modele yeni bir veri eklenmek istediğinde modelin tekrardan eğitilmesi zaman ve maliyet açısından sorun oluşturmaktadır. Bu sorunlardan dolayı, SQuAD için kullanılan dil modelleri üzerinde bu modellerin başarısını artırabilecek, bu modellerin yeniden eğitime ihtiyaç duymadığı doğal dil-tabanlı ve üçlü-tabanlı olmak üzere iki eklenti önerilmesi amaçlanmaktadır.

İlk eklentide doğal dil işleme yöntemlerinden yararlanan doğal dil-tabanlı bir yöntem önerilmektedir. String işlemleri, Varlık İsmi Tanıma ve Cümle Ögeleri yöntemleri kullanılarak kısaca RNP adında sırasıyla sil ve karşılaştır, Varlık İsmi Tanıma ile arama, Cümle Ögeleri etiketleme ile arama yöntemleri geliştirilmektedir. Bu yöntemlerin kullanılabilmesi için ilk olarak paragraf içinde ilgili cümleyi seçme işlemi gerçekleştirilmektedir. Seçilen cümle üzerinde RNP yöntemleri analiz edilmektedir. Bu analiz tüm veri seti ve BERT dil

modellerinin üzerinde uygulanmıştır. Tüm veri seti üzerinde sorular incelendiğinde RNP yöntemleri yaklaşık %19.9 oranında doğru cevapları tespit etmiştir. BERT modellerinin yanlış cevapladığı sorular üzerinde uygulanan analiz sonucunda ise, RNP yöntemleri eklenti olarak BERT modellerinin doğruluk değerini %1.1 ve %2.4 arasında artırmıştır.

Üçlü tabanlı eklentide ise ontoloji yaklaşımından esinlenilmektedir. Bu yöntem ile, ontolojinin özne-yüklem-nesne üçlüsünden yararlanarak üçlü çıkarımı ile cevabın doğru tespit edilmesi amaçlanmaktadır. İlk olarak yine ilgili cümle seçim işlemi soru terimlerine göre gerçekleştirilmektedir. Seçilen cümle üzerinde üçlülerin çıkarılması ile bu üçlüler arasında aday cevap aranmaktadır. Arama işlemi soru terimleri analiz edilerek gerçekleştirilmektedir. Bu eklenti cevabı olan ve olmayan tüm sorular için uygulanmaktadır. BERT, ALBERT, ELECTRA, RoBERTa ve SpanBERT dil modellerinin yanlış yanıtladığı sorular analiz edilmektedir Analizlerin sonucunda, üçlü-tabanlı eklenti, dil modellerinin doğruluk değerini %3.3 ile %7.5 arasında artırmıştır.

Bu eklentiler dil modellerinin yanlış cevapladığı soruları yanıtlayabildiğini ve doğruluk değerini artırabildiğini göstermektedir. Ayrıca amaçlandığı gibi her iki eklentide hiçbir yeniden eğitime ihtiyaç duymamaktadır. Sadece paragraf ve soru girdi olarak her iki eklentiye verildiğinde, soruları dil modelinden ve veri setinden bağımsız olarak analiz edebilmektedir.

Anahtar sözcükler: Doğal Dil İşleme, BERT, Veri Analizi, Soru Cevaplama, Dil Modeli, SQuAD, Bilgi Çıkarımı.

PREFACE

In this thesis, it is aimed to solve the problems of deep learning models. As a result of our research and the advice of my advisor Prof. Dr. Murat Osman Ünalır, we decided to work on deep learning models using the SQuAD for the question-answering. Many deep learning-based language models such as BERT, ELECTRA used on this dataset have been investigated. The main problem of deep learning models is that when a new data is wanted to be given to the model, this model needs to be retrained, so there is a problem in terms of time and cost. Therefore, it has been wondered whether a question answering system can be developed that takes paragraph and question as input on language models and tries to determine the answer without retraining. As a result of our analysis, a solution to this problem has been proposed by suggesting two extensions. Firstly, three methods were developed by examining natural language processing techniques. Secondly, a method that can determine the answer as a result of triple extraction was proposed as extension. As a result of these analyses, some questions answered incorrectly by the language models were able to be answered correctly in both methods. Thus, each extension both increased the success and were able to answer the question without the need for retraining. In addition, both extensions are dataset independent. I think that these extensions we recommend will lead researchers who will work.

İZMİR
26/09/2022

Zekeriya Anıl GÜVEN

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (continued)

Page

TABLE OF CONTENTS (continued)

Page

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES (continued)

LIST OF TABLES

Table **Page**

LIST OF TABLES (continued)

Table **Page**

LIST OF ABBREVIATIONS

Abbreviations	Description
A	Attribute for RDF
AIoT	Artificial Intelligence of Things
ALBERT	A Lite BERT
BERT	Bidirectional Encoder Representations from Transformers
CQA	Community Question Answering
Dev_set	Development set
DL	Deep Learning
ELECTRA	Pre-training Text Encoders as Discriminators Rather Than Generators
IE	Information Extraction
IR	Information Retrieval
KG	Knowledge Graph
LM	Language Model
LSTM	Long Short-Term Memory
MEMEN	Multi-Layer Embedding with Memory Network
ML	Machine Learning
MLM	Masked Language Model
NER	Named Entity Recognition
NLP	Natural Language Processing
NLP-QAS	Natural Language based QAS
O	Object for RDF
OWL	Ontology Web Language
POS	Part of Speech
QA	Question Answering
QAS	Question Answering System
QTP	Question Term Percentage
RC	Remove & Compare

LIST OF ABBREVIATIONS (continued)

Abbreviations	Description
RCNN	Recurrent Convolutional Neural Network
RDF	Resource Description Framework
RoBERTa	Robustly Optimized BERT
RSS	Related Sentence Selection
SDP	Solved Dataset Problem
SemBERT	Semantics-aware BERT
SG-NET	Syntax-Guided Machine Reading Comprehension
SNER	Searching with NER
SpanBERT	Pre-training by Representing and Predicting Spans
SPOS	Searching with POS
SRO	Subject-Relation-Object
SVM	Support Vector Machine
TRP-QAS	Triple based QAS
Train_set	Training set
V	Value for RDF
W3C	World Wide Web Consortium
XML	Extensible Markup Language

1 INTRODUCTION

Nowadays, due to the continuous increase in the amount of data, the processing and analysis of data becomes difficult. The type of big data can vary according to the area used, such as image, text, audio. There are many research areas such as Image Processing, Natural Language Processing (NLP) for the processing and analysis of these data. NLP is a domain of research used for computers to understand and process natural language text or speech data. NLP researchers aim to gather information about how people understand and use language. Thus, appropriate tools and techniques can be developed to enable computer systems to understand and manipulate natural languages to perform desired tasks (Chowdhary, 2020). With these techniques, many operations such as stemming, morphological processing, syntactic parsing, and analysis are performed in NLP. These developed tools and techniques are used in many NLP fields such as text translation, text summarization, sentiment analysis, question answering (QA), topic modeling, document classification, clustering. Weka[1], Orange Data Mining[2] can be given as examples of analysis tools developed for NLP.

Many machine learning (ML) methods are applied for NLP. The aim of ML is to recognize patterns in data that inform how to handle unseen problems. ML methods are given to the model according to the extracted features. ML tasks are divided into three main categories as supervised, unsupervised and reinforcement learning based on the existing learning (Ballı and Sağbaş, 2018). For supervised learning, classification algorithms in which class labels are given along with the input are used, while in unsupervised learning, algorithms such as clustering are used without class labels. Reinforcement learning is learning in which an agent takes actions in an environment to maximize a reward (Carleo et al., 2019). Deep Learning (DL) is a sub-domain of ML and has made rapid progress in the field of artificial intelligence. It also pioneers the solution of long-

[1]https://www.cs.waikato.ac.nz/ml/weka/index.html

[2]https://orangedatamining.com/

standing problems in many fields such as image processing, except for NLP. Many frameworks are used, such as Caffe (Jia et al., 2014), PyTorch (Paszke et al., 2019), TensorFlow (Abadi et al., 2016), etc., which are supported by DL.

QA, which is the main subject of this study, aims to automatically answer the questions asked in natural language. QA systems can benefit from Wikipedia pages, web pages, document texts, etc. (Chen and Yih, 2020). Datasets such as SQuAD[3], NarrativeSC, and HotpotSC in the QA domain contain questions and answers written by reviewers who read a short text (Kwiatkowski et al., 2019). Therefore, how to choose the best answer from the numerous candidate answers for a given question makes it important in QA research. SQuAD, which is used in this study, is a reading comprehension dataset created by Stanford University from Wikipedia pages. The answers to the questions are not multiple choice for SQuAD. This dataset was obtained with the answers given by the crowdworkers. There is also a benchmark platform for SQuAD, where many models are used and these models are compared. Models including many ML and DL methods are used in the benchmark platform. Language models (LMs) that analyze bodies of text data to predict answer are quite often used in QA systems for SQuAD. LMs based on DL are trained with a training set and analyzed according to test set. In the SQuAD benchmark platform, many LMs are used that use the context of words such as BERT (Devlin et al., 2018), RoBERTa Liu et al. (2019), ALBERT(Lan et al., 2019) or do not use the context, such as Transformers (Vaswani et al., 2017).

Since LMs are based on deep learning, the model must be trained from the beginning when any new data is given to the model. Marcus (2020) indicated that whenever a new sentence was added to any article for SQuAD, the LM had answered the question incorrectly and the model had to be retrained.

[3]https://rajpurkar.github.io/SQuAD-explorer/

retraining the model creates problems in terms of time and cost. In this study, instead of retraining the model, two methods are proposed as extensions to solve these problems. These methods are respectively NLP-based methods and triple-based method inspired by ontology.

- First, an NLP-based QA system (NLP-QAS)[4] is proposed as an extension. NLP-QAS includes 3 proposed NLP methods (Remove and Compare, Search with NER, Search with POS tagging), namely RNP, for answer detection on pre-trained LMs. RNP methods are developed using textual techniques such as Part of Speech (POS) tagging and Named Entity Recognition (NER). In the proposed system, firstly, the sentences in the related paragraph are analyzed according to the question terms, and the most appropriate sentence is selected according to the ratio of the question terms in this sentence. RNP methods are applied to the answer detection to the selected sentence. Then, the effect of these methods on the questions answered incorrectly by the pre-trained BERT models was analyzed (Guven and Unalir, 2022).
- Secondly, a QA system (QAS) extension inspired by ontology triples is proposed. The proposed QAS based on this triples (TRP-QAS)[5] utilizes subject-predicate-object triples in ontology. The triples are extracted from the related paragraph or sentence according to the question via the StanfordOpenIE-python[6] library and question terms are searched within these triples. As a result of the analysis applied according to the situation of the question terms being in triples, it is determined whether there is a candidate answer among the triples. The effect of triples on SQuAD is analyzed by analyzing the success of LMs in the literature. In the analysis phase, firstly, the questions that the previously trained BERT, ELECTRA, ALBERT, SpanBERT and RoBERTa LMs answered incorrectly are obtained. The success of TRP-QAS on these questions

[4]https://github.com/anil1055/NLP-based_QAS
[5]https://github.com/anil1055/Triple-based_QAS
[6]https://github.com/philipperemy/stanford-openie-python

is then analyzed. As a result of the analysis, the contribution of the proposed TRP-QAS to the LMs in answering the questions is shown.

The main contributions of our study to the literature are:

- An NLP-QAS extension is proposed for answer detection, which includes sentence selection and NLP-based methods.
- NLP-QAS is a pioneering work that expands the capacity of the BERT model in QASs with RNP methods.
- Experimental results show that the proposed NLP-QAS and RNP methods improve performance on questions that the BERT model cannot answer.
- TRP-QAS based on ontology triples has been proposed as an extension to the SQuAD analysis. Experimental results indicate that the proposed TRP-QAS performed well on questions that LMs could not answer.
- This study is the first to increase the capability of LM with NLP-based RNP methods and triple-based system for SQuAD.
- With both proposed QASs, there is no need for retraining. Both QASs can be also used independently of the dataset, as they focus on questions that LMs have answered incorrectly.

This thesis is structured as follows. The explanation of the main topics such as NLP, QA, ontology and ML used in this study is realized in Section 2. Literature searches on SQuAD, QA and LMs are described in Section 3. Section 4 describes the dataset of this study, the LMs, and the libraries utilized in this study. Section 5 contains a detailed description of the methodologies for both QASs. Section 6 shows the operation of these extensions and the page contents of the designed SQuAD Explorer. Analysis of SQuAD and pre-trained LMs, analysis and discussion of NLP-QAS and TRP-QAS are carried out under Section 7. Finally, Section 8 includes the conclusions of both QASs of this study and informing about future works.

2 BACKGROUND KNOWLEDGE

In this section, the concepts of NLP, QA, ontology, ML and LMs, which are the main topics of this study, are explained in detail under sub-headings.

2.1 Natural Language Processing

NLP is a research field of computer science, artificial intelligence and linguistics that explores the connection between computers and human language. Natural languages are spoken languages that people use to communicate by learning from their environment. Whatever the form of communication, natural languages express our knowledge and emotions and elicit our reactions (Reshamwala, A., Mishra, D., & Pawar, 2013). NLP involves the use of computers to recognize and understand natural languages. It is linked to text mining and corpus linguistics. It also focuses on many areas such as knowledge representation, and logical reasoning (Abdullah Alfaries et al., 2017).

NLP is basically divided into Natural Language Understanding and Natural Language Generation stages. Natural Language Understanding was developed for the task of understanding the text, while Natural Language Generation was developed for the task of creating the text. The sub-stages of these stages are shown in Figure 2.1. From the linguistic levels of text understanding in the figure, phonology refers to sound, while morphology refers to word formation, syntax refers to sentence structure, semantic refers to syntax and pragmatic refers to meaning (Khurana et al., 2022). The ability of an NLP system is determined by how many linguistic levels it uses. When the linguistic levels in understanding the text are mentioned in detail, respectively (Reshamwala, A., Mishra, D., & Pawar, 2013):

- Phonology: It is used to interpret speech sounds within and between words. Sound waves are analyzed by giving sound input to the NLP system. Sound waves are encoded into digitized signal for use with rules or LM.

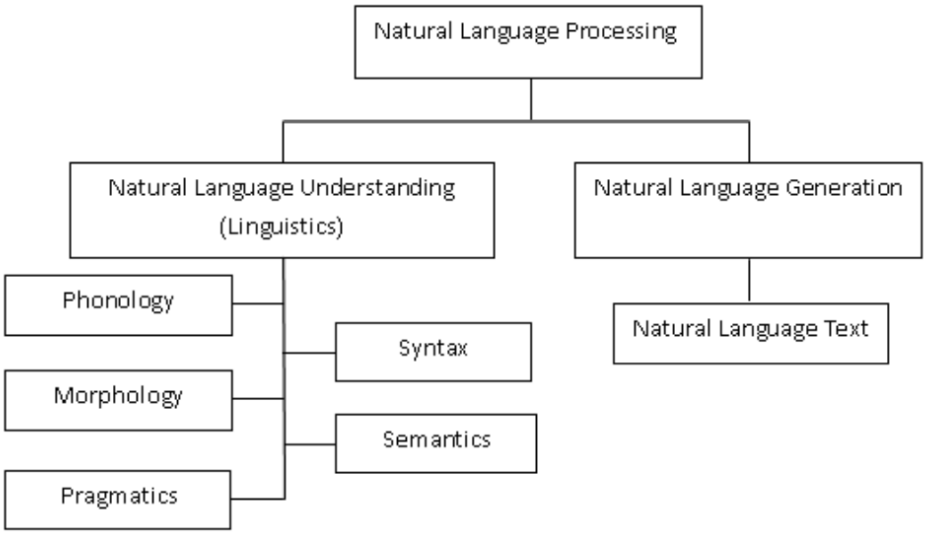


Figure 2.1: Classification of NLP (Khurana et al., 2022)

- Morphology: It is the first stage of the analysis after the input is taken into the system. It explores the decomposition of words into their components and how this process affects grammatical situations. It is useful for identifying parts of speech in a sentence and words that interact together. Morphology gives a systematic description of words in a natural language. It describes a series of relationships between the surface forms of words and their lexical forms. The surface form of a word is its graphical or verbal form. The lexical form is the analysis for the word's lemma and grammatical explanation. For these operations, syntax rules that use the grammar rules of the target language are applied. The syntax uses the grammar and parsing structure of the language.
- Semantics: It constitutes a representation of objects and actions in a sentence. It includes details obtained by adjectives, adverbs and propositions. This process determines what meaning is intended by the user.
- Pragmatics: It is the stage of analyzing the true meaning of an utterance by contextualizing it after removing the ambiguity. The ambiguity faced by the NLP system need to be identified. Certain techniques should

be used to resolve these ambiguity. These ambiguity are divided into three as semantic, referential and local. Semantic ambiguity is the situation where there is more than one possible meaning for a sentence. Referential ambiguity is the indefiniteness of expressions such as the referred pronoun. Local ambiguity, on the other hand, is the situation where the meaning cannot be determined when a part of a sentence is examined, but it is resolved when the sentence is examined as a whole.

In the Natural Language Generation phase, meaningful sentences or paragraphs are produced from an internal representation. This process takes place in four stages. First, goals are set. Then, by evaluating the situation, it is necessary to plan how the goals can be achieved and the available communication resources. In the last stage, it is ensured that the plans are realized as a text (Khurana et al., 2022).

NLP is divided into two broad areas of study: core and application areas. Core areas include basic topics such as language modeling, which emphasizes quantitative relationships between words. Morphological processing, syntactic parsing and semantic processing are examples of core domain processes. Application areas are concerned with extracting useful information such as named entities and relationships, translating text between languages, document summarization, automatically answering questions by inference, classifying and clustering documents, etc (Otter et al., 2018). NLP is used in processes such as sentence segmentation, NER used to find special entities (person, place, time, etc.) in the text, POS tagging, which can separate sentences (object, name, subject, adverb, etc.) in the text, information extraction, semantic role labeling, etc.

2.2 Question Answering

QA refers to a certain type of information retrieval. It is a multidisciplinary field of NLP. When documents and question are given as input to the QAS, this

system aims to bring the correct answer for the question in natural language. Natural or statistical language processing, information retrieval and knowledge representation, and reasoning are important constructs used by QASs. The user of a QAS wants the short, clear and correct answer that the QAS can find (Kolomiyets and Moens, 2011). The main purpose of the QAS is to find the correct answer for interrogative pronouns such as ”who, when, where, why, how”. QASs combine methods of information retrieval and extraction to determine the most appropriate answer, using some sorting method to generate candidate answers (Gupta and Gupta, 2012).

QASs are divided into two main categories, open and closed domain QA. Open domain QA is of a general structure, examining questions on each topic. This domain is based only on universal ontology structures and knowledges. Closed domain QA, on the other hand, is narrower than open domain QA. It's working on questions related to a specific field (music, sports, etc.). Since it is used in a certain field, it can use a large number of field-specific ontologies (Allam and Haggag, 2012).

A general QAS consists of three separate stages: question classification, information retrieval (document processing), and answer extraction. These stages of the QAS are shown in Figure 2.2 as a module. When the stages in this figure are examined, firstly, the question processing stage determines the question focus and classifies the question type, then this stage determines the type of candidate answer. It also obtains semantically equivalent questions from the question. In the document processing stage, paragraph indexing is performed. At this stage, the question is sent to the information retrieval system and receives an ordered list of relevant documents. A document may use one or more information retrieval systems to gather information from a corpus collection. These documents are then filtered and sorted. Finally, in the answer processing module, the processes of determining, extracting and verifying the answers within the set of ordered paragraphs are applied (Allam and Haggag, 2012).

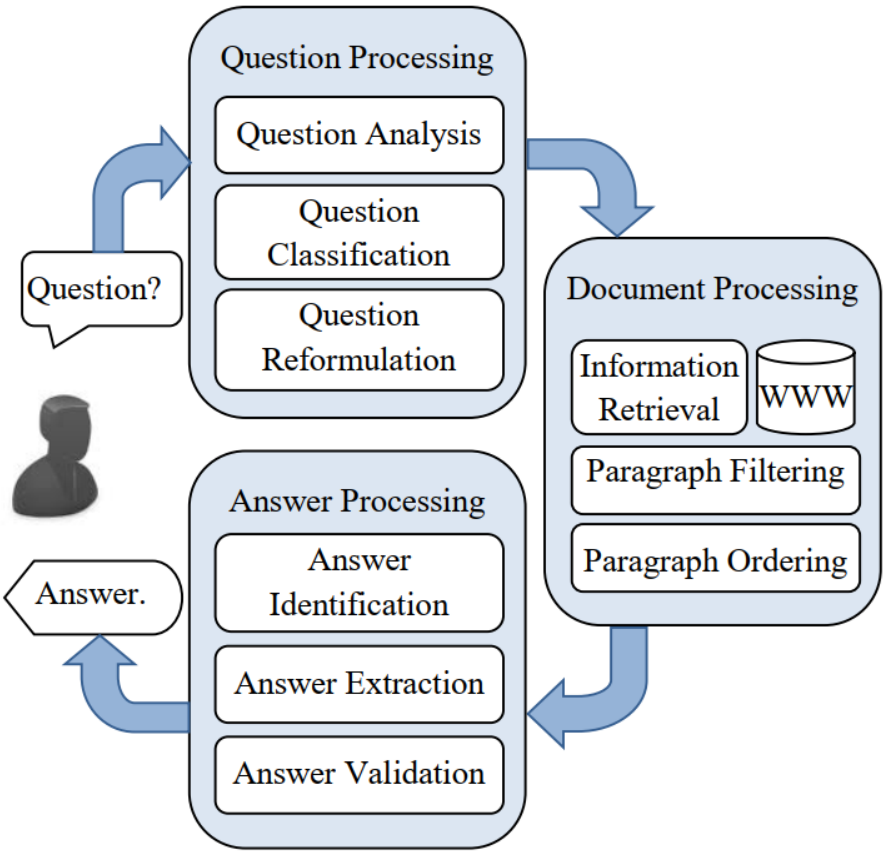


Figure 2.2: QAS Architecture (Allam and Haggag, 2012)

The following processes are performed for the QAS, respectively (Allam and Haggag, 2012):

- The user sends a question to the QAS. The focus of the question is determined by the question analyzer.
- The type of question and candidate answer are determined through question classification.
- Question expansion and reformulation process is applied. Then, the relevant documents are selected by using the keywords in the question for information retrieval.
- These selected documents are filtered and the paragraph expected to contain the answer is shortened. These filtered paragraphs are sorted and forwarded to the answer processing stage.
- Candidate answers are determined according to the answer type and

other definitions. A heuristic set is defined to extract the corresponding answer among the candidate answers.

- The correctness of the extracted answer is confirmed and presented to the user.

QASs are divided into three categories: linguistic, statistical and pattern matching approach. The linguistic approach is applied with techniques such as tokenization, POS tagging and parsing to formulate the user's question into a precise query that only extracts the relevant answer from the structured database. This approach is often used for problems with long-term information needs for a particular field. Statistical approaches are independent of structured query languages and can formulate queries in natural language format. However, this approach cannot define linguistic features as it uses each term independently. Statistical approach uses Support vector machine (SVM), Bayesian, Maximum entropy classifiers for question classification. These statistical approaches analyze the question to predict answer type. The pattern matching approach, on the other hand, uses the expressive power of text patterns. Most QASs automatically learn text patterns from paragraphs by pattern matching rather than using ontology, WordNet. Pattern matching reduces linguistic computations and helps to process heterogeneous web data (Dwivedi and Singh, 2013).

2.3 Ontology

Knowledge-based methods try to think like a human and act like an expert system. It can evaluate data in an additive and semantic way like human learning. It is expected that the use of knowledge-based methods will increase in the future. Ontology can be given as an example as a knowledge-based method (Noy and McGuinness, 2001). Ontologies play a crucial role in enabling Web-based information processing, sharing, and reuse across applications. Ontology is defined as shared conceptualizations of specific domains. An ontology typically contains a hierarchy of concepts within a

domain. It identifies the important features of each concept through an attribute-value mechanism. Other relationships between concepts are indicated through additional logical sentences. Finally, constants (such as ”January”) are assigned to one or more concepts (such as ”Moon”) to determine appropriate types (Decker et al., 2000). An ontology with a set of individual concept examples constitutes a knowledge base (Noy and McGuinness, 2001).

Ontology defines a common vocabulary for researchers who can share knowledge for any domain. It includes definitions of the basic concepts in the field so that they can be interpreted by the machine. In addition, the relations between these concepts are included in the ontology. There are several reasons for developing an ontology (Noy and McGuinness, 2001):

- Achieving shared understanding of the structure of knowledge among people or software agents
- Reuse for domain information
- Clarifying assumptions about the domain
- Separating domain knowledge from operational knowledge
- Analyzing domain information

Classes are the basis of most ontologies. Developing an ontology requires defining the classes in the ontology. Classes describe concepts in the domain. For example, a wine class represents all wines. Specific wines are examples of this class. Bordeaux wine in a glass is an example of the Bordeaux wine class. A class can have subclasses that represent more specific concepts than the superclass. We can divide all wines into three classes as red, white and rosé or, differently, subclass all wines as sparkling and still wine. An example ontology is shown in Figure 2.3. Relationships and related concepts of ”Château Lafite Rothschild Pauillac” wine are given in the figure. Ontology organizes classes in a taxonomic hierarchy. It also defines slots and explains the allowed values. It then fills in the values of the slots for the samples (Noy and McGuinness, 2001).

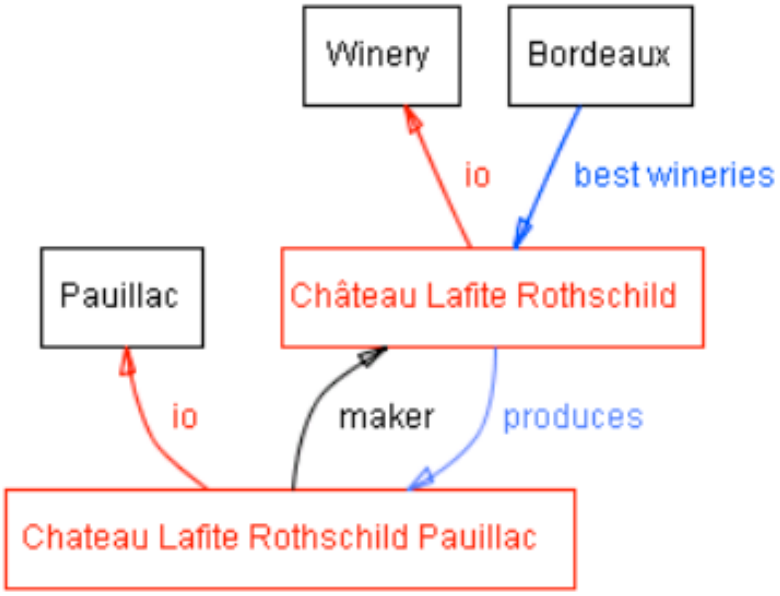


Figure 2.3: An ontology example (Noy and McGuinness, 2001)

Certain steps are required to develop an ontology. In the first stage, the domain and scope for ontology creation should be determined. Then, for this ontology, it should be investigated whether the requirement is met on the existing ontologies. If a new ontology is to be created, the classes to be included in the domain-specific ontology should be defined. The hierarchy of these classes should be arranged as taxonomic (subclass-superclass). Then the slots must be defined and the allowed values for these slots must be specified. Finally, the values of the slots for the samples must be filled. Thus, instances of classes can be defined and knowledge base created as certain slot value information and additional slot constraints are filled. (Noy and McGuinness, 2001).

Ontologies are divided into two types; transcendent and immanent: (Jepsen, 2009).

1. Transcendent ontology: This ontology is defined externally from the authorized and used applications. The periodic table used in chemistry can be given as an example, because this table has been obtained through

long-standing scientific knowledge. This ontology is rarely modified. For example, if all chemists in the world confirm the discovery of a new element, it can be added to categories on the periodic table.

2. Immanent ontology: It is obtained from the information content of the domain. The ontology of all items in a daily newspaper can be an example of this ontology type. The structure of the newspaper will change every day depending on the news of that day. One day's ontology may contain articles on sporting events, and the next day a global crisis report.

Extensible Markup Language (XML) and Resource Description Framework (RDF) are each basis standards for the Semantic Web. XML only handles document structure, while RDF provides a data model that can be extended to handle complex ontology representation techniques. Therefore, RDF provides more suitable mechanisms for the interoperability of ontology representation languages such as Ontology Interchange Language. RDF is the World Wide Web Consortium (W3C) proposition designed to standardize the definition and use of metadata, which are definitions of Web-based resources. RDF basically contains the object(O)-attribute(A)-value(V) triple written as A(O,V). In this relationship, an O object has an attribute A with the value V. This relationship can be defined differently, as a labeled edge A between two nodes, O and V: [O]–A→[V]. Figure 2.4 provides an example RDF triples. According to the figure, an employee with "id132" is named "Jim Lerners", this employee is the author of the book with the ISBN number, and the fee for this book is "62$" (Decker et al., 2000).

Ontology representation languages are recommended by prestigious organizations such as the W3C. Ontology Web Language (OWL) is a standardized language developed for ontology representation (Giray and Ünalir, 2013).

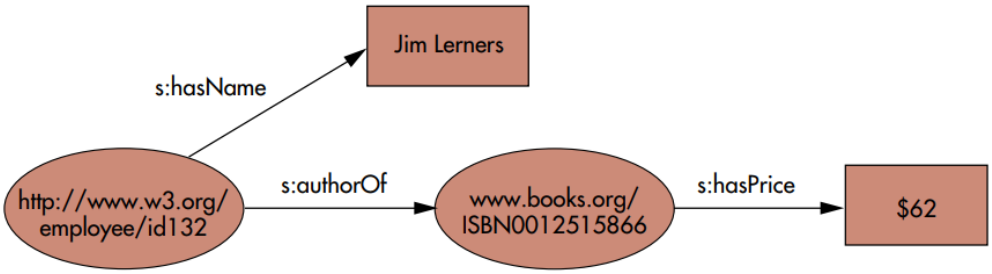


Figure 2.4: The RDF triples example (Decker et al., 2000)

2.4 Machine Learning

With the abundance of available datasets and the abundance of data, the demand for ML is increasing day by day. Many industries are applying ML to extract and analyze relevant data. The purpose of ML is to enable learning from data (Mahesh, 2020). ML studies how to use computers to simulate human learning activities. It also explores the self-development of computers to acquire new knowledge and new skills, identify existing knowledge, and improve performance (Wang et al., 2009).

Compared to human learning, ML learns faster because knowledge accumulation is easier than human. Thus, the results of learning spread more easily. As a result, according to human progress in ML, the capacity of computers will increase and it will have an impact on society (Wang et al., 2009).

ML uses different algorithms to solve data problems. These algorithms may vary depending on the type of problem, the number of variables, the type of the most suitable model, etc. There are many algorithms used in ML (Mahesh, 2020):

1. Supervised Learning: It is a learning method that maps an input to an output based on sample input-output pairs. It outputs a function using a set of labeled training data. The dataset to be used is divided into two as training and testing. The training set has an output variable that needs to be predicted or classified. All algorithms learn patterns from

the training set and apply this pattern to the test set for prediction or classification. The structure of supervised learning is shown in Figure 2.5. Naive Bayes, SVM, Decision Tree algorithms can be given as examples of supervised learning.

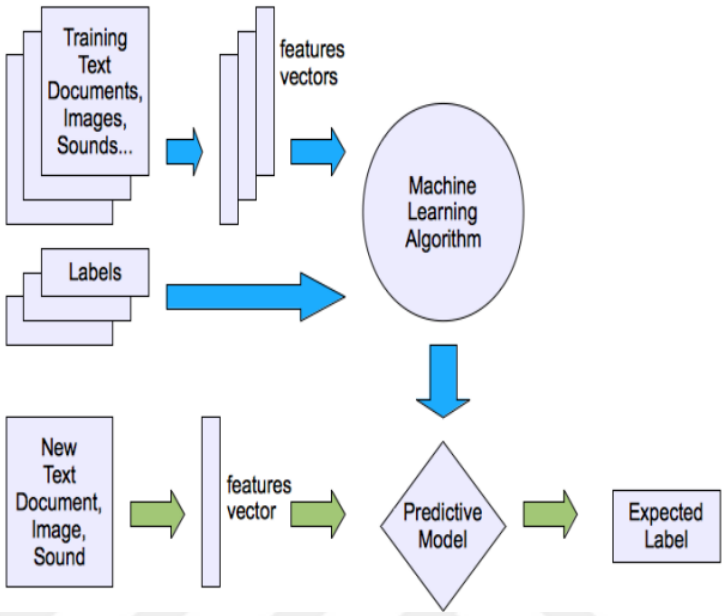


Figure 2.5: The structure of supervised learning (Jayanthi and Mahesh, 2018)

2. Unsupervised Learning: These algorithms learn several features from data without using labeled data. When new data arrives, it uses previously learned features to determine the class of data. The structure of unsupervised learning is shown in Figure 2.6. It is mainly used for clustering and feature reduction. Principal Component Analysis and K-Means can be cited as examples.

3. Semi-supervised Learning: It is a combination of supervised and unsupervised ML methods. It is used in the fields of machine learning and data mining where unlabeled data is available and obtaining labeled data is a very intensive process. Transductive SVM is an example.

4. Reinforcement Learning: It identifies how to take action to maximize some concept of cumulative reward. Reinforcement learning is one of the three key ML paradigms alongside supervised and unsupervised learning.

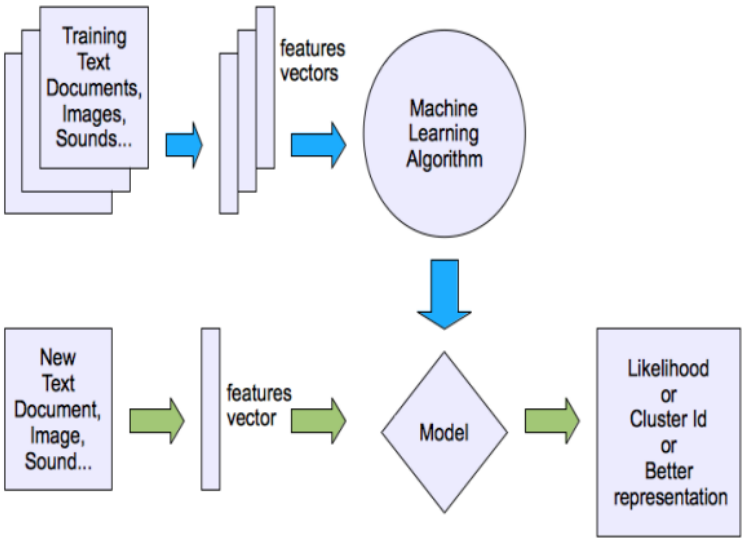


Figure 2.6: The structure of unsupervised learning (Jayanthi and Mahesh, 2018)

5. Ensemble Learning: It is the learning that enables to create the model with more than one learner instead of training the model with one learner. It aims that the models will make more accurate decisions together. Bagging and Boosting methods can be given as examples.

6. Neural Network: It mimics the learning, remembering and generalizing abilities of the structure of biological neural networks in the brain. It is used to automatically perform abilities such as the ability of machines to derive new information, to create and discover new information, by taking advantage of the learning path in the human brain.

		Predicted	
Actual		Negative	Positive
	Negative	TN	FP
	Positive	FN	TP

Figure 2.7: The confusion matrix (Kulkarni et al., 2020)

There are many performance metrics for ML. The confusion matrix is a measure used for classification algorithms from ML methods. The confusion

matrix is shown in Figure 2.7. In this figure, TN indicates the number of correctly classified negative samples, and TP indicates the number of correctly classified positive samples. FP indicates the number of true negative samples misclassified as positive, and FN represents the number of true positive samples misclassified as negative (Kulkarni et al., 2020). Using the confusion matrix, accuracy, precision, recall and F1-score performance measures are calculated through TP, TN, FP and FN. Accuracy is the ratio of the number of all correctly classified samples (TP+TN) to all samples (TP+TN+FP+FN). Precision is the ratio of actually positive correctly classified samples (TP) to the total samples predicted to be positive (TP+FP), while recall is the ratio of correctly classified samples (TP) to the total number of actually positive samples (TP+FN). The F1-score calculates the balance between precision and recall (2*precision*recall /(precision+recall)). The Exact Match (EM) metric is also used for QA. For each question-answer pair, EM = 1 if the characters of the model's prediction exactly match the characters of correct answer(s), EM = 0 otherwise.

2.4.1 Deep Learning

DL is used for any task by applying deep neural networks to large amounts of data. This method focuses on producing an ideal solution to any problem. It is a field that is compatible with artificial intelligence. Artificial intelligence enables a machine to outperform the human brain, while deep learning is a tool for this purpose (Torfi et al., 2020).

Numerous DL architectures have been developed in NLP applications using recurrent neural networks, convolutional neural networks, and iterative neural networks. DL applications are based on architecture as well as feature representation and deep learning algorithm options. These are associated with data representation and learning structure, respectively (Torfi et al., 2020).

NLP's progress often depends on effective language modeling. Statistical language modeling aims to extract probabilistic representations of word strings in language due to the size problem. Obtaining an in-depth representation of language using statistical models for NLP is a major challenge. The primary task in NLP applications is to provide representation of texts such as documents. Text representation is the process of extracting meaningful information for further processing and analysis of raw data. Many deep learning-based LMs have been used for NLP tasks lately. New LMs such as BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), ALBERT (Lan et al., 2019), ELECTRA (Clark et al., 2020) consist of certain layers such as encoder and decoder. LM provides context to distinguish similar words and phrases. It is divided into bidirectional and unidirectional. The unidirectional LM assigns a probability based on the factorization order given the input order. An example of a unidirectional LM is the Transformers model. Transformers relies solely on attention mechanisms, completely abandoning iteration and convolution. The Transformers architecture scales with training data and model size, facilitates parallel training, and captures long-range array properties (Wolf et al., 2019). The Transformer is able to use a longer history by caching previous outputs and using the offset. Bidirectional LMs, on the other hand, assign a probability to the sequence using the word's input order, position, and left-right context. ELMo and BERT are examples of this model. ELMo works with feedforward and backpropagation Long short-term memory (LSTM) to estimate probability. ELMO uses multiple layers of LSTM (Petroni et al., 2020). BERT is a bidirectional representation of the Transformers model. Rather than looking at a single context of the word, it analyzes both left and right contexts. It aims to mask random words and predict these words using the masked language model (MLM) structure (Devlin et al., 2018). In addition, many models such as RoBERTa, ELECTRA, SpanBERT and ALBERT based on the BERT model are used in this study.

3 RELATED WORKS

QA provides answers to questions in natural language. Since the same information can be expressed differently in natural language, it is possible to produce different answers in semantically equivalent questions (Dong et al., 2017). Therefore, QA is a rather difficult NLP research area. In recent years, many studies have focused on this area to enable computers to automatically answer questions in natural language on any topic. ML and DL methods are used in these studies. In this section, studies in the literature are explained. The literature studies for the SQuAD, LMs and QA domain is described under sub-headings.

3.1 SQuAD

Many models have been developed for performance measurement on SQuAD. There is also a benchmarking platform that enables comparison of these models. Among these models, the BERT model is based on SQuAD. Devlin et al. (2018) proposed the BERT model named transformer-based bidirectional encoder representation. The model consists of pre-training and fine-tuning stages. In the first stage, the model is trained with unlabeled data, and in the second stage, all parameters are fine-tuned throughout the downstream task. The F1-score value of BERT on SQuAD was 93.2% and 83.1% for SQuAD 1.1 and SQuAD 2.0, respectively.

Zhang et al. (2019b) proposed a syntax-driven network (SG-Net) model based on the pre-trained BERT model. They aimed to achieve better word representation by adding explicit syntactic constraints to the model with the parse tree structure. Their SG-Net model achieved 87.9% F1-score for SQuAD 2.0.

Yang et al. (2019b) proposed XLNet, a generalized autoregressive pre-training method. Since the BERT's MLM method and the model lack real data and fine-tune inconsistency, the XLNet model aimed to solve this

problem. This model allows learning two-way contexts by maximizing the expected probability in all permutations of factorization. They suggested using two hidden representation sets in the structure of XLNet model. This model obtained 95.1% and 90.6% F1-scores for SQuAD 1.1 and SQuAD 2.0, respectively.

Zhang et al. (2019a) proposed a semantic sensitive model called SemBERT based on the BERT model. The structure of their models consists of a semantic role labeler, a set of coders, and semantic integration components. They analyzed the model using the same weights and fine-tuning procedures as the BERT model. When the SQuAD 2.0 results were examined, their SemBERT model reached 87.9% F1-score.

By optimizing the BERT model, Liu et al. (2019) proposed a more powerful model, namely RoBERTa. For this, they used dynamic masking method and longer arrays instead of larger training set and static masking. They also removed the next sentence prediction process in the BERT model. They analyzed the success of the model on SQuAD and obtained 94.6% and 89.4% F1-score values for SQuAD 1.1 and SQuAD 2.0, respectively.

Because of the memory limit and communication overhead problem in the BERT model, Lan et al. (2019) proposed the ALBERT model with fewer parameters. Two parameter reduction techniques, factorized embedding and parameter sharing, were used in this model. They also modeled inter-sentence consistency with the self-monitoring loss function. As a result of the application of ALBERT model, they showed that the number of parameters was considerably reduced compared to BERT, and the performance improved. As a result of the analysis, this model reached 89.7% F1-score for SQuAD 2.0.

Joshi et al. (2020) proposed the SpanBERT model to better represent and predict text spans compared to the BERT model. They applied masking with adjacent random spans rather than random markers included in the BERT

model. They also trained span boundary representations to predict the entire content of masked spans. As a result of the analysis, the SpanBERT model outperformed BERT, reaching an F1-score of 94.6% for SQuAD 1.1 and 88.7% for SQuAD 2.0.

Zhang et al. (2020) proposed the Retro-Reader model, which combines a two-stage reading and validation strategy. Their strategy is the rough reading that first briefly explores the SQuAD paragraph, while the second is the intensive reading that provides the prediction. They tested their proposed model on SQuAD 2.0 and obtained 90.9% F1-score.

Yamada et al. (2020) proposed the LUKE model, which includes a new pre-training task based on the BERT model. The model takes words and entities as independent tokens and extracts contextual representations of these tokens. They also proposed a self-attention mechanism that can be aware of the presence. Their model achieved 95% F1-score on the SQuAD 1.1.

Clark et al. (2020) proposed the ELECTRA model to pre-train transformer networks using less computation than BERT. It has been applied to Transformers text encoders. ELECTRA models distinguish between ”actual” input tokens produced by another neural network and ”fake” input tokens. The main idea is to train a text encoder to distinguish input tokens from high-quality negative samples produced by a small transformer network. When the analysis for SQuAD 2.0 was performed, it reached an 91.4% F1-score.

Hu et al. (2018) developed the attention reader method and proposed the Reinforced Mnemonic Reader model. In the first stage, they established a re-attention mechanism that reduces attention deficit and excess problems in multi-round alignment architectures. They then developed the dynamic-critical reinforcement learning approach to address the problem of convergence suppression. As a result of their proposed model, this model obtained 88.5% F1-score for SQuAD 1.1.

Pan et al. (2017) developed a Multi-Layer Embedding with Memory Network (MEMEN) neural network architecture because the vectors used in past attention methods underestimated the weight of keywords in the query sentence. In the coding layer of the model, they used the classical skip-gram model for the syntactic and semantic knowledge of the words. They also proposed a memory network to capture important information from texts. They derived this memory network from exact routing matching of query and snippet. With their proposed MEMEN model, they achieved 82.66% F1-score for SQuAD 1.1.

Liu et al. (2017) proposed a phase conductor (PhaseCond) framework for attention models in two significant ways. This framework consists of multiple phases that implement a stack of attention layers and an internal or external stack of fusion layers that regulate the flow of information. They also coded the question and passage embedding layers from different perspectives and improved the dot-product attention function for PhaseCond. With the PhaseCond framework, they achieved 84.0% F1-score for SQuAD 1.1.

Xiong et al. (2018) proposed a hybrid target combining traditional cross-entropy loss with reinforcement learning and a trained measure of word overlap. For the target, they used the rewards resulting from word overlap to resolve the misalignment between the evaluation metric and the optimization target. They also improved dynamic coattention networks. As a result of the experiments, the model performed well for long questions, between question types and input lengths. At the analysis stage, this model reached 86.0% F1-score for SQuAD 1.1.

Huang et al. (2018) thought that an approach that uses all the information from the word placement level to the highest level representation would be more successful in answering the question. Therefore, they proposed a model called FusionNet. Since models using neural networks in all representation layers are difficult to learn, they obtained an attention scoring function using

these layers with less training overhead. They used this attention function at multi-level layers of context. As a result of the analysis performed on SQuAD 1.1, they obtained 85.9% F1-score.

Liu et al. (2018) proposed a multi-step neural network model called stochastic answer network for QA. During the training of the model, they changed the number of reasoning steps. In addition, stochastic dropout process was applied in the last layer estimations in the answer generation module. During decoding, they produced answers that consider the average of the estimates in all steps rather than the last step. While their proposed model refines the estimation on successive steps, each step is still trained to produce the same answer. As a result of the experimental studies, they showed that it significantly increased the robustness of the model and the accuracy value for the datasets. When the results for SQuAD 1.1 were analyzed, 86.49% F1-score value was measured.

Salant and Berant (2018) investigated the effect of context on the reading comprehension task. For this, they proposed a neural module that examines the positive effect of context use by separating contextual and non-contextual representations. With this module, they have implemented token embedding by switching between contextual and non-contextual representations. By adding the recommended module to a pre-trained language model, contextual information is transferred to the model. They analyzed their modules in the development set for SQuAD 1.1 and obtained 84% F1-score.

Hu et al. (2019) proposed a useful read-then-verify system in identifying unanswered questions for the reading comprehension task. The system they propose calculates probabilities for unanswered questions in addition to extracting candidate answers for questions. In addition, they used an answer validator in the system, which decides whether the candidate answer according to the passage and question in the input text is required. They analyzed their proposed system on SQuAD 2.0 and obtained 74.2% F1-score.

Wang and Jiang (2020) investigated the integration of reading comprehension models with neural networks of people's knowledge. They proposed a method using WordNet to extract semantic connections from each passage-question pair. In addition, they have developed an end-to-end model called Knowledge Assisted Reader, which can use this extracted information. In the analysis phase of the models, it gave an F1-score value of 83.5% for SQuAD 1.1.

To determine whether the question is answerable, Sun et al. (2018) proposed the U-Net unified model, which can be learned end-to-end. The U-Net model consists of three components: an answer marker that predicts candidate answers, an unanswered marker that does not select a text range when there is no answer, and an answer confirmer that indicates non-response to questions. As a result of the universal node they developed, they obtained the question and the context transition with a single adjacent token. The answerability of the question is learned by advancing this node on both the question and the passage. They tested the U-Net model on SQuAD 2.0 and obtained 72.6% F1-score.

Ram et al. (2021) proposed a new pre-training phase called iterative range selection. When the snippet with multiple set of repeating spans is given to the model, they mask all the repeating spans in each set except one span. With this process, they aimed to select the correct span in the passage for each masked span. Masked spans are replaced with a custom marker to determine the response range and trained the system with markers. They tested this training phase for SQuAD 2.0 and measured 72.7% F1-score.

3.2 Question Answering & Language Models

Pre-trained LMs are also used in most SC studies, with the exception of SQuAD. These studies have also been applied in fields such as medicine, law, etc. QA studies of LMs used for other fields are explained under this heading.

Chakraborty et al. (2021) proposed a text-based data mining tool that enables literature search for researchers in the biomedical domain. For this, they developed a neural-based deep contextual model based on the BERT model for QA. In the training phase, they benefited from the BREATHE[7] dataset, which is one of the largest datasets in the biomedical domain. As a result of the analysis, they achieved the most advanced results in QA fine-tuning tasks.

Yoon et al. (2020) analyzed the success of BioBERT, a pre-trained biomedical LM, on biomedical questions. The questions consist of factoid, list and yes/no questions. They performed best in the 7th BioASQ Challenge (Task 7b, Phase B) when the model's success was measured.

To enable generalization to different QA tasks, Su et al. (2019) proposed a structure that can learn the representation shared between tasks. They applied to a pre-trained LM. The success of their model was measured by fine-tuning many datasets. As a result of the analysis, they showed that the pre-trained model they proposed was more successful than the BERT model.

Since there are very few pre-trained LMs that can answer the questions in the field of artificial intelligence of things (AIoT), Zhu et al. (2022) proposed the pre-trained RoBERTa AIoT, which makes up for this shortcoming. They have created an AIoT corpus for the pre-training phase of the RoBERTa and BERT language models. For this, they utilized from the AIoT-oriented Wikipedia web pages. Experimental results of the created models showed significant improvements for AIoT.

Beltagy et al. (2019) proposed SciBERT, a pre-trained LM based on BERT, to address the lack of scientific data. They carried out the pre-training phase on a large multi-domain scientific publications community. Compared to BERT, their proposed model has been quite successful.

[7]https://www.breathedatahub.com/

Zhou et al. (2018) proposed an recurrent convolutional neural network (RCNN) for answer selection in community question answering (CQA), a community-driven online QA website. Their proposed method combines convolutional neural networks with recurrent neural network to capture both the semantic match between the question and the answer and the semantic correlations embedded in the answer string. As a result of the analysis, they showed that RCNN improved on the basic model.

Martinez-Gil et al. (2019) proposed a new method for automatic answering of multiple choice questions. As a result of an empirical evaluation applied on a dataset of legal questions, they showed the positive effect of the proposed method.

Esposito et al. (2020) proposed a hybrid Query Extension approach based on lexical sources and word embeddings to improve the retrieval of related sentences from documents. First, they took the synonyms and hypernyms of the related 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 was sorted and filtered by wording and the meaning of the question.

Yeh and Chen (2020) proposed an alternative approach for the QA system called QAInfomax, which aims to help models avoid getting caught in surface biases in data during learning. For this, they maximized the mutual knowledge between paragraphs, questions and answers. The proposed QAInfomax achieved the best performance on the AdversarialSQuAD dataset without additional training data.

Yasunaga et al. (2021) proposed an end-to-end model called QA-GNN, using LMs and knowledge graphs (KG) for QA. In the proposed model, the fitness score that calculates the fitness level of the conditional QA nodes according to the QA context, and the joint reasoning that connects the QA and KG

resources with the training graphs, and updates the representations jointly in the message transition, is developed. As a result of the analysis, they showed that QA-GNN gave better results than the existing LM and LM+KG models for CommonsenseQA, OpenBookQA, MedQA-USMLE datasets.

Yang et al. (2019a) developed an end-to-end QA system called BERTserini that integrates BERT with the open source Anserini information retrieval (IR) library. The BERTserini system integrates the BERT-based reader from IR to identify answers from the Wikipedia article corpus. With its two-stage pipeline architecture, its systems have improved performance. When a fine-tuning step was applied to the pre-trained BERT, they determined the answer spans with high accuracy.

Carrino et al. (2020) developed the Translate Align Retrieve method to automatically translate SQuAD 1.1 into Spanish. They then used this dataset to train the fine-tuned multi-BERT LM. As a result, they analyzed the models with the MLQA and XQuAD benchmarks used across languages. As a result of their analysis, they obtained 68.1% F1-score in the Spanish MLQA corpus and 77.6% F1-score in the Spanish XQuAD corpus.

Inspired by SQuAD, Möller, Timo; Reina A.; Jayakumar, Raghavan; Pietsch (2020) created a COVID-QA dataset of 2019 question/answer pairs using articles on COVID-19. They performed the analysis by training the RoBERTa model, which was fine-tuned on SQuAD, with COVID-QA. As a result of experimental studies, they obtained 59.53% F1-score for COVID-QA.

Due to the lack of Persian QA datasets, Abadani et al. (2021) obtained the Persian Question Answer Dataset (ParSQuAD), which was created by translating the SQuAD 2.0. They created the translation in two versions, manual and automatic datasets. BERT, ALBERT and multilingual BERT models analyzed both datasets with and reached 56.66% for manual and 70.84% F1-score for automatic.

In addition, LMs can be applied in many areas except the QA domain. Peinelt et al. (2020) proposed tBERT, a simple architecture that combines topics with BERT for semantic similarity estimation. They showed that tBERT provides improvements in multiple semantic similarity estimation datasets versus a fine-tuned BERT. Sun et al. (2019) has proposed a new solution for aspect-based sentiment analysis using a sentence pair classification task. They fine-tuned the pre-trained BERT model. As a result of their analysis, SentiHood and SemEval-2014 showed that their suggestions were successful in Task 4 datasets. Qu et al. (2019) proposed a conceptually simple but highly effective approach called history answer embedding. They developed a rule-based method for date selection. They have provided seamless integration of the conversation history into a conversation QA model based on BERT. Adhikari et al. (2019) achieved the most advanced results for document classification by fine-tuning BERT. They also showed that BERT can be decomposed into a simple neural model, providing competitive accuracy at a lower computational cost than BERT. Al-Garadi et al. (2021) developed a BERT-based model to improve classification performance in prescription drug abuse classification. They compared the success of the proposed model with BERT-like models using a publicly available Twitter prescription drug abuse dataset and performed empirical analysis of BERT-based models. In addition to these literatures using pre-trained LMs, there are areas such as sentiment analysis (Singh et al., 2021; Li et al., 2021), text summarization (Ma et al., 2022) and image processing (He et al., 2020).

4 MATERIALS

Many materials and methods are used, including dataset, many NLP libraries, LMs in this study. In this section, materials and methods are explained separately under sub-headings.

4.1 SQuAD

SQuAD, created by Stanford University, is the reading comprehension dataset used for QA. It includes a question-answer pair for each article by making use of the articles on Wikipedia. There are two versions of the dataset. For SQuAD 1.1 version, each question has an answer and the answers are not multiple choice. There are 23215 paragraphs and 107785 question-answer pairs in a total of 536 articles. The answers were obtained by the crowdworkers' own answers (Rajpurkar et al., 2016). SQuAD 2.0 version, on the other hand, uses the same article and question-answer pairs as SQuAD 1.1. In addition, 53775 unanswered questions for the same articles have been added to this version. Unanswered questions were obtained by textual operations such as using opposite words, making words positive or negative. The purpose of SQuAD 2.0, in addition to answering questions, is not to answer questions that have no answers (Rajpurkar et al., 2018). Figure 4.1 gives an example of answering questions related to a paragraph in an article for the SQuAD 2.0 version. The example includes questions answered correctly (green bar), answered incorrectly (red bar), and questions with no answer ($< NoAnswer >$). The accuracy of the answer is determined by comparing the actual answer (ground truth answers) and the predicted (prediction) answer in each question.

Both SQuAD versions were randomly split into training (80%), development (10%) and test (10%) clusters. The test set is hidden for SQuAD because it is used in the benchmark platform to measure the success of the models. Statistics for both SQuAD versions are given in Table 4.1. Statistics for both SQuAD versions are given in Table 4.1. In this table, the questions with no

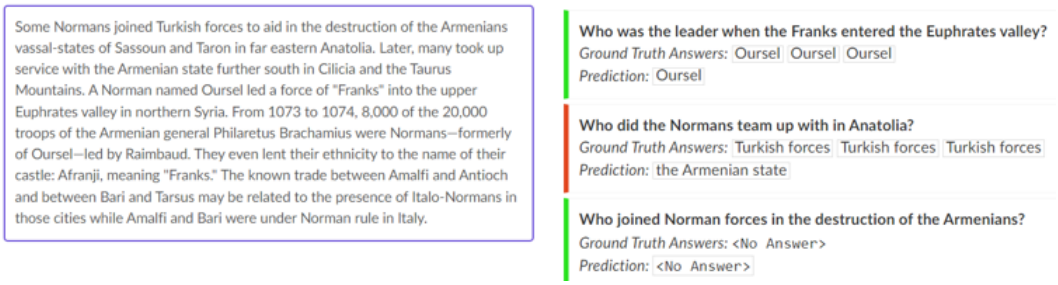


Figure 4.1: The example of a Wikipedia article for SQuAD 2.0 (Rajpurkar, 2022)

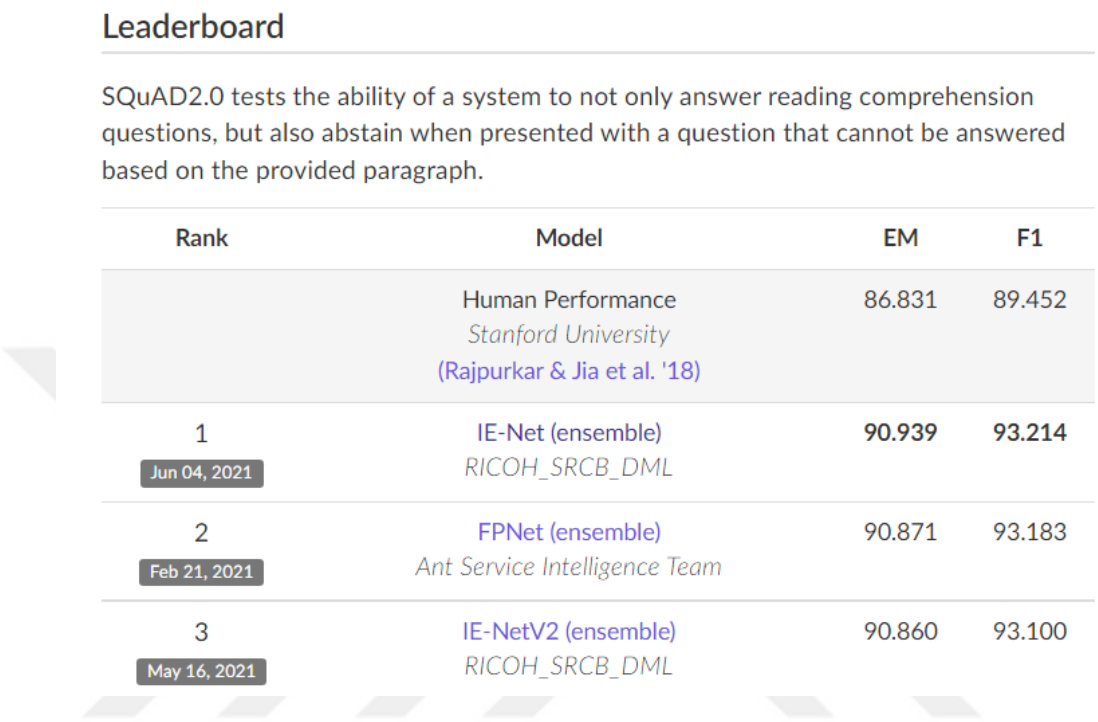
answers (negative examples) and the added number of articles are also given. In this study, analysis was performed on SQuAD 2.0 and, as mentioned, the development set was used during the evaluation phase, since the test set is confidential.

Table 4.1: Statistics of both SQuAD versions

		SQuAD 1.1	**SQuAD 2.0**
Train	Total examples	87599	130319
	Negative examples	0	43498
	Total articles	442	442
	Articles with negatives	0	285
Development	Total examples	10570	11873
	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

A benchmark platform was created to compare the results according to the analysis of SQuAD. Both LMs, unidirectional and bidirectional, are used on this platform. The success of the analyzed models is evaluated according to the

EM and F1-score F1 value. The success of the models is listed separately for both SQuAD 1.1 and SQuAD 2.0. Figure 4.2 shows the comparison of models on the platform for SQuAD 2.0. The results of the models are also compared by considering the success (Human Performance) of people's answers to SQuAD.



Leaderboard

SQuAD2.0 tests the ability of a system to not only answer reading comprehension questions, but also abstain when presented with a question that cannot be answered based on the provided paragraph.

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Jun 04, 2021	IE-Net (ensemble) *RICOH_SRCB_DML*	**90.939**	**93.214**
2 Feb 21, 2021	FPNet (ensemble) *Ant Service Intelligence Team*	90.871	93.183
3 May 16, 2021	IE-NetV2 (ensemble) *RICOH_SRCB_DML*	90.860	93.100

Figure 4.2: SQuAD benchmark platform (Link)

4.2 Language Models

The LM gives the probability distribution over the word sequences. LM provides context for distinguishing similar words and phrases (Petroni et al., 2020). Bidirectional LMs are used in this study. In bidirectional DM, the word's input order, position, and left-right context are important. Using this information, the models aim to assign probabilities to the sequence. ALBERT, RoBERTa, ELECTRA, SpanBERT models obtained based on BERT and BERT from bidirectional models are explained in detail under subsections.

4.2.1 BERT

It is a Transformer-based language representation model, defined as bidirectional encoder representation. It produces multiple and contextual representations of words. It uses the MLM method in the preprocessing stage. With this method, the model aims to randomly mask some tokens and predict this masked token based only on its context. The model consists of two stages: pre-training and fine-tuning. In the first step, the model is trained using unlabeled data for different pre-training tasks. In the other stage, all parameters are fine-tuned with data labeled according to the downstream task in the model initiated with pre-trained parameters. Figure 4.3 shows both stages of the BERT model. The model uses the same architecture in the pre-training and fine-tuning phases. Models are initialized for different tasks with the same pre-trained model parameters (MNLI, NER, SQuAD, etc.). The BERT model is built in two dimensions, BERT-Base (12 layers, 768 hidden dimensions and 12 attention head with 110M total parameter) and BERT-Large (24 layers, 16 attention head, 1024 hidden dimensions and 340M total parameter) (Devlin et al., 2018).

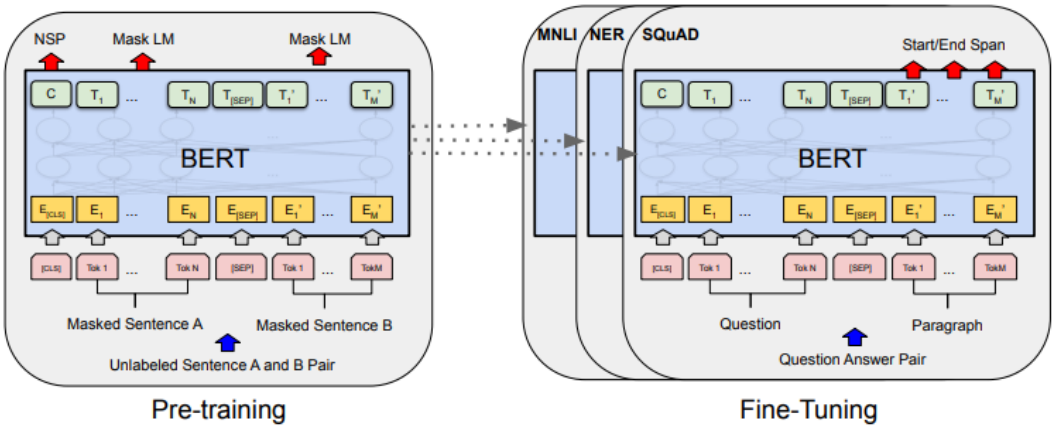


Figure 4.3: The stages of BERT model ([CLS]: a special token prefixed with each input, [SEP]: a special separator token) (Devlin et al., 2018)

BERT passes each token (words in the input text) through the token, segment, and position embedding layers. The layers are shown in Figure 4.4. First, each word in the token layer is converted to a vector representation. In the segment embedding layer, the input text pair is simply combined and fed into the model. There are only 2 vector representations in this layer. The first vector (index 0) is assigned to all tokens belonging to input 1, the last vector (index 1) is assigned to all tokens belonging to input 2. In the position embedding layer, BERT understands that an input text is given (Medium.com, 2021).

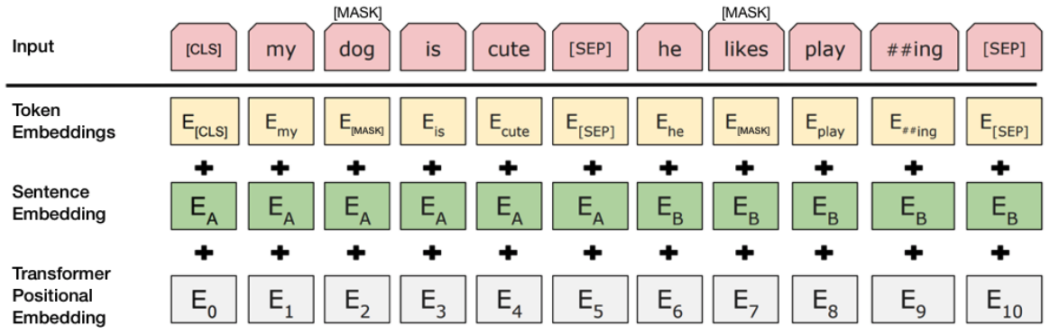


Figure 4.4: The embedding layers of BERT model (Devlin, 2019)

The BERT model can be used pre-trained for different textual tasks such as text classification, text summarization, text generation, machine translation, except for the QA domain.

4.2.2 RoBERTa

Training for the BERT model is costly, as it takes a long time to train high-dimensional data. In addition, the selection of appropriate hypermeters is also important in terms of the success of the model, time and cost. This is why the RoBERTa model, Robustly Optimized BERT, was developed. The model is based on applying changes in the pre-training phase of BERT. These changes include using longer arrays, applying the masking method dynamically rather than statically, and using larger groups over more data. Additionally, the next

sentence prediction task in the BERT model has been removed (Liu et al., 2019).

Figure 4.5 shows how the RoBERTa model works. When this figure is examined, the RoBERTa model takes an input span and context (span sentence). This model is sent to the classifier by combining the embedding of the [CLS] token, the average embedding of all tokens, and the span length(Chernyavskiy et al., 2021). With the proposed RoBERTa model, studies have been carried out on many tasks and datasets. The model, which can also be used as pre-trained, performed better than the BERT model.

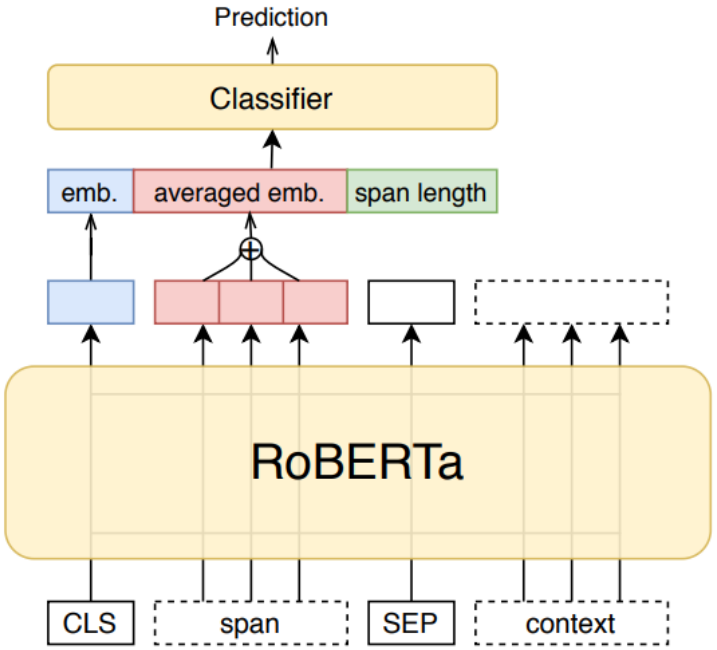


Figure 4.5: The operations of RoBERTa model (Chernyavskiy et al., 2021)

4.2.3 ALBERT

The problem of memory limit and tranmission overhead is quite common in LMs like BERT. Therefore, the ALBERT model, which has very few parameters compared to BERT, has been proposed. The ALBERT model uses two parameter reduction techniques. The first is factorized embedding parameterization, which can decouple the size of hidden layers from the size of

word embedding. In the second technique, the interlayer parameter is shared to prevent the parameter from growing with the depth of the network. In addition, a self-monitoring loss is used in the model to model inter-sentence consistency. As a result of the application of the model, the number of parameters is significantly reduced compared to BERT. Since a configuration similar to the BERT model is applied, it has been shown that this model has 18 times fewer parameters and can be trained approximately 1.7 times faster (Lan et al., 2019).

4.2.4 ELECTRA

While using MLM language models like BERT effectively, it requires a large amount of computation. To overcome this problem, the ELECTRA model does not use masking on the input. Instead, some tokens are sampled through the network and exchanged for alternatives. Rather than training a model that predicts the correctness of the deteriorated markers, the model aims to train a model that tries to distinguish whether the markers have been changed. So ELECTRA models distinguish between ”actual” and ”fake” input tokens generated by a neural network (Clark et al., 2020).

An example of the structure of ELECTRA is shown in Figure 4.6. Examining this figure, the word ”cooked” was masked and the word guessed via the small MLM was ”ate”. The ELECTRA model has determined that this word has been changed with the ”replaced” phrase. As a result of analyzing the ELECTRA

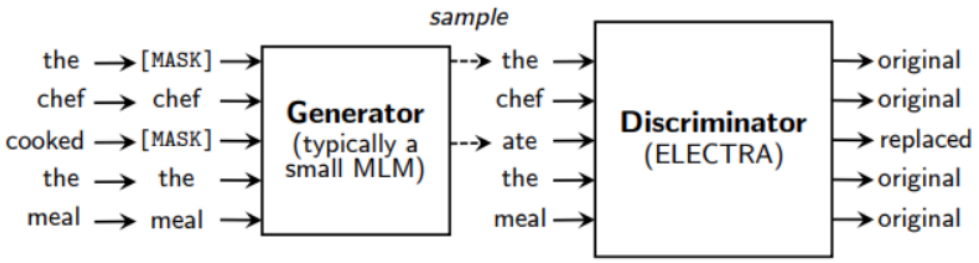


Figure 4.6: The structure of ELECTRA model (Clark et al., 2020)

model for certain tasks, it was computationally more efficient and showed

better performance than BERT.

4.2.5 SpanBERT

The SpanBERT model has been developed to better represent and predict text spans. Adjacent random spans (distance) are masked instead of random markers in this model. In addition, this model trains span boundary representations to predict the entire content of masked spans without relying on individual token representations. SpanBERT outperformed BERT when evaluated for different tasks such as QA and co-reference resolution (Joshi et al., 2020).

Example for SpanBERT training is shown in Figure 4.7. For the example, the range "an American football game" is masked. The span boundary objective (SBO) uses the output representations of the x4 and x9 (in blue) boundary markers to predict each marker in the masked span area (Joshi et al., 2020).

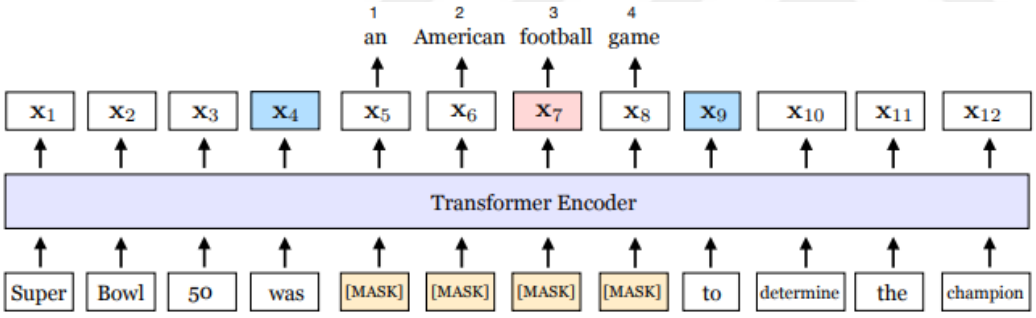


Figure 4.7: The example of SpanBERT training (Joshi et al., 2020)

4.3 Utilized Libraries

In this study, certain libraries were used for the operations and analysis applied in the texts and sentences. These libraries are explained in detail under sub-headings.

4.3.1 WordNet

WordNet[8], an electronic word database, is considered the most important resource available to researchers in computational linguistics, text analysis, and many related fields. Its design is inspired by current psycholinguistic and computational theories of human word memory. English nouns, verbs, adjectives, and adverbs are organized into synonymous sets, and each essentially expresses a lexicalized concept (Miller, 1998). An example for WordNet is shown in Figure 4.8. There are close meaning words, short definitions and example sentences of the word ”car”.

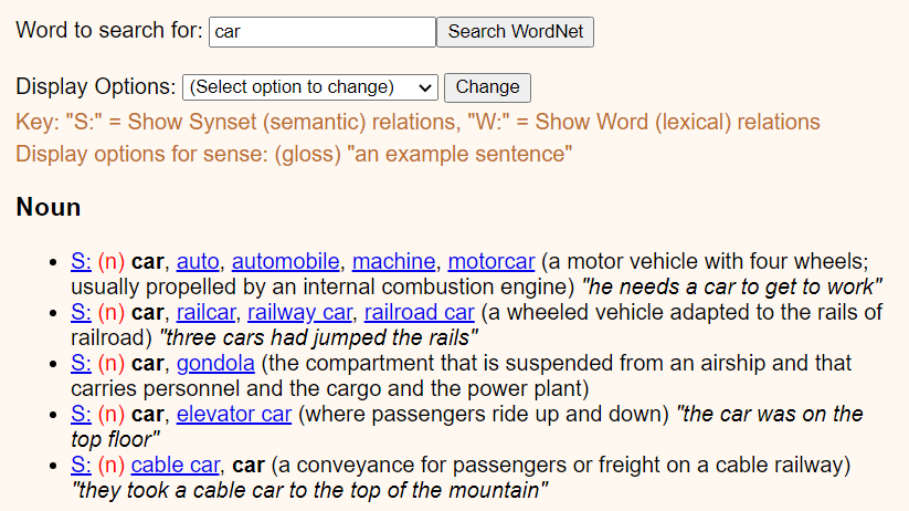


Figure 4.8: An example for WordNet (WordNet, 2021)

The main relationship between words in WordNet is synonyms, as between car and automobile. Synonyms are words that express the same concept and can be used interchangeably in many contexts, and are grouped into unordered synsets. Each of WordNet’s 117000 synsets is connected to other synsets through a small number of ”conceptual relationships”. Additionally, a synset contains a short description and in most cases one or more short sentences that illustrate the use of synset members (Fellbaum, 2005). In addition, terms with wider semantic scope in WordNet are called hypernym, and more specific

[8]http://wordnetweb.princeton.edu/perl/webwn?

ones are called hyponym.

4.3.2 StanfordCoreNLP

CoreNLP is a Java-based NLP library developed by Stanford University. CoreNLP library supports many NLP tasks such as POS tagging, NER, relation extraction, information extraction (IE), sentiment analysis (Manning et al., 2015). The StanfordOpenIE library, which is part of CoreNLP[9], is used for open IE. Open IE typically provides for the extraction of relation groups from plain text. It is an advantage that the schema is not specified in advance in this inference method, it specifies two variables in the text with a relation (subject: Mark Zuckerberg; relation: established; object: Facebook) (Angeli et al., 2015). The StanfordOpenIE-python version of StanfordOpenIE created for python was used in this study.

4.3.3 spaCy

It is an open source library developed in Python, used for NLP tasks such as lemmatization, entity linking, POS tagging, NER. It allows to create applications that process and understand very large texts. It can be used for IE or creating NLP systems, or for text preprocessing (Spacy, 2021). They have their own models for performing NLP tasks. The spaCy library was used for the NER task in this study. The en_core_web_sm[10] model is used to perform this task.

4.3.4 NeuralCoref

NeuralCoref[11] is pipeline extension that describes and solves co-reference sets using a neural network. NeuralCoref consists of two sub-modules. It uses a feed-forward neural network to identify a potential co-reference set using the SpaCy

[9]https://github.com/stanfordnlp/CoreNLP

[10]https://github.com/explosion/spacy-models/releases/download/en_core_web_sm-2.1.0/en_core_web_sm2.1.0.tar.gz

[11]https://github.com/huggingface/neuralcoref

library's tagger, parser, and NER annotations and calculates a co-reference score for each (HuggingFace, 2021a). In this study, the NeuralCoref library was used for the co-reference resolution (CoRes), which found the real nouns of the pronouns (Detailed information about this study is given in Section 5.2.1). Analysis was performed by updating SQuAD with CoRes applied.

4.3.5 AllenNLP

AllenNLP[12] is a platform and library built on PyTorch[13] which is used for many NLP tasks such as reading comprehension, sentiment analysis, NER, CoRes. This library has been developed on DL methods. There is also a platform with tutorials, API documentation, pre-trained LMs, and source code (Gardner et al., 2019). In this study, SQuAD was updated by using the AllenNLP library for CoRes (Detailed information about this study is given in Section 5.2.1) process and analysis was performed with updated SQuAD.

4.3.6 Tokenizer

The process of splitting a sentence into tokens is necessary to perform the analysis of words. In addition, the paragraphs in SQuAD should be divided into sentences in order to be used in sentence selection. Tokenize [14] module of NLTK[15] was used for these two tasks. In this module, word_tokenize is used for token parsing, and sent_tokenize and PunktSentenceTokenizer[16](Punkt) methods are used for splitting paragraphs into sentences.

4.3.7 WordNetLemmatizer

The lemmatization process reduces the word to root by applying morphological analysis to the word. With this operation, it transforms words that have the same meaning (separated, separation, separately) into the same form

[12]https://github.com/allenai/allennlp
[13]https://pytorch.org/
[14]https://www.nltk.org/api/nltk.tokenize.html
[15]https://www.nltk.org
[16]https://www.nltk.org/api/nltk.tokenize.punkt.html

(separate). Thus, these words affect the analysis positively without evaluating them separately. In this study, NLTK's WordNetLemmatizer[17] method was used for lemmatization. SQuAD was updated after all words were rooted and the effect of lemmatization was examined.

[17]https://www.nltk.org/api/nltk.stem.wordnet.html

5 METHODOLOGIES

Under this section, two methodologies based on NLP techniques and triples are mentioned. In the NLP-QAS extension, RNP methods are developed by using NLP methods. Analysis is performed on these methods for questions that LM cannot answer. In the TRP-QAS, triples extracted from sentences or paragraphs are used. For questions that the LMs cannot answer, analysis between these triple is applied. The detailed explanations of these two QAS extensions are explained under the sub-headings.

5.1 NLP-based QAS

Firstly, question terms are analyzed in the sentences of each paragraph to answer detection. Before analysis, paragraphs are parsed into sentences. Then, certain preprocessing techniques, some optional, are applied to the question terms and sentences. These preprocessing techniques are shown in Figure 5.1. The use of unnecessary data is prevented by these preprocesses. These preprocesses carried out are:

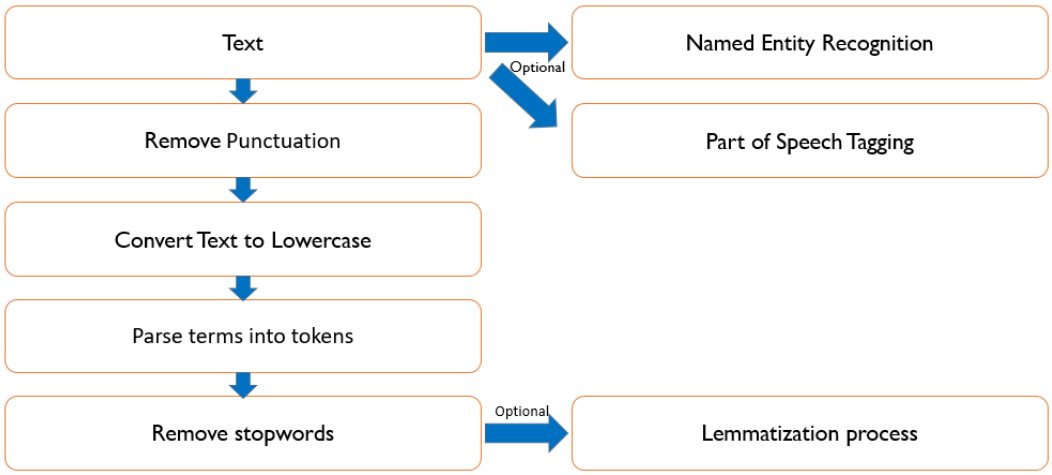


Figure 5.1: NLP preprocesing techniques

- Punctuation marks are removed.

- Texts are converted to lowercase. Thus, the complexity of writing the same words differently is eliminated.
- The texts are splitted into tokens. Duplicate tokens are removed to create unique tokens. Thus, the data volume is reduced.
- Tokens are filtered through stopwords. Finally, these redundant tokens are removed.
- There are proper names such as place, person, time in SQuAD articles. These proper names are detected by NER and POS tagging technique. After the proper names are determined, the terms used separately are combined and used as a single term ([Didier], [Drogba] - [Didier Drogba]).
- Lemmatization technique is applied on words. The same words, which can be in different forms, are obtained in a single form by lemmatization. Lemmatization can also be analyzed together with POS tagging. This process is applied as optional.

After preprocessing, an NLP-QAS extension is proposed for answer detection. In NLP-QAS, firstly, the most related sentence selection (RSS) process is carried out to search for the answer. Afterwards, the success of RNP methods for answer detection is analyzed in this sentence. An example of NLP-QAS operation is shown in Figure 5.2. Question, paragraph and answer are taken as input and processes between question and answer are analyzed by RNP methods. First of all, the sentences of the question and the related paragraph are preprocessed and paragraph is parsed into sentences. After the sentence is parsed into tokens, the question terms are compared with the terms of each sentence. The question term percentage (QTP) value is obtained for the ratio of the question terms in this sentence. The QTP of each sentence is sorted in descending order. For answer detection, the sentence containing the answer with the highest QTP value is selected. Then, an answer is searched on the selected sentence with RNP methods. As a result, the answer found by RNP methods is determined as the candidate answer.

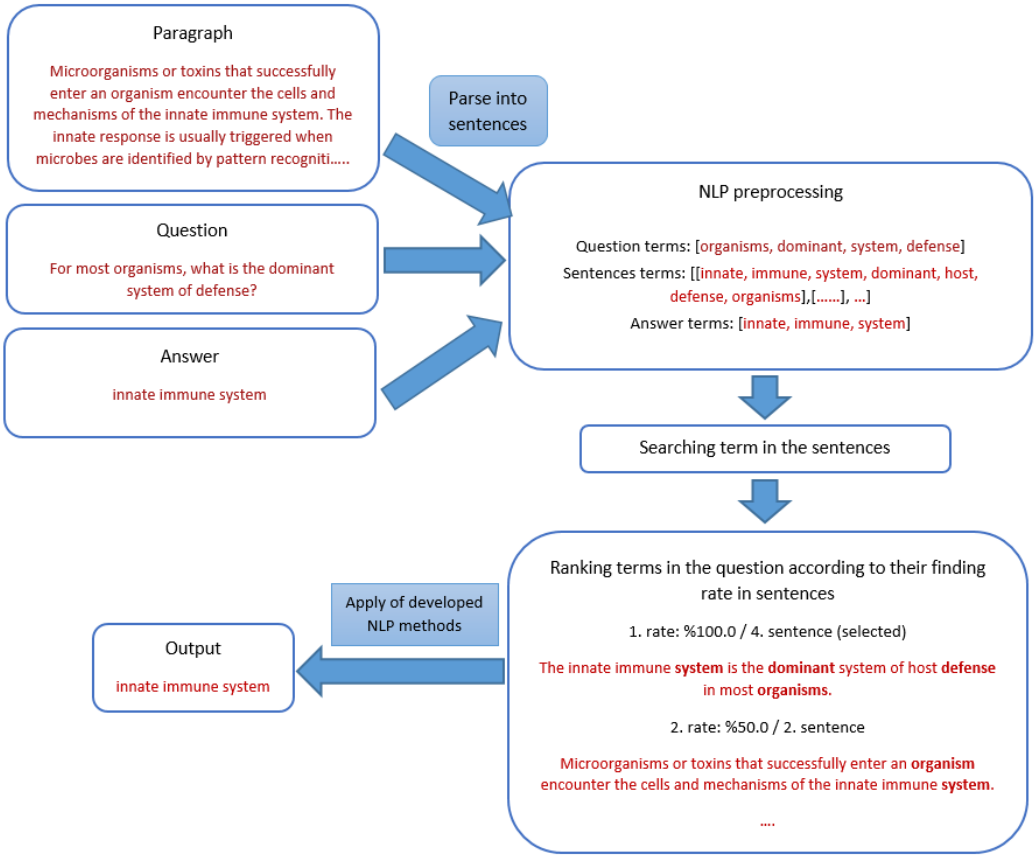


Figure 5.2: Analysis structure of the NLP-QAS for SQuAD

Pseudocode for RSS of NLP-QAS is shown in Figure 5.3. As stated in the pseudocode, only questions with answers are processed. As shown in Figure 5.3, for paragraphs in all articles, firstly the question is parsed into terms according to the preprocess selection (lines 11-16). Then, the question terms are searched within the sentences in the paragraph and the sentence containing the answer is selected among the sentences ordered according to QTP (lines 18-23). However, if the sentence cannot be found according to QTP, the sentence containing the answer in the paragraph is selected for answer detection (lines 24-28).

NLP-based RNP methods are proposed for more successful answer detection in NLP-QAS. RNP methods have been developed with NLP techniques such as NER, POS, string processes. RNP methods were used to answer detection

```
1  initialization SQuAD dataset;
2  reading titles, paragraphs, questions, answers lists;
3  for i in range(len(titles)):
4      title ← titles[i];                                          // Article's title
5      questionList ← questions[i];                                // Title's question list
6      for j in range(len(questionList)):                          // Reading the paragraph's questions
7          for k in range(len(questionList[j])):
8              paragraph ← paragraphs[j][k];                       // Reading paragraph
9              question ← questionList[j][k];                      // Reading question
10             answer ← answers[j][k];                             // Reading answer
11             if preprocess = TRUE:                               // Preprocess is selected
12                 question ← PreProcess(question);                // Preprocess function
13                 questionTerm ← create question term list according to question;
14             else:
15                 question ← perform just punctuation to question;
16                 questionTerm ← create question term list according to question;
17             if len(answer) ≠ 0:
18                 sentences ← SearchTerms(paragraph, question);   /* searching question
                    terms in sentences,sorting sentences according to QTP */
19                 if len(sentences) ≠ 0:
20                     for l in range(len(sentences)):
21                         if sentences[l] has answer:
22                             AnswerDetection(sentences[l], answer, question);  /* Sending values
                                to the function in order to answer detection with
                                developed methods */
23                             break;
24                 else:                                           // Examine sentences without QTP
25                     sentences ← paragraph's all sentences;
26                     for m in range(len(sentences)):
27                         if sentences[m] has answer:
28                             AnswerDetection(sentences[m], answer, question);
```

Figure 5.3: Pseudocode of RSS

in the sentence selected as a result of the QTP. Three RNP methods have been proposed and these three methods are mentioned under sub-headings.

5.1.1 Remove and Compare

When applying the remove and compare (RC) method, question terms are removed from the selected sentence (lines 7-9). Then, stopwords are removed from both the selected sentence and the actual answer (lines 10-14). If the term count in the selected sentence is equal to the term count in the answer, the remaining terms in the selected sentence are combined and the sentence becomes the candidate answer (lines 15-17). This candidate sentence is compared to the actual answer (lines 18-19). If the sentence and

the actual answer are equal, the candidate answer is determined as correct. The pseudocode of this method is shown in Figure 5.4.

```
1  runningMethod ← running method's number;
2  question,actualAnswer,sentence ← related terms for methods;
3  detectedAnswer ← 0;                                   // Detected answer count
4  if runningMethod = 1:                                                  // RC Method
5      questionTerms,answerTerms,sentenceTerms ← Performed preprocess;
6      stopwords ← load NLTK stopwords list;
7      for i in length(questionTerms):          // Removing question terms from sentence
8          if sentenceTerms has questionTerms[i]:
9              Removing questionTerms[i] in the sentenceTerms
10     for i in length(stopwords):                  // Removing stopwords from term lists
11         if sentenceTerms has stopwords[i]:
12             Removing stopwords[i] in the sentenceTerms
13         if answerTerms has stopwords[i]:
14             Removing stopwords[i] term in the answerTerms
15     if length(sentenceTerms) = length(answerTerms):     // Lists' term count is equal
16         sentence ← create sentence with sentenceTerms list;
17         answer ← create answer with answerTerms list;
18         if sentence = answer:                          // sentence and answer are equal
19             detectedAnswer ← detectedAnswer + 1 ;                // Answer is detected
```

Figure 5.4: Pseudocode of RC method

5.1.2 Searching with NER

In the SNER method, answer detection is applied by using the NER method. NER identifies entity names in texts with different labels. That is, the NER method extracts entity types such as person, place, and time in a sentence (Specifications, 2020). Some entity labels and label's descriptions from NER are mentioned in Table 5.1.

In the application phase of the SNER method, the question pronoun is first sought in the question sentence. The most appropriate NER entity label (PERSON, DATE, etc.) is selected according to Table 5.1 (lines 2-6) for answer detection by question pronoun. Then, it is checked whether this entity label exists in the NER statements of the sentence selected according to QTP. If this sentence includes this entity label, the term for this entity label is chosen as the candidate answer (lines 8-12). Finally, the candidate and actual answers are compared, and if both are equal, the candidate answer is called correct (lines 13-14). The pseudocode of the SNER method is shown in Figure 5.5.

Table 5.1: Some NER labels and descriptions

NER Label	Description
PERSON	Person's name&surname, etc.
NORP	Nation, religion, etc.
FAC	Building, airport, bridge, etc.
ORG	Company, agency, institute, etc.
GPE	Country, city, etc.
LOC	Mountain range, locations, etc.
EVENT	Storm, war, sport events
WORK_OF_ART	Book, song, etc.
LANGUAGE	National languages
DATE	Associated date and period names
TIME	Time expressions such as clock
PERCENT	Ratio, percentage expressions
QUANTITY	Quantity expressions

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

Figure 5.5: Pseudocode of SNER method

5.1.3 Searching with POS tagging

In SPOS method, POS tagging method is used for answer detection. Special tags for each word are determined in POS tagging. With these tags, the entire

structure of the sentence is extracted (Rachiele, 2018). POS tags and tags' descriptions are shown in Table 5.2.

Table 5.2: Some POS tags and descriptions

Tag	**Description**
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
JJ	Adjective
IN	Preposition or subordinating conjunction
NN	Noun, singular or mass
NNP	Proper noun, singular
RB	Adverb
WDT	Wh-determiner
WP	Wh-pronoun
WRB	Wh-adverb

In the first step of the SPOS method, the answer tag is determined by extracting the POS tag of the question pronoun as in Table 5.2 (lines 2-10). After removing the question terms from the sentence selected according to QTP, the remaining terms in this sentence is parsed into POS tags (lines 13-14). If there is a POS tag with an answer tag in this sentence, the term of this tag is selected as the candidate answer. However, it is checked whether the next term has the same tag. If there is a term with the same tag, the term is added to the candidate answer (lines 16-26). Candidate and actual answers are then compared. If both are equal, the candidate answer is correct (lines 27-28). The pseudocode of the SPOS method is shown in Figure 5.6.

An example for RNP methods is shown in Table 5.3. This table shows that the answers were determined by these methods. The example shows that since the RC method is independent of question pronouns, a direct answer is

```
1  if runningMethod = 3:                                              // SPOS Method
2      posQuestion ← Split into POS tag(word,tag) of question ;
3      if posTag.tag has WRB:                          // Pos tag indicates wh- adverb
4          if posQuestion.word = when:
5              answerTag ← CD
6          if posQuestion'.word = where:
7              answerTag ← NN
8      if posQuestion.tag has WP$:          // Pos tag indicates possessive wh-pronoun
9          if posQuestion'.word = whose:
10             answerTag ← NNP
       /* There are many answerTag values(IN,NN, etc.)  according to pos
       tag:NNP,WP,etc.  So If condition count are more                  */;
11     if answerTag ≠ ∅:
12         answer ← ∅;
13         sentence ← Removing question terms from sentence;
14         posSentence ← Split into POS tag(word,tag) of sentence;
15         index ← 0 ;
16         for i in length(posSentence):
17             if answerTag = posSentence[i].tag:
18                 answer ← posSentence[i].word + ' '
19             if index + 1 ¡ length(posSentence) − 1:       // create answer as incremental
20                 for j ← index+1 in length(posSentence):
21                     if answerTag = posSentence[j].tag:
22                         answer ← posSentence[j].word + ' '
23                     else:
24                         break;
25                 break;
26             index ← index + 1
27         if actualAnswer = answer:                  // Answer and actual answer is equal
28             detectedAnswer ← detectedAnswer + 1;                  // Answer is detected
```

Figure 5.6: Pseudocode of SPOS method

obtained when question terms and stopwords are removed from the related sentence. For SNER and SPOS, the question pronouns ”what country” and ”when”, respectively, indicate that the answer should have the ”GPE” NER label and the ”CD” POS tag, respectively. In this example, the correct answer is reached with these answer tags. In addition, the expanded version of Table 5.3 is given for each method in Table 8.1, Table 8.2 and Table 8.3 respectively RC, SNER, SPOS in the Appendix section 8.

Table 5.3: 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') Answer: France (tag: GPE)
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'), ...] Answer: 911 (tag: CD)

5.2 Triple-based QAS

In the TRP-QAS extension, all LMs are first analyzed on SQuAD. All question-answer pairs that the models answered correctly or incorrectly are saved for use. Before the analysis, the real nouns of the pronouns in the entire dataset are found with CoRes, and SQuAD is updated with this operation. With the TRP-QAS extension, the analysis is performed on the question-answer pairs that the LMs answered incorrectly by using the sentence or paragraph of the question. It is aimed to determine the answer by extracting subject-predicate-object triples on the relevant sentence or paragraph. The design of the TRP-QAS is given in Figure 5.7. The previous information briefly mentioned in this paragraph is explained in the sub-headings.

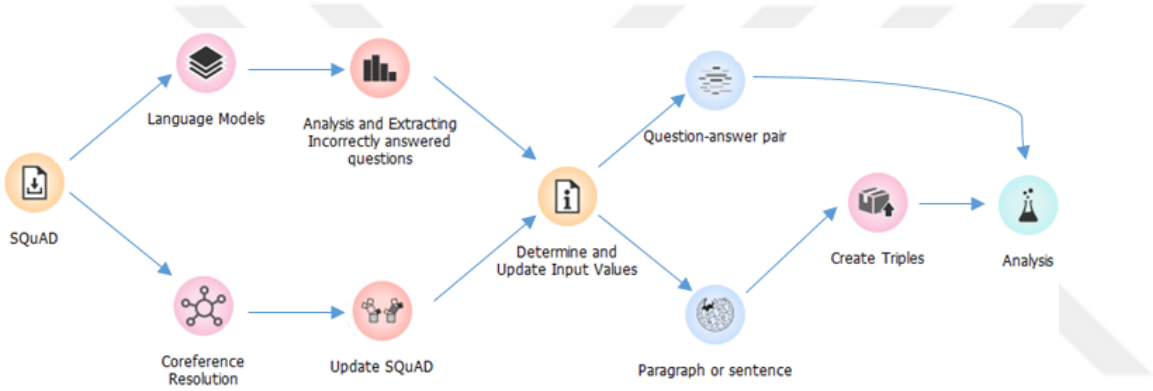


Figure 5.7: The design of TRP-QAS

5.2.1 Co-reference Resolution

CoRes is the task of identifying linguistic expressions that refer to the same real-world entity in natural language (Zheng et al., 2011). CoRes aims to resolve duplicate references for an object in a document. When CoRes is applied to NLP domains such as machine translation, sentiment analysis, QA, and summarization, it has the potential to improve the accuracy value greatly. Coreference terms may have completely different grammatical structures and functions, yet may refer to the same linguistic entity. Here, the entity can be a single object or an object group that together form a new single entity

(Sukthanker et al., 2020). Detecting the real noun form of the pronoun is a CoRes process. In this study, Huggingface's NeuralCoref and AllenNLP libraries were utilized to implement CoRes. CoRes is applied to the entire SQuAD. An example sentence of SQuAD for the CoRes process is shown in Figure 5.8. The word "their" is examined in the sentence. Candidate terms that can express the word "their" are marked through the NeuralCoref library, and each term's probability is calculated. The candidate term with the highest probability is selected instead of a pronoun. The final version of the sentence is "The Normans were the people who in the 10th and 11th centuries gave **the Normans (their)** name to Normandy, a region in France." has been.

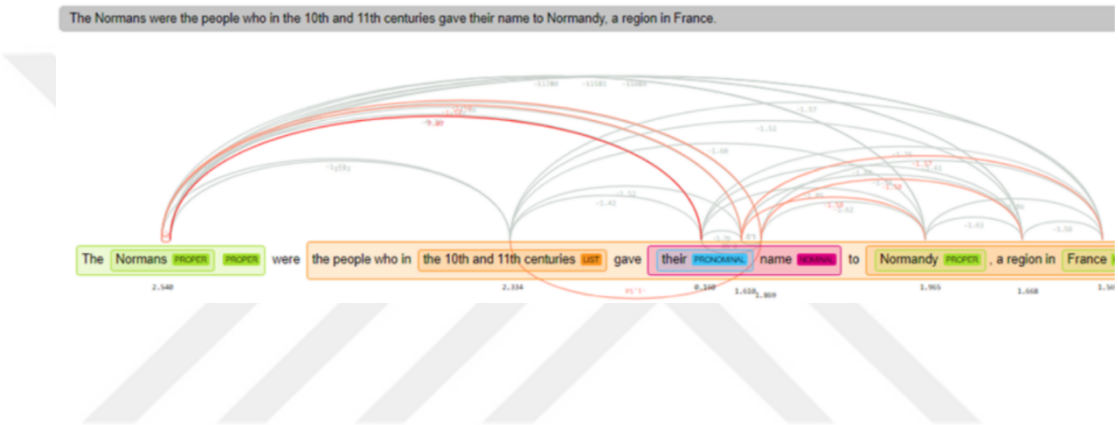


Figure 5.8: The CoRes example for SQuAD (HuggingFace, 2021b)

5.2.2 Create Triples

Ontology, which is a knowledge-based approaches, is a structure made up of RDF triples. Ontology indicates the concepts and properties of each concept describing various features and attributes of the concept, restrictions of concepts, and the relationship between the concepts (Noy and McGuinness, 2001). RDF and OWL are used to create an ontology. RDF is a machine-readable metadata model and language. RDF consists of three structures: subject, predicate, and object. Both subject and object can be anonymous objects. The predicate describes the relationship between both objects. The RDF specification can be viewed as a graph represented as a node (subject)-end (predicate)-node (object) structure (Gutierrez et al., 2007). Ontology can

be obtained through these RDF triples. Exemplary RDF triples are given in Figure 5.9. In this example, it is mentioned that the country of Berlin is Germany and its total area is 891.7km. Berlin (subject), 891.7 (object), and Germany (object) indicate objects, AreaTotalKm and Country indicate predicates.

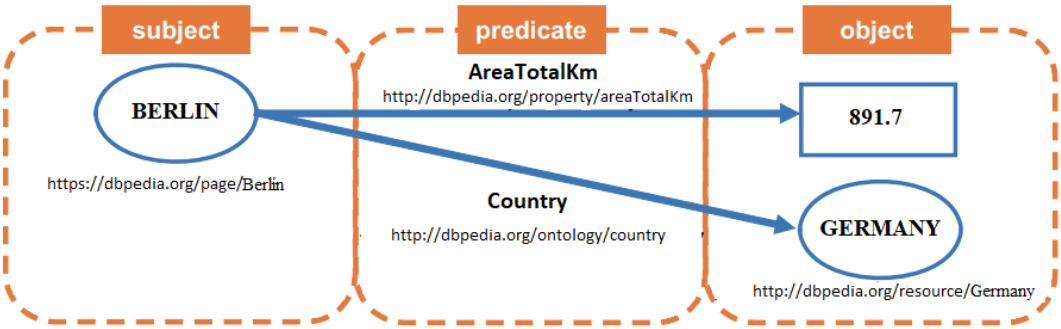


Figure 5.9: The example for RDF triples

Inspired by the RDF triples in ontology, triple-based analysis was proposed on SQuAD. It is aimed to extract triples on the related texts according to the question. The StanfordOpenIE library, which can extract subject-relation-object triples on any sentence, was used for this operation. Triples were obtained through this library on the sentence or paragraph of the questions that LMs couldn't answer correctly on SQuAD. Triple-based analysis was performed on the questions answered incorrectly. An example of the triples formed is given in Table 5.4. Table 5.4 shows that any sentence can contain more than one triple.

5.2.3 Implementation of All Processes

Initially, SQuAD's analysis is performed on all LMs, and questions answered incorrectly by the models are saved. RSS method is used to search for the answer in the most related sentence, rather than the paragraph before triples are created. As mentioned in Section 5.1, this RSS method uses questions, paragraphs, and answers as input. The pre-processed paragraph (stopword

Table 5.4: Triple examples according to some sentences

Sentences	Triples
As part of the agreements, both BSkyB and Virgin Media agreed to terminate all High Court proceedings against each other relating to the carriage of their respective basic channels.	'subject': 'Virgin Media' 'relation': 'terminate' 'object': 'High Court proceedings'
Other components are often present; pumps (such as an injector) to supply water to the boiler during operation, condensers to recirculate the water and recover the latent heat of vaporisation, and superheaters to raise the temperature of the steam above its saturated vapour point, and various mechanisms to increase the draft for fireboxes.	subject': 'mechanisms' 'relation': 'draft for' 'object': 'fireboxes'
	'subject': 'superheaters' 'relation': 'raise' 'object': 'temperature above saturated vapour point'
Ergänzungsschulen are secondary or post-secondary (non-tertiary) schools, which are run by private individuals, private organizations or rarely, religious groups and offer a type of education which is not available at public schools.	'subject': 'Ergänzungsschulen' 'relation': 'are' 'object': 'secondary'

deletion, punctuation, etc.) is parsed into sentences, and the question is parsed into tokens. The question terms are then searched within each sentence in the paragraph. According to the QTP value in the sentences, the related sentence is selected as candidate sentence. For each incorrectly answered question-answer pair, candidate sentences are obtained. The paragraphs are processed without any process for the questions that have no answer ($< NoAnswer >$). Then, a separate processes are performed for the questions that have answers and no answers in the creating triple-stage. Separate processes are performed for creating triple:

- For the questions that have answers (lines 12-26), question, sentence, and answer are taken as input. Since the updated SQuAD development set by applying CoRes will be used in the analysis phase, the candidate sentence previously found with RSS is obtained again according to the sentence order for the updated SQuAD. All triples of this sentence are extracted through the StanfordOpenIE library (lines 15-16). Then, the question terms are searched in all triples. If the question terms are found in two of any triples (for example subject-predicate), the remainder part (object) is accepted as the candidate answer (lines 18-23). If this candidate answer is equal to the actual answer, the TRP-QAS accepts the answer as correct (lines 24-26).
- For the questions that have no answer (lines 27-42), questions, paragraphs, and answers are taken as input. First, the triples of the paragraph

are obtained. Question terms are searched in all triples (lines 32-39). If the question terms aren't found in at least two of the triples, the TRP-QAS accepts that there is no answer (lines 35-36). However, if only part of these triples remained as candidate answer after the question term was searched, NER analysis is applied to this part. That is, the NER label (PERSON, DATE, LOCATION, etc.) is searched according to the question pronoun (who, when, where, etc.). If there is no answer for the NER label, the system again accepts that there is no answer (lines 37-42). Thus, the TRP-QAS answers correctly because it detects that there is no answer(lines 43-45).

In addition, the pseudocode for the operation of the TRP-QAS is shown in Figure 5.10. Step by step, all parts are specified in pseudocode.

Examples of correct answers by the TRP-QAS extension for both the question has an answer and no answer are given in Figure 5.11 and Figure 5.12. Figure 5.11 contains a question that has an answer. The triples (T1, T2) of the selected sentence of the question are extracted for this figure. Then, the question terms are removed from the T1 and T2 triples. The example shows that only the term 'Central Bridge' remains when the question terms are removed from the triples. This term is accepted as a candidate answer. Figure 5.12 gives an example of a question with no answer. Since there is no answer, the most appropriate sentence selection process cannot be applied. Because of this, all triples of the paragraph to which CoRes are applied are obtained. If a single term (T5, T6, T9, T10) remains among the triples after the search for the question terms, the suitability of the remaining term is analyzed according to the NER label (Who: PERSON). The example shows that even if there is a remainder term (orange terms), this term doesn't refer to PERSON according to NER. That is why 'No Answer' is accepted as the candidate answer.

```
initialization SQuAD dataset;
cores_dataset ← update SQuAD with coreference resolution process ;
informations ← all question-answer pairs and paragraph list in SQuAD ;
analyzing LMs with SQuAD and creating files from incorrectly answered question for each LM;
lm_files ← create list: information in analyzed files for each LM and selected sentence according to QTR ;
execute StanfordOpenIE library;
for file in lm_files:                                   // BERT, ELECTRA, ALBERT, RoBERTa LM file
    texts ← read file ;
    detect_answer ← 0 ;
    for text in texts:
        question_lm, answer_lm ← parsing text and obtaining question, answer information;
        if answer_lm ≠ 'No Answer':                                   // Question has an answer
            for paragraph, sentence, question, answer in informations:
                if question = question_lm:          // Finding a wrongly answered question
                    sentence_cores ← finding related sentence in cores_dataset through sentence;
                    triples ← creating triples of sentence_cores via StanfordOpenIE library;
                    answer_list ← candidate answer list for triple ;
                    for triple in triples:                          // Analyzing triples
                        subject, predicate, object ← parsing triple ;
                        searching for question terms in the subject, predicate, object ;
                        if question terms are found in two of any triples:
                            candidate_answer ← remainder triple ;
                            answer_list.append(candidate_answer);
                    if len(answer_list) == 1  &  candidate_answer == answer:    // True Answer
                        detect_answer += 1 ;
                        break;
        else:                                                       // Question has no answer
            for paragraph, sentence, question, answer in informations:
                if question = question_lm:
                    triples ← creating triples of paragraph;
                    no_answer ← False;
                    for triple in triples:
                        subject, predicate, object ← parsing triple ;
                        searching for question terms in the subject, predicate, object ;
                        if question terms aren't found in at least two of triples:
                            no_answer ← True;
                        else:
                            if NER tag determined by question pronoun is not in the remaining triple:
                                no_answer ← True;
                            else:
                                no_answer ← False;
                                break;
                    if no_answer == True:                           // Detecting no answer
                        detect_answer += 1 ;
                        break;
```

Figure 5.10: The pseudocode of TRP-QAS

Sentence: Legally, the Central Bridge is the boundary between High and Upper Rhine.

Question: What is the legal boundary behind the High and Upper Rind?

Answer: Central Bridge

Sentence triples

T1: 'subject': 'Central Bridge', 'relation': 'legally is boundary between', 'object': 'High'

T2: 'subject': 'Central Bridge', 'relation': 'is', 'object': 'boundary'

Figure 5.11: Examples of the triple-based system for has answer

Paragraph: Even before the Norman Conquest of England, the Normans had come into contact with Wales. Edward the Confessor had set up the aforementioned Ralph as earl of Hereford and charged him with defending the Marches and warring with the Welsh. In these original ventures, the Normans failed to make any headway into Wales.

Question: Who made Edward the Confessor Earl? NER answer type: Who: PERSON (Search Person)

Answer: No Answer

Paragraph triples

T1: 'subject': 'Normans', 'relation': 'had come into', 'object': 'contact with Wales'

T2: 'subject': 'Normans', 'relation': 'had come before', 'object': 'Norman Conquest of England'

T3: 'subject': 'Normans', 'relation': 'had come into', 'object': 'contact'

T4: 'subject': 'Confessor', 'relation': 'had set up', 'object': 'Ralph'

T5: 'subject': 'Confessor', 'relation': 'charged', 'object': 'Edward' – 'charged' is not PERSON

T6: 'subject': 'Edward', 'relation': 'had set up', 'object': 'Ralph' – 'had set up' is not PERSON

T7: 'subject': 'Confessor', 'relation': 'had set up', 'object': 'aforementioned Ralph'

T8: 'subject': 'Edward', 'relation': 'had set up', 'object': 'aforementioned Ralph'

T9: 'subject': 'Edward', 'relation': 'had set up Ralph as', 'object': 'earl of Hereford' – 'had set up Ralph as' is not PERSON

T10: 'subject': 'Confessor', 'relation': 'had set up Ralph as', 'object': 'earl of Hereford' – 'had set up Ralph as' is not PERSON

T11: 'subject': 'Normans', 'relation': 'failed In', 'object': 'original ventures'

Figure 5.12: Examples of the triple-based system for no answer

6 IMPLEMENTATION

SQuAD Explorer application was created with Django framework using Python programming language. Django is an open source high-level Python Web framework that supports rapid development and pragmatic design (Django, 2020).

Paragraphs of all articles, question-answer pairs of paragraphs are stored in JSON format on the SQuAD platform. Therefore, SQuAD was transferred to the MongoDB database via Python programming language and used in the application so that the data can be read easily and quickly. MongoDB is a document-based database with scalability and flexibility through user-implemented querying and indexing (MongoDB, 2020). MongoDB was preferred in this study because it can store JSON and similar documents in the database.

6.1 Services

In this study, functions were created for operations using many libraries via Python:

- Reading dataset: SQuAD imported to MongoDB database is pulled from MongoDB via function when SQuAD Explorer is run. All paragraphs, questions and answers are saved in the arrays.
- Listing data: The listing process for examining the data is performed with this function. Paragraph selection is made according to the article selected for SQuAD. Afterwards, the paragraph and the question-answer pairs belonging to the paragraph are listed.
- Searching data: It is a function that allows searching on the entire SQuAD read from the database according to the term entered from the keyboard. The output of the function indicates the article, the paragraph number and the sentence in which the found terms are included.

- Sorting sentence: It is the function that performs RSS operation. According to the selected question of the paragraph, question terms are searched on the sentences in the paragraph that are parsed into sentences. With the function output, the sentences are ordered according to the QTP value, and the related sentence, QTP value, and question terms are listed. The function includes optional pre-processing.
- Statistic extraction: It is the function where statistics of SQuAD are obtained. Analysis of question pronouns, numbers of questions with an answer and no answer for SQuAD, the term rate that can be used when stopwords are removed are extracted through this function.
- Creating graphic: Statistics for SQuAD are obtained with this function. Using the values required for statistics, the graph is plotted in the function via the Plotly[18] library.
- CoRes process: CoRes is the function that implements the operation. It obtains the noun forms of the pronouns of the paragraph that are input to the function.
- Applying RNP methods: RNP methods are applied to the selected sentence as a result of RSS according to the question with the function. In the related sentence, the operations of the RC, SNER and SPOS methods are performed respectively. If a candidate answer cannot be found with one method, the other method is analyzed. The function output returns candidate answer and the number of correct answers.
- Triple extraction: According to the related sentence and paragraph, triple extraction is performed via the StanfordOpenIE library. This function returns an array of triples. The candidate answer is determined according to the question terms among the triples in the arrays.
- Analysis of LM's with TRP-QAS: Questions answered incorrectly by LMs were saved before analysis. It is the function that is applied to

[18]https://plotly.com/python/

TRP-QAS for each LM in all saved questions. The candidate answer is searched among the triples obtained by using the related sentence or paragraph of the question. The output of the function returns the candidate answer and the number of correct answers.

6.2 User Interface

The web pages were created for the SQuAD Explorer application. This explorer includes pages such as data review, term search, RSS processing and demos for extensions. These pages are explained in detail under this section.

6.2.1 Main Page

There is detailed information about SQuAD on the home page of SQuAD Explorer. The image of the main page is shown in Figure 6.1. For versions 1.1 and 2.0 of SQuAD, redirection to the relevant page is applied. Access to the training and development set is also provided.

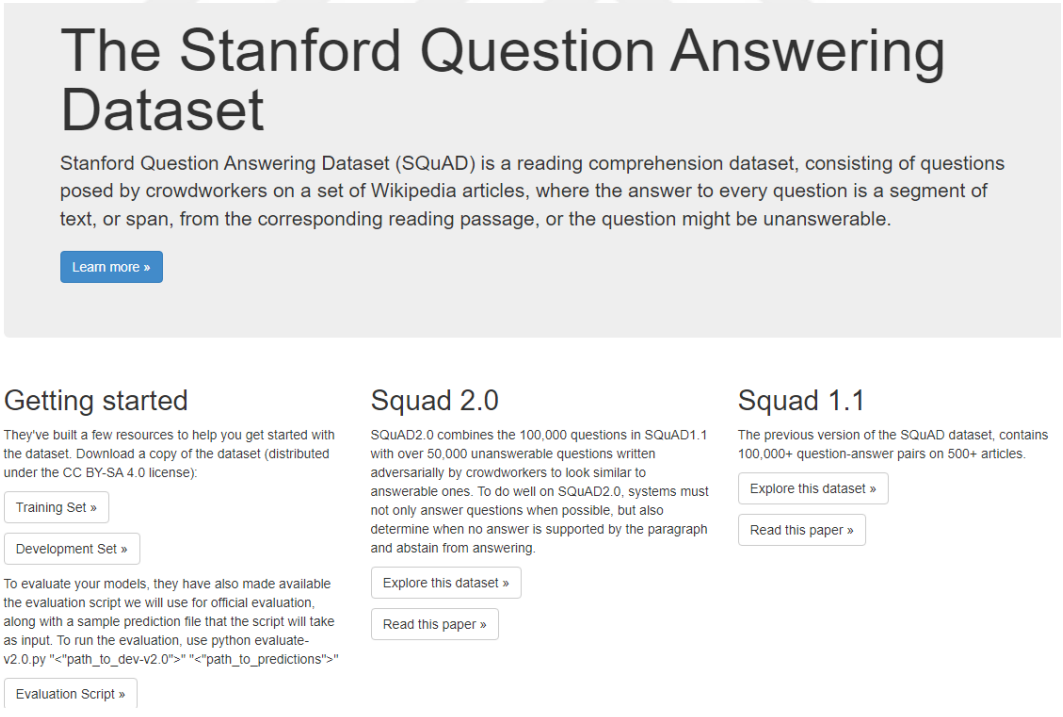


Figure 6.1: Main page in SQuAD Explorer

6.2.2 Data Review

Reading SQuAD from the MongoDB database can be done through the SQuAD Explorer platform. An example of examining data on the application is shown in Figure 6.2. By selecting the training and development set in the dataset, a selection can be made between the listed articles. Paragraph and question-answer pairs can also be listed by selecting any paragraph of the selected article.

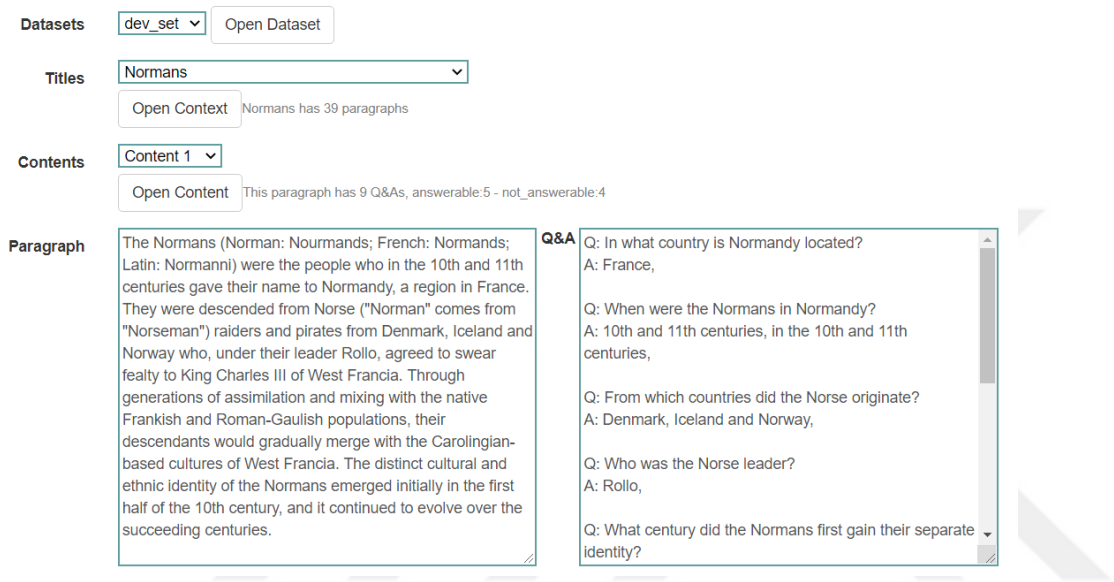


Figure 6.2: Listing data in SQuAD Explorer

6.2.3 Searching Data

The search process in the entire dataset of the term or terms entered from the keyboard is performed on this page. As an example, the search process is shown in Figure 6.3. As a result of the search, the article in which the term is included, the paragraph number and the sentence order are specified. It is possible to enter more than one term on the search page with ",".

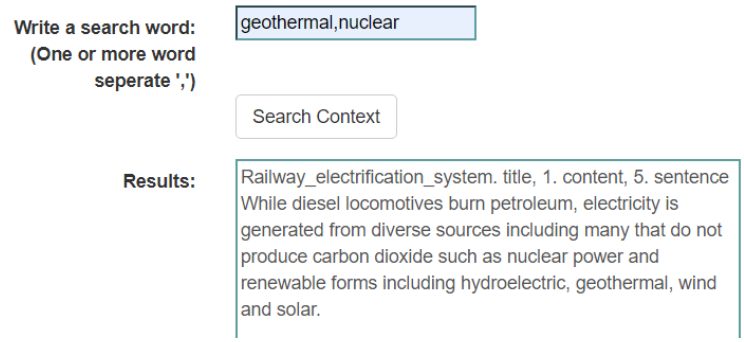


Figure 6.3: Searching term in SQuAD Explorer

6.2.4 Sentence Selection Analysis

RSS stage is applied in both NLP-QAS and Triple-QAS analyzes. As explained before (Section 5.1), the most related sentence with the QTP obtained according to the presence of question terms in the sentences is selected as a candidate for answer detection. For this process, analysis can be performed in SQuAD Explorer as in Figure 6.4. Candidate sentences of the question belonging to the selected article are sorted in percentile according to the QTP.

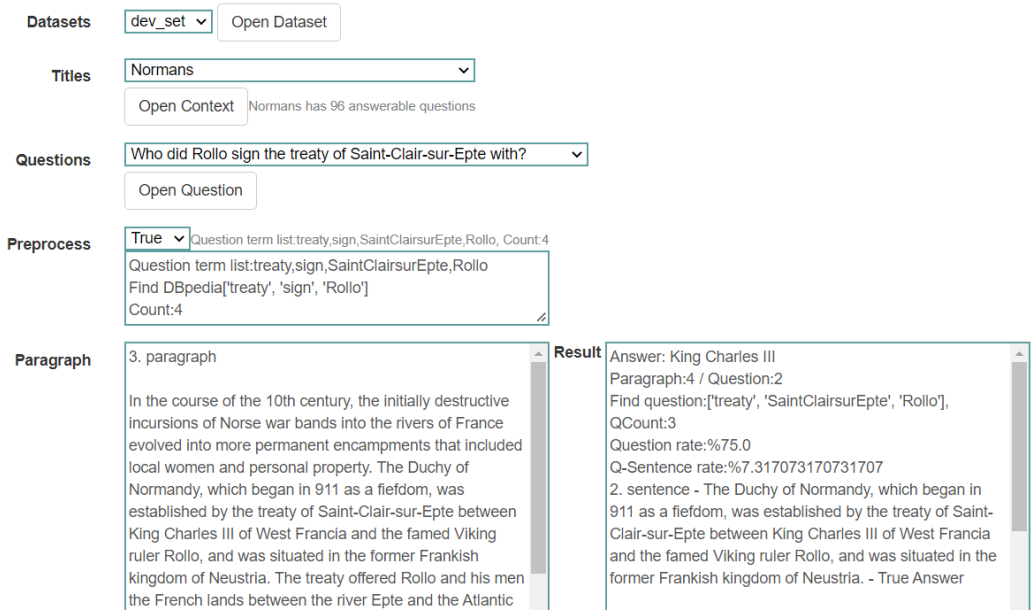


Figure 6.4: Sentence selection in SQuAD Explorer

6.2.5 SQuAD Graphics

It is the page created for drawing some graphics for the analyzes made on SQuAD. Python's plotly library was used. The order of the sentence selected according to RSS, the detection rates of the sentences, the detection rates of the correct answer with RNP, the success rate of the RNP methods separately are obtained on this graphic page. The graph of QTP rates in sentence selection for NLP-based QAS is shown in Figure 6.5 as an example.

Figure 6.5: Statistic page in SQuAD Explorer

6.2.6 Demo for RNP

It is the page created for the demo of the RNP methods proposed in NLP-QAS. An example demo where LMs answer incorrectly but RNP methods answer correctly is shown in Figure 6.6. In this figure, after the question is selected for the RC method, the paragraph of the question is listed. RSS method is performed for the paragraph and the related sentence is selected according to the QTP value and the actual answer. The operations applied

for RC are shown on the selected sentence. As a result of the procedures, the candidate answer (James Hutton) and the actual answer are equal, so the method determines the answer correctly. There are examples in SNER and SPOS in this demo.

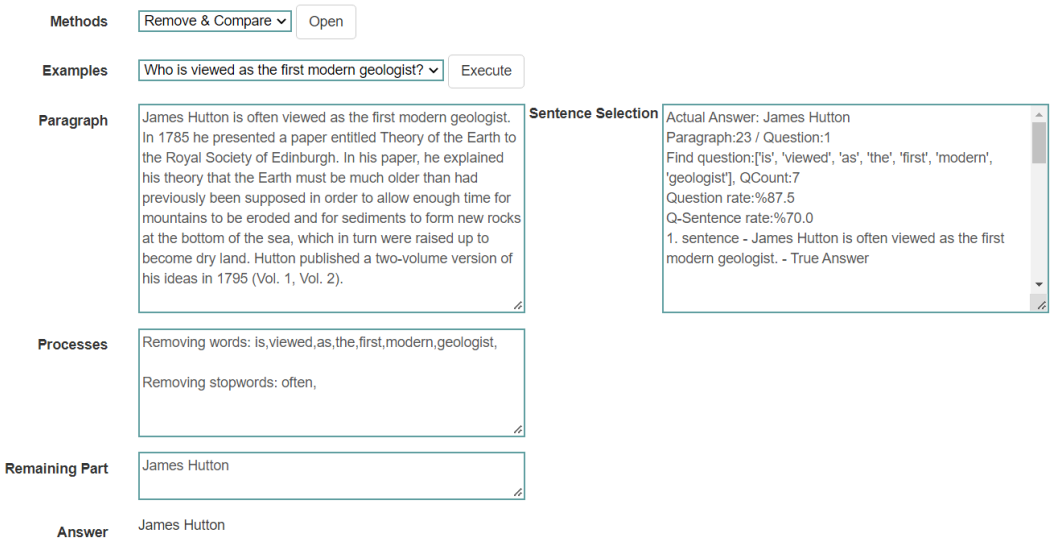


Figure 6.6: Demo page for RNP methods

6.2.7 Demo for Triple-based QA

This is the page created for the demo of the proposed method for TRP-QAS. Applying a triple-based method for a sample question that LMs answered incorrectly is shown in Figure 6.7. The paragraph of the question is extracted first. Then, the relevant sentence is selected by applying the RSS process on this paragraph and triple extraction is applied for the selected sentence. TRP-QAS is applied according to the question terms among the extracted triples. The remaining part in the found triple is determined as the candidate answer (Virgin Media). Since the candidate and the actual answer are equal, TRP-QAS answers the question correctly.

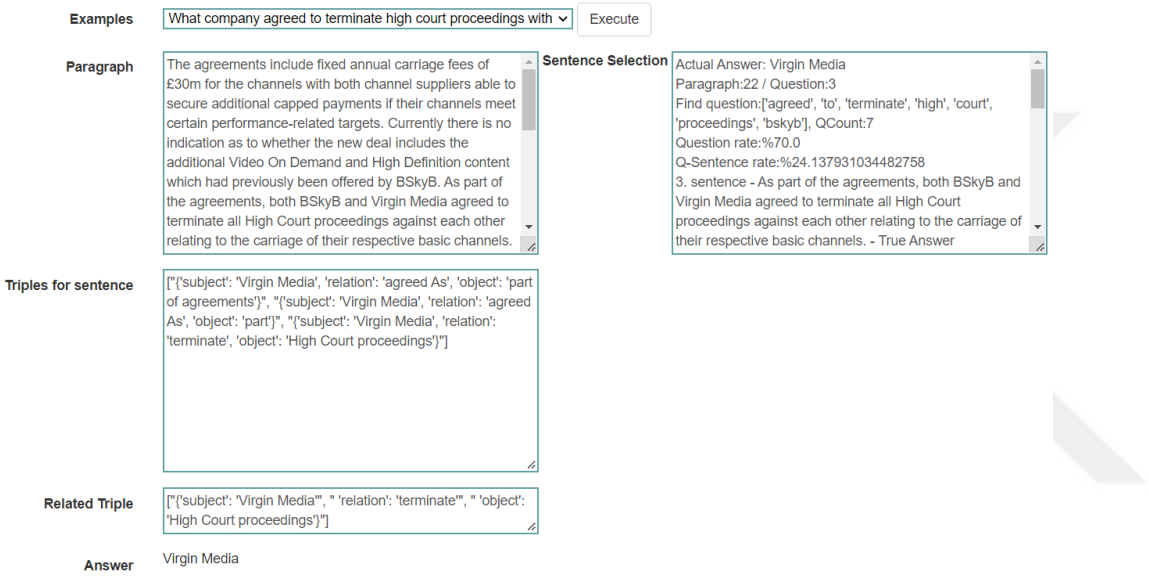


Figure 6.7: Demo page for Triple-based QAS

7 EXPERIMENTS

The analysis of the SQuAD, analysis of the sentence selection the performed NLP-QAS and TRP-QAS are explained under different sections. Preprocessed terms, question distribution and the distribution of the question pronouns on SQuAD were analyzed in the first subsection. The second subsection describes analysis of sentence selection containing answers according to QTP. Then, the analysis of answer detection was performed with RNP methods on the selected sentences in the third subsection. Last subsection describes the analysis of TRP-QAS with creating triples.

7.1 Analysis of SQuAD

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 of the development set (Dev_set) are shown in Figure 7.1. When the stopwords

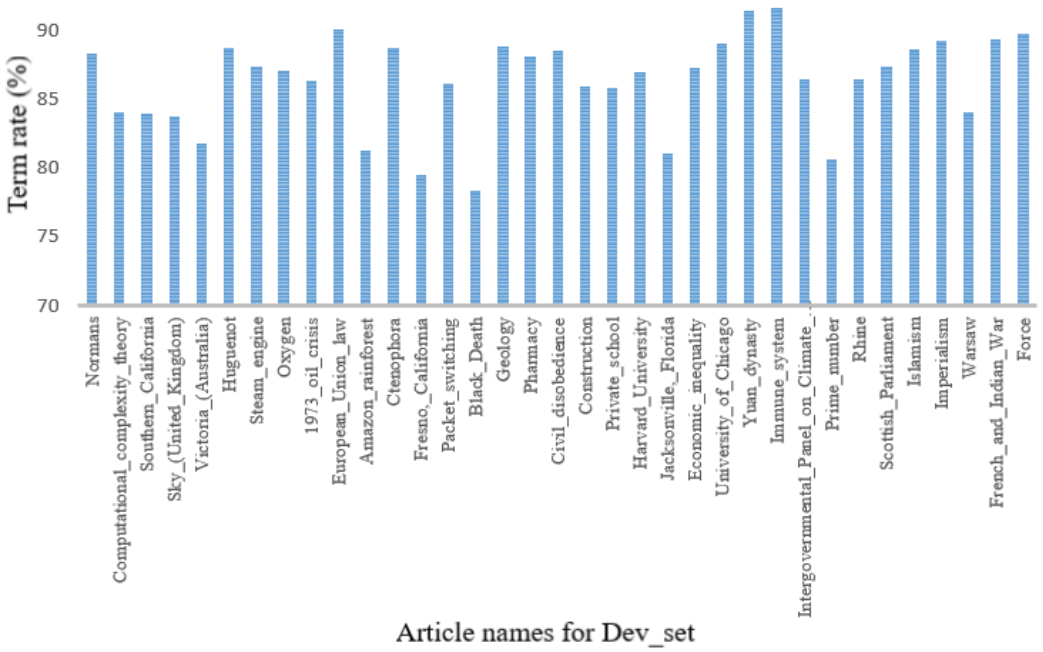


Figure 7.1: Term rate (%) that can be used after preprocessing for each article in the Dev_set

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 training set (Train_set) is 86.5%. Since it is not appropriate to represent 442 articles for Train_set in the figure, only the 35 articles in Dev_set is shown in Figure 7.1 for SquAD.

The question statistics in SQuAD are shown in Table 7.1. 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.

Table 7.1: Question statistics of the dataset

	Dev_set (%)	**Train_set (%)**
Total question count	11873 (%100)	130319 (%100)
Question count without answer	5945 (%50.08)	43498 (%33.38)
Question count with answer	5928 (%49.92)	86821 (%66.62)

The distribution of the question pronouns have been analyzed for questions that have a answer. 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 7.2 indicates the distributions of the 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. The table also shows that the "Others" label in the table is used for questions that do not contain question pronouns. These question types include filling in the blanks (_____ in both liquid and gas form can fastly result in an explosion.), choosing with or conjunction (Is fertilization internal or exeternal in most species?), yes/no (Does bskyb carry any control over a channels content?) questions. In addition, there are questions with missing question pronouns (name a type of Toyota compact trucks?) in the "Others" label.

Table 7.2: Distribution of question pronouns for questions with an answer

Question pronouns	**Dev_set**	**Train_set**	**Total pronoun**	**Rate (%)**
What	3561	49123	52684	56.80
Who	537	9813	10350	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

Since the questions that have no answer for TRP-QAS will also be examined, the statistics of question pronouns were obtained in these questions. Table 7.3 indicates the distributions of the question pronouns. A similar distribution is seen in this table as in the previous Table 7.2. However, the rate of questions labeled ”Others” has decreased considerably.

Table 7.3: Distribution of question pronouns for questions that have no answer

Question pronouns	**Dev_set**	**Train_set**	**Total pronoun**	**Rate (%)**
What	3723	26667	30390	61.50
Who	619	5159	5778	11.69
How	605	3871	4476	9.06
When	462	3142	3604	7.29
Which	175	2166	2341	4.74
Where	225	1574	1799	3.64
Why	44	677	721	1.46
Others	62	242	304	0.62

In the final stage of the SQuAD analysis, the NER process was applied. All entities with the NER label were analyzed for paragraphs of SQuAD articles. These labels and the number of entities containing these labels are given in Table 7.4. This table shows that many entities for ORG (Company,

institute, etc.), GPE (Country, city, etc.), PERSON (Person name&surname), DATE (Date, period name) labels are included in the SQuAD. Within the OTHERS tag, there are many tags such as ORDINAL (Ordinal number), LAW, PRODUCT, etc.

Table 7.4: The NER statistic for SQuAD

NER Labels	**Dev_set**	**Train_set**	**Total**	**Rate (%)**
ORG	3374	51153	54527	19.90
GPE	2684	43563	46247	16.88
PERSON	2317	38895	41212	15.04
DATE	1903	35969	37872	13.82
NORP	1703	27999	29702	10.84
CARDINAL	1604	23683	25287	9.23
OTHERS	842	14157	14999	5.48
LOC	680	8598	9278	3.39
FAC	225	4063	4288	1.57
WORK_OF_ART	139	3041	3180	1.16
EVENT	133	2881	3014	1.10
QUANTITY	170	2187	2357	0.86
MONEY	83	1572	1655	0.60
LANGUAGE	40	1016	1056	0.39
Total label count	15897	258071	273968	

7.2 Analysis of Sentence Selection

RSS 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. The sent_tokenize and punkt methods of NLTK libraries have been used for sentence parsing. In order to select the related sentence obtained by these methods, the previously explained operations (Section 5.1) have been performed in Figure 5.3. The

statistics of these methods for RSS are shown in Table 7.5. The punkt method, which gives a better result, has been selected for sentence parser in this study.

Table 7.5: The success of NLTK methods in RSS

	sent_tokenize		**punkt**	
	Dev_set	*Train_set*	*Dev_set*	*Train_set*
Sentence selection count	5668	81312	5688	81766
Total question count	5928	86821	5928	86821
Selection rate (%)	95.59	93.65	**95.93**	**94.17**

Then, it was understood that SQuAD has been parsed incorrectly into sentences due to a problem such as not having punctuation marks at the end of some sentences, starting the next sentence with a lowercase letter 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 RSS is shown in Table 7.6.

Table 7.6: The effect of the SDP method to RSS (Y: Yes, N: No)

	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	81766	5878	82955	5720	82288	5879	85956
Total question count	5928	86821	5928	86821	5928	86821	5928	86821
Selection rate (%)	95.93	94.17	99.15	98.99	96.48	94.78	**99.16**	**99.00**

After SDP method for correction, lemmatization technique was performed to all questions, answers and sentences in the entire dataset. Then, its effect has been analyzed for RSS. After lemmatization, paragraphs are parsed into sentences with “punkt with SDP”. POS tagging is optionally analyzed as it can be used in lemmatization. The effect of lemmatization is shown in Table 7.7. Table shows that the lemmatization was less successful in sentence selection compared to the previous Table 7.6.

Table 7.7: The effect of lemmatization on RSS 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	63004	4660	65713	4509	62862	4624	65470
Total question count	5928	86821	5928	86821	5928	86821	5928	86821
Selection rate (%)	76.5	72.56	**78.59**	**75.68**	76.04	72.40	77.98	75.4

To summarize all the performed operations for RSS, the chart regarding the selection rate of sentences is shown in Figure 7.2. This figure shows that the most successful method of RSS is "punkt with SDP". When the lemmatization has performed in the entire dataset, the selection rate of sentences has decreased approximately 20%.

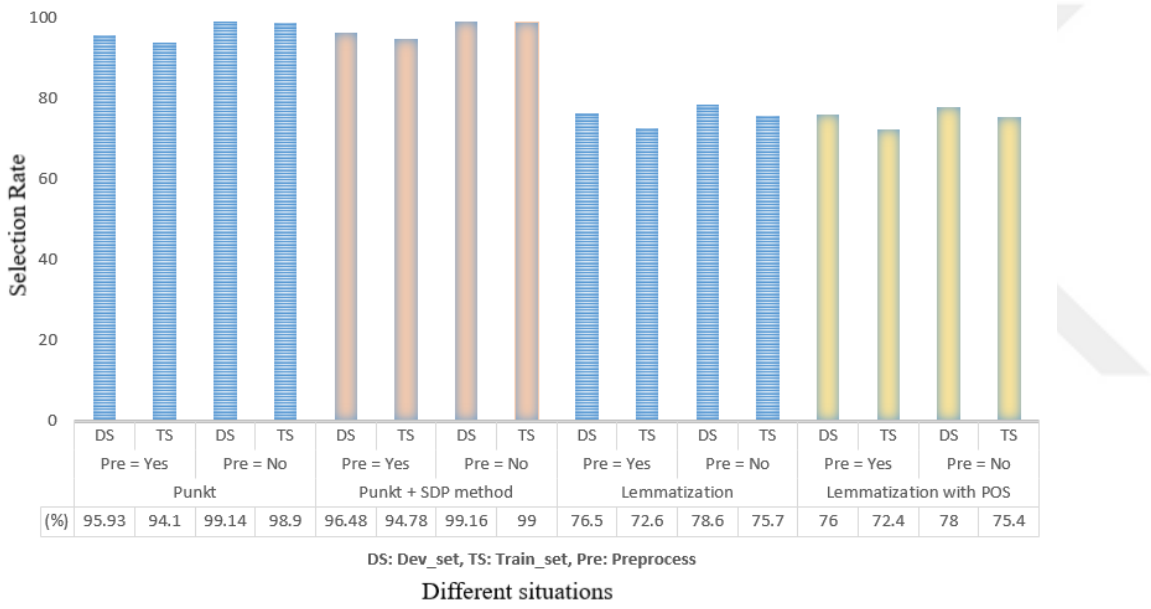


Figure 7.2: The success of performed methods for RSS

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 Figure 7.3. This 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.

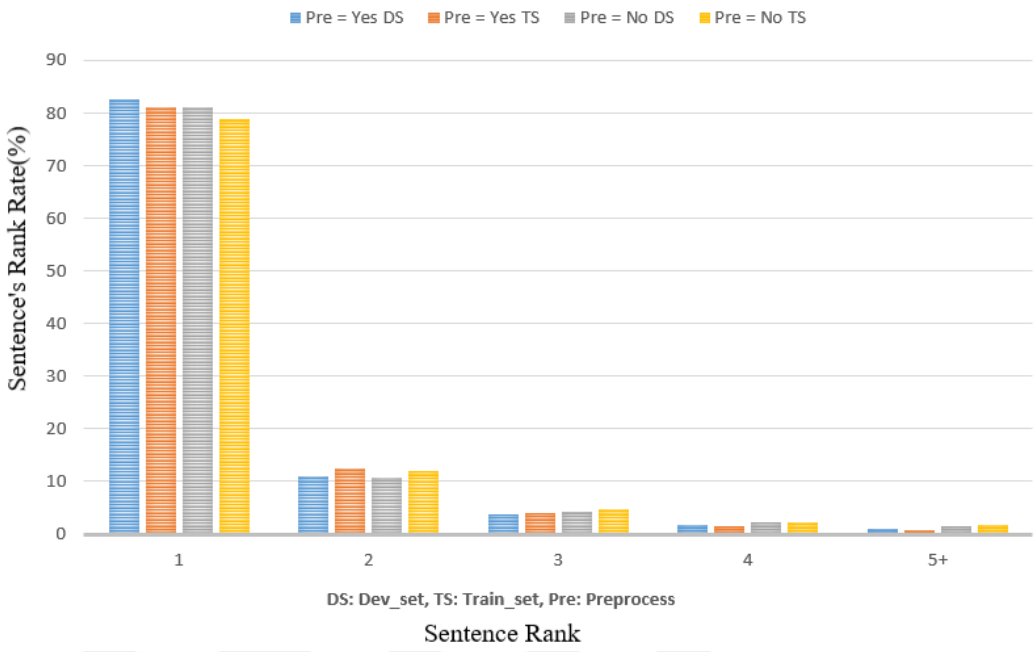


Figure 7.3: Statistics on the rank of the selected sentence according to QTP

Finally, the created chart for the QTP values of the chosen sentences is shown in Figure 7.4. 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.

7.3 Analysis of NLP-QAS

The proposed NLP-QAS is analyzed under different headings in this section. Answer detection analysis is performed with RNP methods on the sentences selected under the first sub-heading. In the other sub-heading, analysis is carried out on the BERT model with RNP methods.

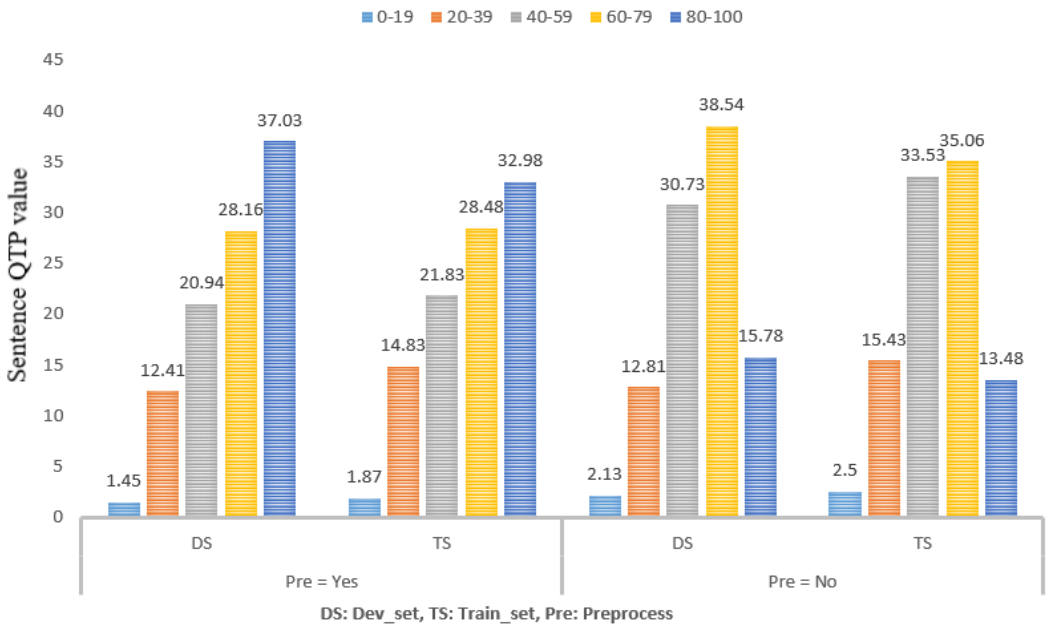


Figure 7.4: QTP range statistics for the selected sentence

7.3.1 Analysis of Answer Detection

Answer detection has been analyzed with RNP methods according to the selected sentences. The accuracy has been determined as the criteria for performance metric. 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 accuracy of these methods on answer detection for the selected sentences is shown in Table 7.8. This table shows that while RNP methods using the punkt method for sentence parsing detected answers between 12.5% and 14.9% accuracy, RNP methods provided about 3% accuracy increase with the effect of SDP. In the next stages for this study, 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

Table 7.8: Statistics of answer detection on selected 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	12157	736	12713	869	15210	889	15839
Total related sentence	5688	81766	5878	85955	5720	82288	5879	85956
Accuracy (%)	12.59	14.87	12.52	14.79	**15.19**	**18.48**	15.12	18.42

conjunction with POS tagging. The accuracy of these operations are shown in Table 7.9. Compared to the success of Table 7.8, the accuracy rate has increased by approximately 1.5%.

Table 7.9: 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	10190	519	9869	535	10299
SPOS	289	4574	294	4695	300	4650	304	4775
Total answer detection	926	16052	949	16704	953	16418	975	17089
Total related sentence	5720	82288	5879	85956	5720	82288	5879	85956
Accuracy (%)	16.18	19.5	16.14	19.42	**16.66**	**19.95**	16.58	19.88

In addition, the lemmatization technique has been performed to the question, answer and sentence after RSS and before the answer detection. Table 7.10 indicates the accuracy of RNP methods for this technique. It has been observed that the accuracy has decreased slightly compared to Table 7.9.

Finally, the lemmatization technique has been performed for the entire dataset before the RSS and answer detection. The effect of this technique for answer detection is shown in Table 7.11. This table shows that this technique has had a negative effect on answer detection as well as RSS. The count of answer detection is very low compared to Table 7.9. The reason for this is that the sentence structure has changed due to lemmatization.

Table 7.10: The effect of lemmatization after RSS 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	14905	878	16704	868	15000	889	15597
Total related sentence	5720	82288	5879	85956	5720	82288	5879	85956
Accuracy (%)	14.95	18.11	14.93	18.03	**15.17**	**18.22**	15.12	18.14

Table 7.11: 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	11508	717	12027	708	11539	721	11976
Total related sentence	4536	63004	4660	65713	4509	62862	4624	65470
Accuracy (%)	15.43	18.34	15.38	18.3	**15.70**	**18.35**	15.59	18.3

The accuracy of the RNP methods for answer detection is shown in Figure 7.5 according to all situations. Applying lemmatization with POS tagging was more successful for each method in answer detection with RNP. Hence, only this method for lemmatization is shown in this figure. The accuracy values are the total rate at which the all RNP methods for answer detection. Figure 7.5 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.

For the most successful method in Figure 7.5, the statistics of answering the questions of the RNP methods together were obtained. The success of RNP methods in answering the same questions together is given in Figure 7.6. This figure shows that only SNER was the RNP method that answered the

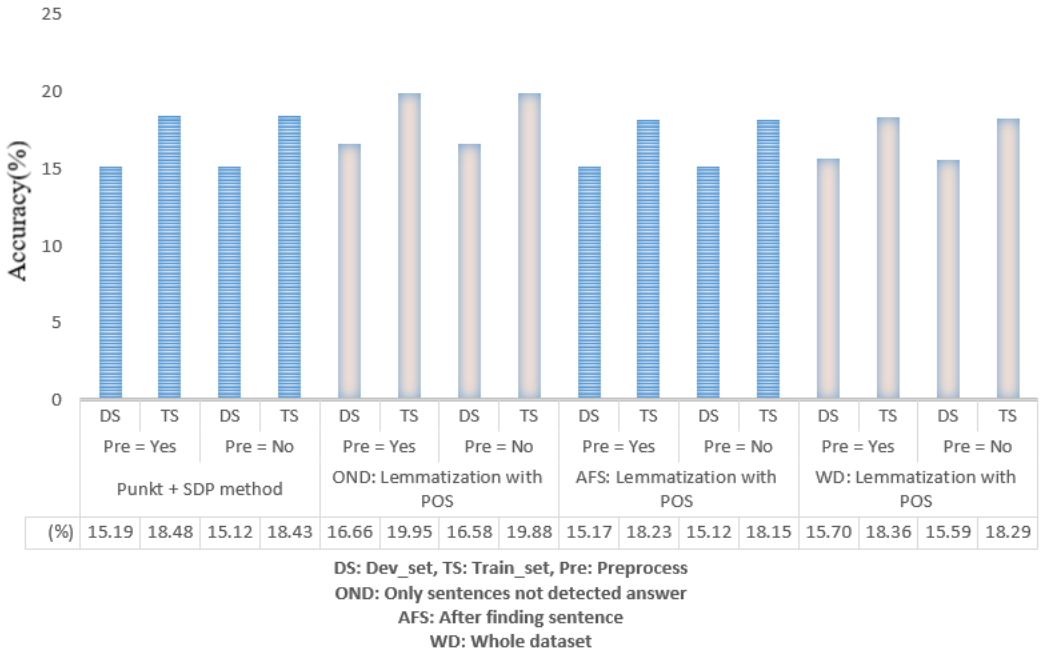


Figure 7.5: Statistics of all situations for answer detection

most questions, with 37.5% to 48.6% answering the question correctly. The most successful methods, in which both methods detect the correct answers, are SNER-SPOS, these methods answer all questions correctly between 12.4% and 17.5%. The number of questions that all methods can find together is very low, at 0.3%.

According to the Table 7.9, the most successful situation was chosen and this situation was compared with applying WordNet and CoRes process in the analysis. First, synonyms of question terms were obtained with WordNet. The synonyms of the question terms that are not in the sentences in which the question terms are searched in the paragraph are also searched within the sentences. Accordingly, the QTP value of the question terms has been updated and there is a possibility that the sentence to be selected will change. The sentences obtained with WordNet's synonymy were used for answer detection with RNP methods. As a result of applying the selected sentences according to Wordnet to three NLP methods by applying lemmatization (with

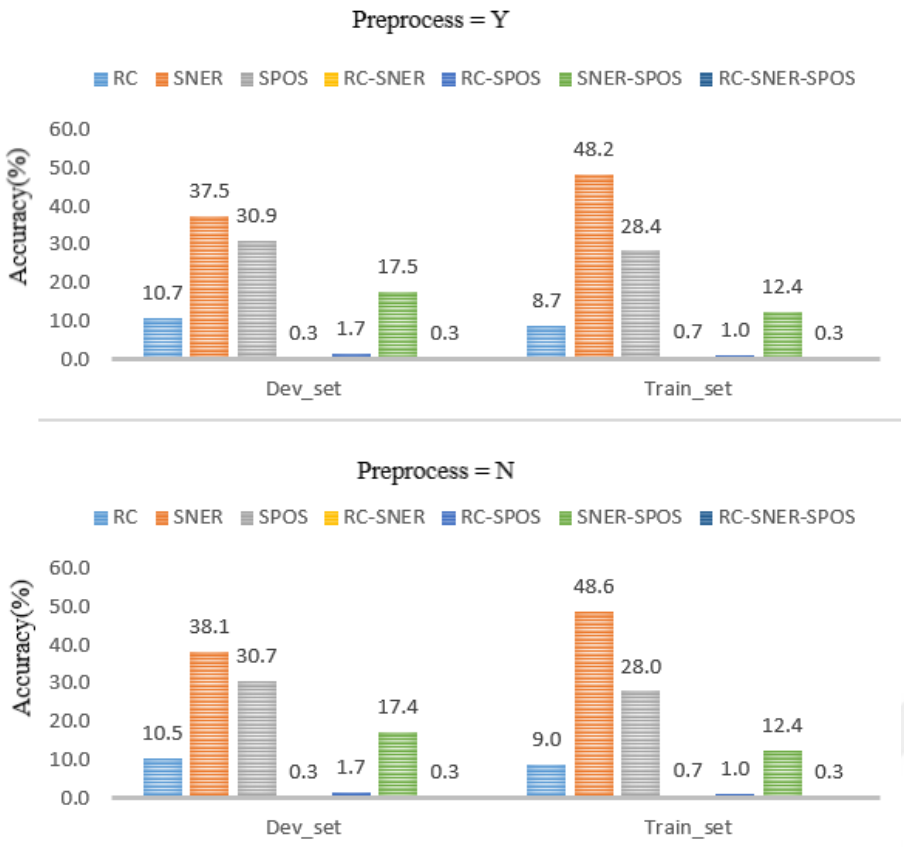


Figure 7.6: Together statistics of RNP methods for answer detection

POS tagging), its success in answer detection is shown in Table 7.12. When this table is examined,an increase has been achieved in the total number of selected sentences according to the question terms and in the answer detection compared to Table 7.9. All proposed RNP methods detected more answers than Table 7.9.

In the next step, CoRes libraries were used to detect the noun form of pronouns. The texts obtained with the libraries were given to the StanfordOpenIE library and used for subject, relation, object (SRO) extraction. CoRes process has been applied to all SQuAD. The success of NeuralCoref and AllenNLP libraries were analyzed for answer detection in selected sentences.

For the CoRes process, analysis was first performed through the NeuralCoref library. SQuAD has been updated with NeuralCoref and it is aimed to

Table 7.12: The success of the selected sentences with the WordNet application in answer detection

	Lemmatization with POS tagging			
	Preprocess = Y		*Preprocess = N*	
	Dev_set	*Train_set*	*Dev_set*	*Train_set*
RC	145	2028	147	2110
SNER	521	10022	535	10322
SPOS	304	4692	307	4792
Total answer detection	970	16742	989	17224
Total related sentence	5773	83765	5889	86274
Accuracy (%)	**16.80**	**19.99**	16.79	19.96

determine the answer with the SRO information extracted via StanfordOpenIE of this dataset. For this, the question terms are searched in the SRO triples. When there is only one part left in the triples, this part is accepted as the candidate answer and is compared with the actual answer. The success achieved as a result of creating SROs of SQuAD updated with NeuralCoref is shown in Table 7.13. In addition to the previous processes, this process was applied only on the sentences whose answer could not be detected.

Table 7.13: The success achieved as a result of creating SROs of SQuAD with CoRes (with NeuralCoref) applied

	Lemmatization with POS tagging			
	Preprocess = Y		*Preprocess = N*	
	Dev_set	*Train_set*	*Dev_set*	*Train_set*
Answer detection with WordNet	970	16742	989	17224
Detection count with NeuralCoref	19	197	18	196
Total answer detection	989	16939	1007	17420
Total related sentence	5773	83765	5889	86274
Accuracy (%)	**17.13**	**20.22**	17.10	20.19

Analysis was also performed with the AllenNLP library in the CoRes process. For SQuAD updated with AllenNLP, SRO information was extracted and answer detection was applied. As a result of the creation of SROs of the dataset updated with AllenNLP, the success achieved in answer detection is shown in Table 7.14. All operations were applied on the sentences whose answer could not be detected, over the previous operations as in Table 7.13.

Table 7.14: The success achieved as a result of creating SROs of SQuAD with CoRes (with AllenNLP) applied

	Lemmatization with POS tagging			
	Preprocess = Y		*Preprocess = N*	
	Dev_set	*Train_set*	*Dev_set*	*Train_set*
Previous answer detection count	970	16742	989	17224
Detection count with AllenNLP	18	197	17	195
Total answer detection	988	16939	1006	17419
Total related sentence	5773	83765	5889	86274
Accuracy (%)	**17.11**	**20.22**	17.08	20.18

The effect of all applied processes on answer detection for WordNet and CoRes is shown in Figure 7.7. This figure shows that the most successful method has been answer detection with SRO extraction as a result of using the NeuralCoref library for CoRes although close with AllenNLP result. Since the NeuralCoref library was more successful for CoRes, only NeuralCoref was used for analysis of TRP-QAS.

7.3.2 Analysis of BERT with RNP Methods

The BERT LM was tested with RNP methods 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 for benchmark.

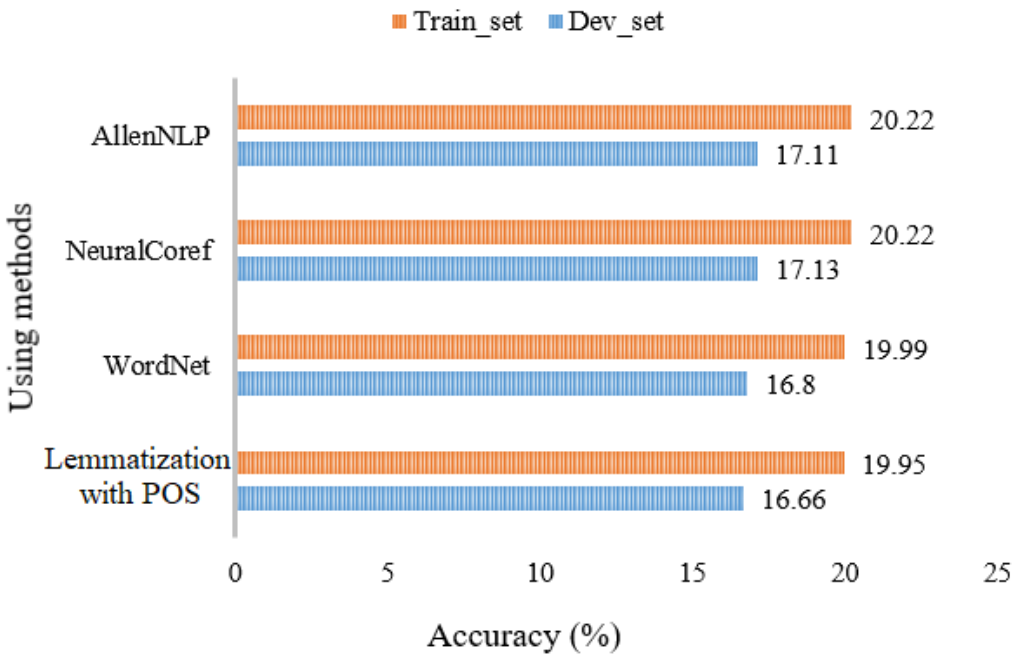


Figure 7.7: The effect of applied WordNet and CoRes processes on answer detection

The BERT-Base and BERT-Large pre-trained 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 et al., 2018). 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 (BBU)[19], BERT-Base-Cased (BBC)[20], BERT-Large-Uncased (BLU)[21] ve BERT-Large-Cased (BLC)[22] pre-trained models have been tested with Dev_set. The accuracy value for answer detection of these pre-trained LMs is shown in Table 7.15. Both BERT-Large models achieved an accuracy over 82.9%, while BERT-Base models achieved less than 77.1%

[19]https://huggingface.co/twmkn9/bert-base-uncased-squad2
[20]https://huggingface.co/deepset/bert-base-cased-squad2
[21]https://huggingface.co/bert-large-uncased-whole-word-masking-finetuned-squad
[22]https://huggingface.co/bert-large-cased-whole-word-masking-finetuned-squad

accuracy.

Table 7.15: The accuracy of pre-trained BERT models for answer detection

	BBU	**BBC**	**BLU**	**BLC**
True answer detection	4570	4319	4915	4958
Total question count	5928	5928	5928	5928
Accuracy (%)	77.09	72.88	82.91	**83.64**

The answers of 1013 questions for BLU, 970 questions for BLC, 1358 questions for BBU and 1609 questions for BBC couldn't be correctly answered for these pre-trained LMs (Table 7.15). The RNP methods have been used sequentially for answer detection of these questions. RNP methods have been performed to the selected sentences according to QTP. The accuracy of these methods on BERT models are shown in Table 7.16. This table shows that these methods have detected answers that the pre-trained LMs couldn't answer correctly.

Table 7.16: The accuracy of RNP methods on pre-trained 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
Accuracy (%)	7.88	8.76	7.01	6.60

RNP methods have been performed one by one to analyze the accuracy of each method. The accuracy of each method is shown in Table 7.17. This table shows that the most successful method was SPOS, and the most unsuccessful was RC for each method. SPOS method detected more than 50% of the correct answers.

Table 7.17: The accuracy of each RNP method separately

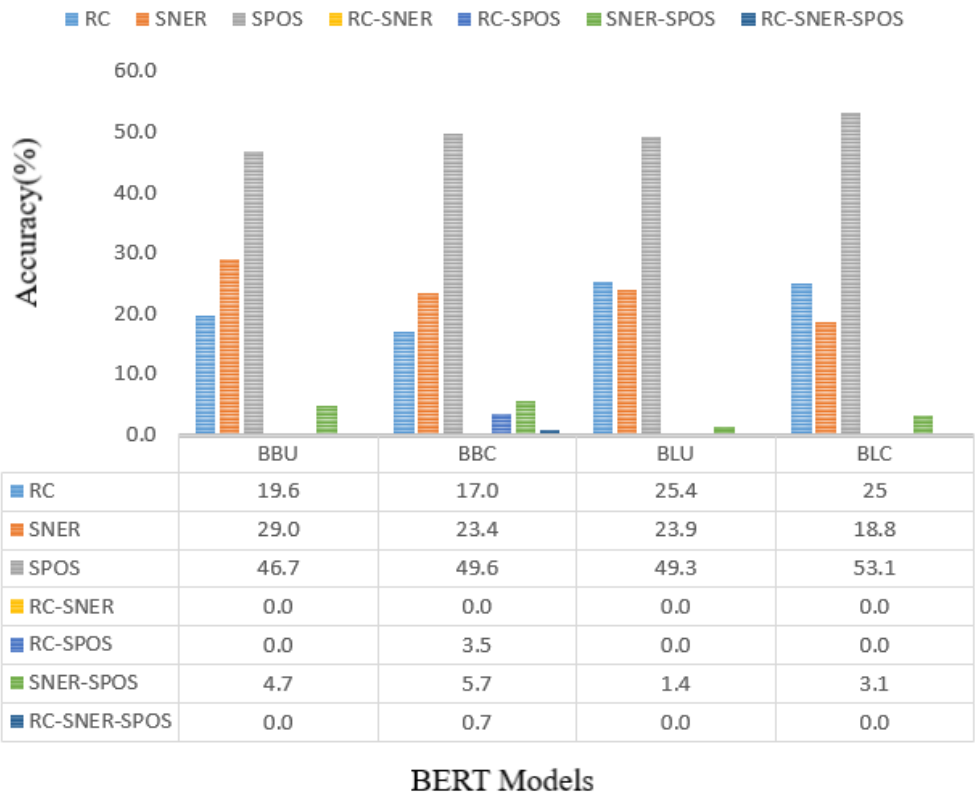
	BBU (%)	**BBC (%)**	**BLU (%)**	**BLC (%)**
RC	21 (19.62)	30 (21.27)	18 (25.35)	16 (25)
SNER	36 (33.64)	42 (29.79)	18 (25.35)	14 (21.875)
SPOS	55 (51.40)	84 (59.57)	36 (50.71)	36 (56.25)
Total answer detection	107	141	71	64

On the questions answered incorrectly by the BERT model, the ability of all RNP methods to answer the same question together was analyzed. Statistics for the co-answerability of the questions for the RNP are given in Figure 7.8. This figure shows that the rate of RNP methods to answer the same question together is quite low, only RC-SPOS methods for BBC and SNER-SPOS methods were able to answer the answers together. The answer that all methods answered correctly is only available for the BBC, but the rate is quite low with 0.7%. RNP methods mostly answered all questions by itself, SPOS was the most successful among these methods.

Using pre-trained BERT LMs and RNP methods together has increased the accuracy. The effects of RNP methods for pre-trained BERT models are shown in Table 7.18. This table shows that the accuracy has increased due to combined use. This table indicates that NLP techniques aren't used enough in the BERT LMs.

Table 7.18: The effects of using pre-trained BERT models and RNP methods together on answer detection

	BBU	**BBC**	**BLU**	**BLC**
Correct answer count of BERT	4570	4319	4915	4958
Correct answer count of RNP methods	107	141	71	64
Total answer detection	4677	4460	4986	5022
Total question count	5928	5928	5928	5928
Accuracy (%)	**78.89**	**75.23**	**84.11**	**84.71**



	BBU	BBC	BLU	BLC
RC	19.6	17.0	25.4	25
SNER	29.0	23.4	23.9	18.8
SPOS	46.7	49.6	49.3	53.1
RC-SNER	0.0	0.0	0.0	0.0
RC-SPOS	0.0	3.5	0.0	0.0
SNER-SPOS	4.7	5.7	1.4	3.1
RC-SNER-SPOS	0.0	0.7	0.0	0.0

Figure 7.8: Co-answerability statistics of the RNP method on BERT model

As a result, the accuracy of pre-trained BERT LMs with or without RNP methods are shown Figure 7.9. This figure shows that the accuracy 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 LMs.

After analyzing the success of the RNP methods on the pre-trained BERT models, question pronouns that these models could not answer have been analyzed. This analysis results are shown in Table 7.19. This table shows that these models can hardly answer open-ended questions such as ”why, what”. Models are successful for questions like ”who, when” 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

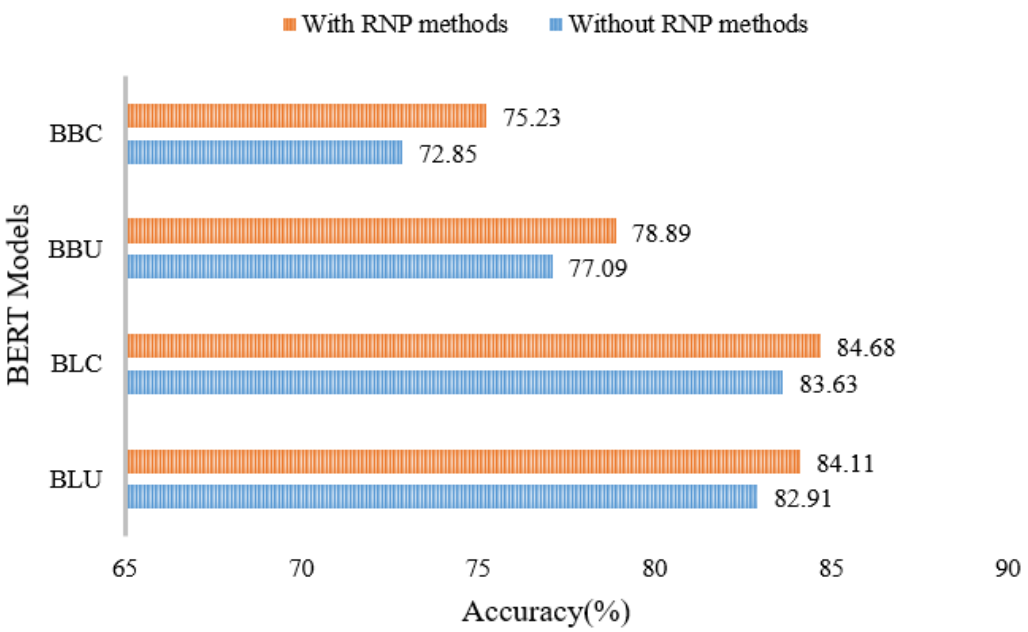


Figure 7.9: Statistics of the pre-trained BERT models for answer detection

Table 7.19: 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
How	145 (22.62)	166 (25.89)	128 (19.96)	105 (16.38)	641
Who	93 (17.31)	103 (19.18)	66 (12.29)	57 (10.61)	537
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
Why	49 (51.04)	45 (46.87)	42 (43.75)	45 (46.87)	96
Others	29 (46.77)	36 (58.06)	23 (37.09)	24 (38.70)	62

by these methods are shown in Table 7.20. This table indicates 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'.

Table 7.20: Distribution of question pronouns for 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

7.3.3 Discussion

SQuAD is a benchmark platform for QAS. It provides a reading comprehension dataset for testing the performance of QAS. Generally, they are based on DL models based on BERT, ALBERT, ELMo, etc. The motivation of this study is to focus on the questions that BERT models don't answer. We developed three natural language based methods, namely RNP, that increases the performance of pre-trained BERT models for QAS.

In Table 7.21, we outlined the basic differences between the proposed RNP methods and the pre-trained BERT model. As seen in Example 1, the proposed RC method is applied for the question that cannot be answered by the pre-trained BERT model, but the pre-trained BERT model did not find the answer correct, while the RC detected the answer correctly. For Example 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 for Example 2 and Example 3. 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 RNP methods can easily be applied to increase the accuracy of the pre-trained BERT models.

Table 7.21: 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 kilometres (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 kilometres (764 miles). In 1932 the German encyclopedia Knaurs Lexikon stated the length as 1,320 kilometres (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 kilometres (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

For answer detection, the success of RNP methods was first analyzed separately. As a result of the analyses, while the most successful method was SNER when applied to all SQuAD, SPOS was most successful when applied only to questions answered incorrectly by BERT. Here, the reason why these SNER and SPOS methods are more successful is that they search answers according to the label by examining the question pronoun. The RC method only applies deletion process in the selected sentence according to the question terms, it does not pay attention to the meaning of any term.

7.4 Analysis of Triple-based QAS

Analysis of LMs on SQuAD, analysis of proposed TRP-QAS extension on questions that LMs cannot answer correctly are mentioned under sub-headings. It is also discussed in the results for this extension.

7.4.1 Analysis of Pre-trained LMs

Initially, QA analysis of the pre-trained LMs used in the system was performed on SQuAD. The development set of SQuAD was used for this analysis. The exact match, F1-score, and accuracy value evaluation metrics of all pre-trained LMs were obtained through the Haystack[23] library. The results of these metrics are shown in Table 7.22. This table shows that for SQuAD, RoBERTa model had the best F1-score, while SpanBERT model had the best accuracy.

The accuracy values of all LMs were also calculated for the questions both have answers and no answers. The statistics of the accuracy values of these LMs are given in Figure 7.10. When the accuracy values of LMs are examined, the most successful models are SpanBERT and ELECTRA. However, while SpanBERT is more successful for questions with an answer, RoBERTa and ALBERT are more successful for questions that no answer. In addition,

[23]https://haystack.deepset.ai/

Table 7.22: The evaluation metrics of LMs

	Exact Match (%)	**F1-score (%)**	**Accuracy (%)**
BERT	70.64	75.23	75.90
ALBERT	73.39	80.28	77.50
ELECTRA	75.20	80.71	79.41
SpanBERT	76.15	81.06	**80.70**
RoBERTa	**78.47**	**82.65**	79.33

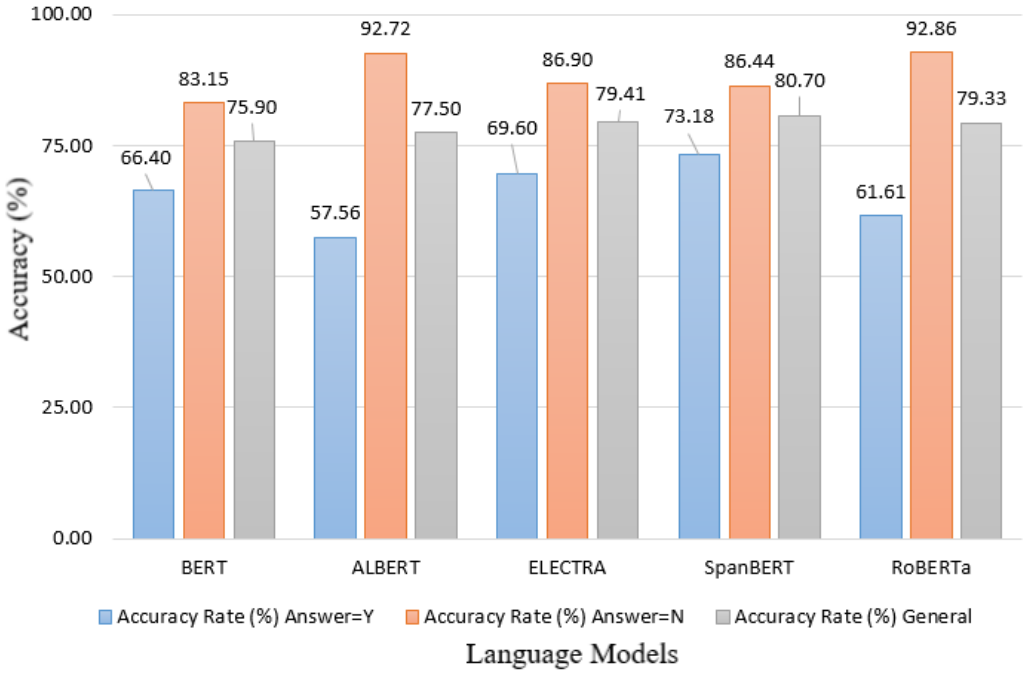
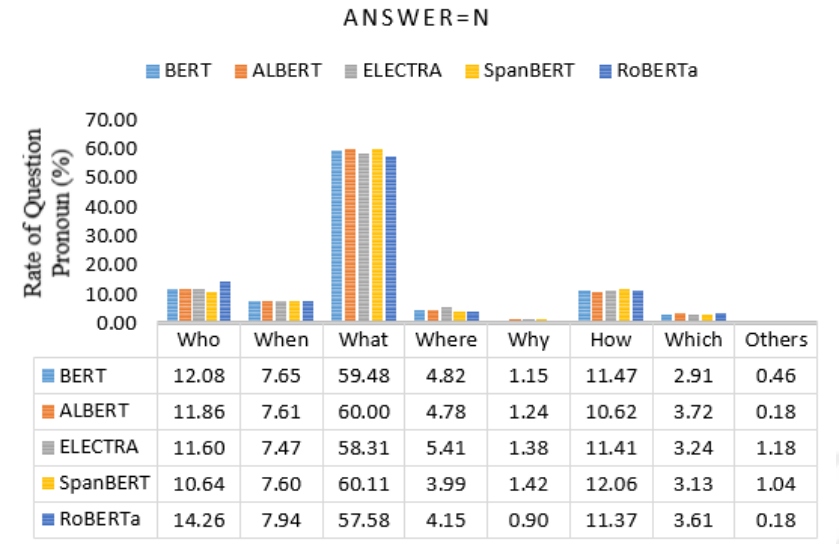


Figure 7.10: The statistics of the accuracy of LMs according to answer status

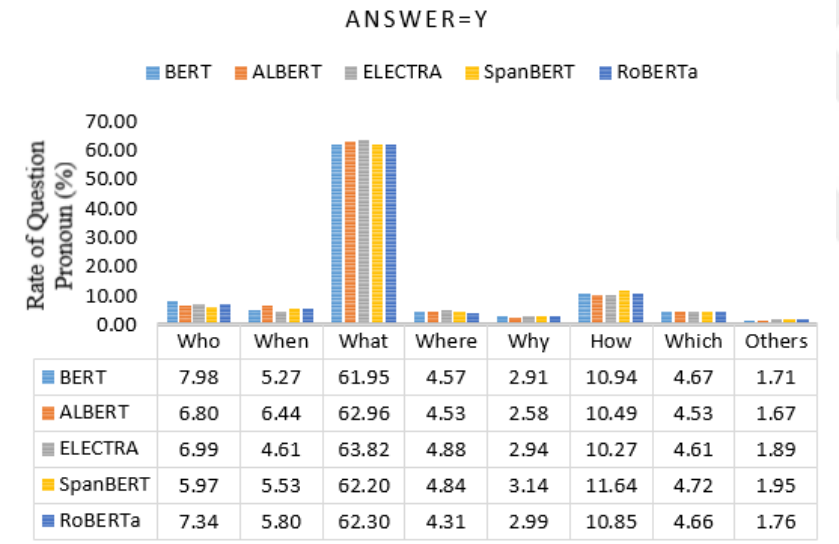
this figure shows that question has no answers (Answer=N) are detected more accurately than the question has answers (Answer=Y)(Total Answer=Y Count: 5928, Total Answer=N Count: 7763).

Finally, the question pronouns of the questions that answered incorrectly by LMs were analyzed. The distribution of question pronoun for these situation are given in Figure 7.11. This figure shows that for all LMs in each answer type, the question pronoun ”what” was answered incorrectly with a very high rate of

over 57.5%. This is because this question pronoun is open-ended. The number of question pronoun for questions that all LMs could not answer correctly are also given in Table 8.4 in the Appendix section 8.



	Who	When	What	Where	Why	How	Which	Others
BERT	12.08	7.65	59.48	4.82	1.15	11.47	2.91	0.46
ALBERT	11.86	7.61	60.00	4.78	1.24	10.62	3.72	0.18
ELECTRA	11.60	7.47	58.31	5.41	1.38	11.41	3.24	1.18
SpanBERT	10.64	7.60	60.11	3.99	1.42	12.06	3.13	1.04
RoBERTa	14.26	7.94	57.58	4.15	0.90	11.37	3.61	0.18



	Who	When	What	Where	Why	How	Which	Others
BERT	7.98	5.27	61.95	4.57	2.91	10.94	4.67	1.71
ALBERT	6.80	6.44	62.96	4.53	2.58	10.49	4.53	1.67
ELECTRA	6.99	4.61	63.82	4.88	2.94	10.27	4.61	1.89
SpanBERT	5.97	5.53	62.20	4.84	3.14	11.64	4.72	1.95
RoBERTa	7.34	5.80	62.30	4.31	2.99	10.85	4.66	1.76

Figure 7.11: The distribution of the question pronoun of questions that answered incorrectly by LMs

7.4.2 Analysis of Triples

After performing the analysis of all pre-trained LMs on SQuAD, TRP-QAS was performed on the questions that these LMs answered incorrectly. By utilizing the operations in the previously mentioned Section 7.2, a

triple creation process was performed on each question's selected sentence (Answer=Y) or paragraph (Answer=N). The statistics obtained for LMs as a result of searching for answers on these triples are shown in Table 7.23. This table includes the number of answers found with triples, the number of questions answered incorrectly by the LMs, and the accuracy value of the TRP-QAS on these questions for each model. For all LMs, the correct answer rate for questions with an answer is relatively low, while the accuracy rate for questions that no answers is higher. This table shows that answers can be accessed with the TRP-QAS for both questions with an answer and no answers.

Table 7.23: Statistics of LMs as a result of applying a TRP-QAS

Models	Finding answer with triples			Answered incorrectly			Accuracy Rate (%)		
	Answer=Y	***Answer=N***	***Total***	***Answer=Y***	***Answer=N***	***Total***	***Answer=Y***	***Answer=N***	***Total***
BERT	11	1019	1130	1992	1308	3300	0.55	77.91	31.2
ALBERT	16	447	463	2516	565	3081	0.64	79.12	15.0
ELECTRA	18	808	826	1802	1017	2819	1.00	79.45	29.3
RoBERTa	12	437	449	2276	554	2830	0.53	78.88	15.9
SpanBERT	12	815	827	1590	1053	2643	0.75	77.40	31.29

Which question pronouns were found in the answered questions were analyzed and statistics were obtained for all question pronouns. Table 7.24 shows that the TRP-QAS for questions with an answer detects correctly the question pronoun "what" more often, but the rate of answering the questions is low for these pronouns. However, the TRP-QAS extension is quite successful in questions that no answer. Although this extension highly determined that the answer is not in the paragraph for all question pronouns and correctly answers a large number of "what" question pronouns.

In addition, the success of our TRP-QAS extension was analyzed by applying the stopwords removal pre-process to the question. It is thought that the stopwords in the question affect the system negatively when searching in triples. Since this process negatively affects the triple extraction, this preprocessing was applied only to the question, not the paragraph or sentence.

Table 7.24: Statistics of question pronouns answered correctly according to LMs (no pre-processing)

	Answer = Y														
	BERT			**ALBERT**			**ELECTRA**			**RoBERTa**			**SpanBERT**		
Question pronouns	*True*	*Total*	*Rate*	*True*	*Total*	*Rate*	*True*	*Total*	*Rate*	*True*	*Total*	*Rate*	*True*	*Total*	*Rate*
Who	1	159	0.62	1	171	0.58	2	126	1.58	2	167	1.20	0	95	0
When	1	105	0.95	2	162	1.24	1	83	1.20	0	132	0	1	88	0.12
What	8	1234	0.64	12	1584	0.76	13	1150	1.13	7	1418	0.48	10	989	1.01
Where	1	91	1.10	1	114	0.88	1	88	1.14	1	98	1.02	1	77	1.29
Why	0	58	0	0	65	0	0	53	0	0	68	0	0	50	0
How	0	218	0	0	264	0	0	185	0	0	247	0	0	185	0
Which	0	93	0	0	114	0	1	83	1.20	2	106	1.88	0	75	0
Others	0	34	0	0	42	0	0	34	0	0	40	0	0	31	0
	Answer = N														
Who	154	158	97.5	66	67	89.6	118	118	100	78	79	98.7	111	112	99.1
When	94	100	94	40	43	93	73	76	96	43	44	97.8	76	80	95
What	550	778	70.7	243	339	71.7	419	593	70.7	225	319	70.5	445	633	70.3
Where	60	63	95.2	27	27	100	53	55	96.4	22	23	95.7	41	42	97.6
Why	7	15	46.7	5	7	71.4	10	14	71.4	4	5	80	9	15	60
How	122	150	81.3	52	60	86.7	98	116	84.5	52	63	82.5	101	127	79.5
Which	26	38	68.4	13	21	61.9	25	33	75.8	12	20	60	21	33	63.6
Others	6	6	100	1	1	100	12	12	100	1	1	100	11	11	100

The results of the TRP-QAS after pre-processing are given in Table 7.25. When this table is examined, it is seen that the QAS answers more questions correctly than in Table 7.23. For data in Table 7.23 and Table 7.25, statistics were obtained for which question pronouns were found in correctly answered questions and for all question pronouns. These tables show that the question pronoun "what" is mostly answered for both operations.

Table 7.25: Statistics of LMs as a result of applying a triple-based system after question pre-processing

Models	**Finding answer with triples**			**Answered incorrectly**			**Accuracy Rate (%)**		
	Answer=Y	***Answer=N***	***Total***	***Answer=Y***	***Answer=N***	***Total***	***Answer=Y***	***Answer=N***	***Total***
BERT	20	1142	1153	1992	1308	3300	0.55	87.31	34.9
ALBERT	15	499	515	2516	565	3081	0.64	88.32	16.7
ELECTRA	13	907	925	1802	1017	2819	1.00	89.18	32.8
RoBERTa	15	489	501	2276	554	2830	0.53	88.27	17.7
SpanBERT	12	914	926	1590	1053	2653	0.75	86.80	35.04

Similarly, which question pronouns were found in the questions answered for Table 7.25 were analyzed and statistics are shown in Table 7.26. When the questions with an answer are examined, the system has determined the

Table 7.26: Statistics of question pronouns answered correctly according to LMs (with pre-processing)

	Answer = Y														
	BERT			**ALBERT**			**ELECTRA**			**RoBERTa**			**SpanBERT**		
Question pronouns	*True*	*Total*	*Rate*	*True*	*Total*	*Rate*	*True*	*Total*	*Rate*	*True*	*Total*	*Rate*	*True*	*Total*	*Rate*
Who	1	159	0.62	1	171	0.58	1	126	0.79	1	167	0.60	0	95	0
When	2	105	1.90	1	162	0.62	1	83	1.20	0	132	0	2	88	2.27
What	14	1234	1.13	11	1584	0.69	10	1150	0.87	11	1418	0.77	10	989	1.01
Where	2	91	2.20	1	114	0.88	1	88	1.14	1	98	1.02	0	77	0
Why	0	58	0	0	65	0	0	53	0	0	68	0	0	50	0
How	0	218	0	0	264	0	0	185	0	0	247	0	0	185	0
Which	1	93	1.07	1	114	0.88	0	83	0	2	106	1.88	2	75	2.67
Others	0	34	0	0	42	0	0	34	0	0	40	0	0	31	0
	Answer = N														
Who	155	158	98.10	66	67	98.50	118	118	100	78	79	98.73	111	112	99.1
When	98	100	98.00	43	43	100	76	76	100	44	44	100	80	80	100
What	648	778	83.30	284	339	83.77	499	593	84.15	267	319	83.70	522	633	82.4
Where	63	63	100	27	27	100	55	55	100	23	23	100	42	42	100
Why	9	15	60.00	5	7	71.42	12	14	85.71	5	5	100	11	15	73.3
How	129	150	86.00	53	60	88.33	104	116	89.66	53	63	84.13	107	127	84.3
Which	34	38	89.47	20	21	95.24	31	33	93.94	18	20	90.00	30	33	90.9
Others	6	6	100	1	1	100	12	12	100	1	1	100	11	11	100

question pronoun "what" in higher numbers, but the rate of finding answers is again very low for question pronouns. For questions that no answer, the TRP-QAS was quite successful, with all question pronouns fairly well determined that the answer was not in the paragraph.

The variation of the accuracy values of the LMs after applying all the operations with the TRP-QAS is given in Figure 7.12. The TRP-QAS answered questions that models could not answer correctly in QA. Thus, its positive effect on the LMs used for SQuAD has been demonstrated. Figure 7.12 also shows that when the TRP-QAS was performed, there was an increase between 3.3% and 7.5% for all LMs' accuracy.

7.4.3 Discussion

Before performing the TRP-QAS analysis, BERT, ALBERT, ELECTRA, RoBERTa, SpanBERT LMs pre-trained with SQuAD were analyzed in this study. Among the models, SpanBERT was the most successful with an accuracy value of 80.7%, while the most successful model was RoBERTa with

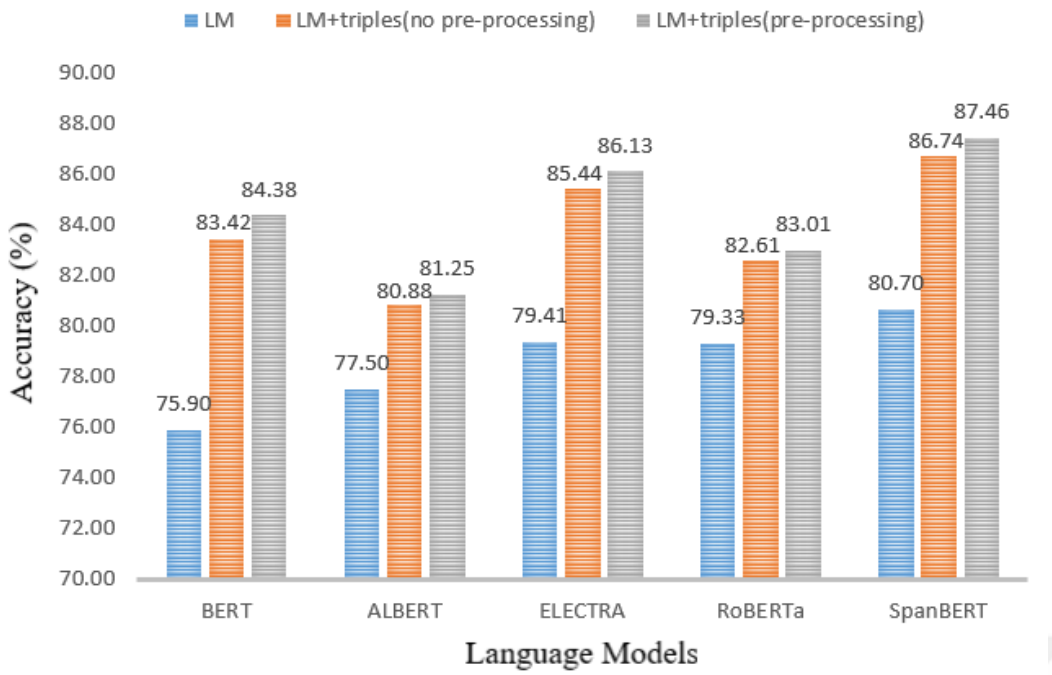


Figure 7.12: Accuracy values of LM's with TRP-QAS

an F1-score of 82.65%. All analyzed LMs outperformed the BERT model. This is because these models were developed based on BERT, taking into account the problems in the BERT structure. In addition, accuracy values were obtained separately for the question that has answers and no answers from the LMs (Figure 7.10). According to these values, LMs more accurately detected questions that have no answer. The reason for this is those LMs can't correctly extract answers from question pronouns that have answers. Especially since the question pronoun "what" is an open-ended expression, it is quite difficult to determine the answer.

In the next step, the success of the proposed TRP-QAS extension on questions answered incorrectly by LMs is analyzed. This proposed system for all LMs has been successful, increasing the accuracy. The TRP-QAS was more successful with questions that have no answers (Table 7.23 and Table 7.25). This is because triples are more likely not to be found, according to the question terms. In questions that have answers, the answer detection rate is lower

since determining the answer from the question pronoun is a more complex process. Moreover, the success of TRP-QAS on questions answered incorrectly by LMs is given in Figure 7.13. The TRP-QAS extension determined the correct answers between 15% and 35.2% on the questions that LMs answered incorrectly. This figure also shows that TRP-QAS performed the best answer detection in the BERT model, while ALBERT was the least successful. The reason for this is that the question count analyzed for BERT is higher than other LMs.

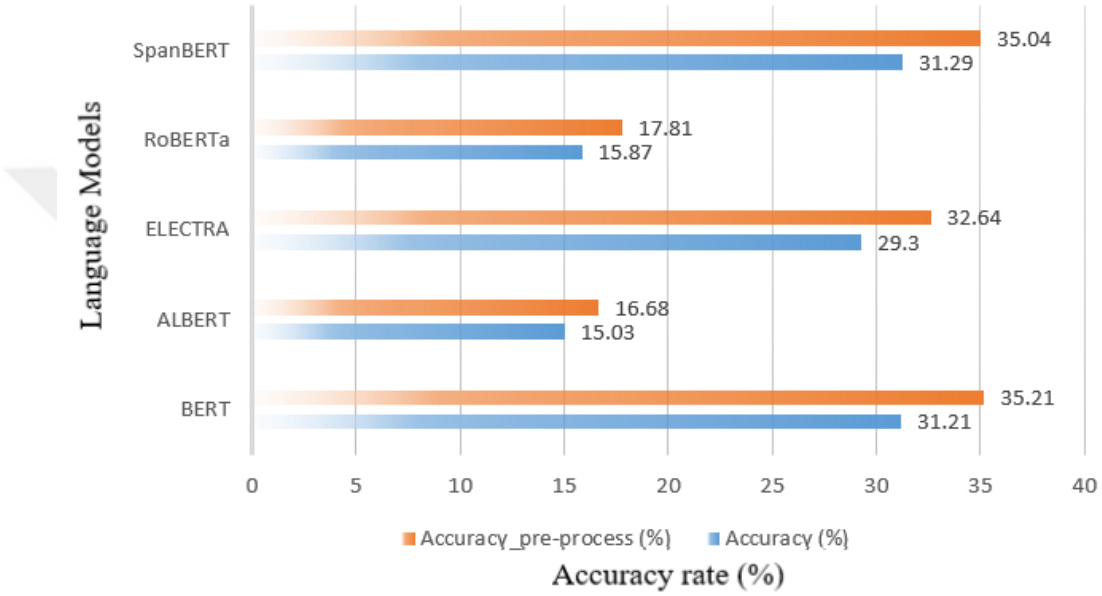


Figure 7.13: TRP-QAS accuracy values for questions answered incorrectly by LMs

The overall success of TRP-QAS increased the accuracy of the LMs between 3.3% and 7.5% (Figure 7.12). The greatest increase in accuracy has been in the BERT model because there are more questions the model answers incorrectly. Thus, more questions were analyzed with TRP-QAS for BERT compared to other LMs. The least increase in success was in the RoBERTa model. The reason for this is that the TRP-QAS analyzed fewer questions than other models.

It is thought that stopwords in questions may have a negative impact on the success of the TRP-QAS extension. Because these stopwords are searched within the triples and when found, they occupy the triple as false. Therefore, the accuracy values of the LMs were analyzed by removing the stopwords in the questions. With this pre-process, the accuracy values of LMs are increased between 0.4% and 0.9% compared to no pre-process on questions (Figure 7.12).

As a result, when the accuracy values of all LMs were compared, the SpanBERT model was the most successful with 87.46%, while the worst model was ALBERT with 81.25%. This result shows that the TRP-QAS was able to detect the least correct answer for the ALBERT model.

8 CONCLUSION AND FUTURE WORK

Many models and methods have been developed for QA. Among these models are deep learning-based LMs. Retraining is required when LMs are to be analyzed with newly added data. Thus, time and cost become a problem. In this study, NLP-QAS and TRP-QAS extensions are proposed as extension for QAS, because of these problems. These extensions investigated the answerability of the questions by analyzing the questions that the LMs answered incorrectly. The NLP-QAS extension consists of two phases: sentence selection and answer detection. First, an analysis was performed on SQuAD with the proposed NLP-QAS. Many variations such as punkt, SDP and punkt and lemmatization were used in sentence selection. Among the variations, ”punkt with SDP” was the most successful method in sentence selection. Since RSS is a common step in both QAS, it is used for NLP-QAS and TRP-QAS. In the answer detection phase, RNP methods named RC, SPOS and SNER have been proposed. As a result of sentence selection applied according to QTP, RNP methods try to detect the answer correctly on the selected sentence.

First analysis with RNP was performed on the entire SQuAD. The most successful result for answer detection was obtained by applying lemmatization only for the sentences whose answer could not be detected, in addition to the correct answers by RNP. As a result of the analysis, the questions were answered with approximately 16.7% accuracy for Dev_set and 19.95% accuracy for Train_set.

In the next analysis of NLP-QAS, it has been shown that the questions can be answered by applying RNP methods together with BERT models. Questions regarding answers that could not be detected by the original BERT models were analyzed. As a result of the application of RSS and RNP methods, the questions answered incorrectly by the BERT models were determined correctly between 6.6% and 8.76% by NLP-QAS. When the success in all BERT models was analyzed, the accuracy of answer detection increased by approximately

1.1% to 2.4% with RNP methods. Thus, using an NLP-QAS extension for SQuAD has been shown to improve overall performance in BERT models for the QAS. In addition, the question pronouns of the questions answered incorrectly by the BERT models were analyzed. Among the question pronouns, the pronoun ”what” was answered incorrectly the most.

TRP-QAS, which is proposed as the second extension, is based on the extraction of triples from the related sentence or paragraph and the analysis of these triples according to question terms. Before the TRP-QAS analysis, the success of pre-trained BERT, ALBERT, ELECTRA, SpanBERT and RoBERTa LMs was measured, after which the success of these models was also analyzed for questions with an answer and no answer. As a result of this analysis, it was shown that all LMs answered the questions that have no answer more accurately. When the questions answered incorrectly by all LMs were examined, the questions containing the question pronoun ”what” were the most incorrectly answered. After this analysis, the analysis of TRP-QAS was applied. In the TRP-QAS, the RSS step is applied for the questions with an answer, while the paragraph is used for triple extraction for the questions that have no answer. The TRP-QAS extension was analyzed on SQuAD with BERT, ALBERT, ELECTRA, SpanBERT and RoBERTa LMs. Questions answered incorrectly by these LMs were extracted and saved for use in TRP-QAS. It has been shown that this QAS can answer questions that these LMs cannot answer correctly, and that the TRP-QAS can be used as an extension to LMs. When the success of the LMs was examined, the accuracy value increased between 3.3% and 7.5% with TRP-QAS. The most successful LM was SpanBERT with 87.46%.

Statistics were also extracted for the questions answered correctly by TRP-QAS extension. This QAS more successfully answered the questions that have no answer. In addition, it was investigated which question pronouns were answered correctly. LMs were not able to answer the a large number of questions, which included ”what” question pronoun. With the TRP-QAS,

many correct answers were given to the question pronoun ”what”. However, the percentage of other question pronouns was higher than ”what”.

Considering the all results, both QA extensions increased the accuracy values on the LMs. These QA extensions work independently of the dataset as they focus on questions that LMs have answered incorrectly. It also generates candidate answer using NLP methods without the need for any retraining process. So, increasing accuracy has been achieved without retraining. It can also be used in both extensions by applying weighting together with the LM output. In addition, both extensions can be used first for QA before exporting to the LM.

The training phase of LMs can take hours or days, depending on the hardware used for SQuAD. Therefore, either a good-featured computer or paid servers or platforms should be used during the training phase. The training phase takes hours and the cost is constantly increasing in use, even in these paid servers or platforms. When new data is added to these LMs, the model needs to be retrained with this data in order to answer correctly. Thus, a problem arises in terms of time and cost. With NLP-QAS and TRP-QAS extensions, questions can be answered correctly without the need for retraining these LMs. Because, regardless of the dataset, paragraph, question and answer are taken as input and according to the method in the QAS extension, this question can be answered in seconds or 2-3 minutes. Thus, both extensions provide great advantages in terms of cost and time.

In future works, it is aimed to investigate the effect of RNP methods on derivative LMs of BERT such as ALBERT, ROBERTa, ELECTRA and SpanBERT. In addition, as a result of the application of NLP techniques such as lemmatization and stemming to the original SQuAD, it is desired to measure the effect of NLP techniques by training LMs with this updated SQuAD.

For TRP-QAS, it is considered to be analyzed on different QA datasets. Afterwards, it is aimed to investigate the effect of using NLP-QAS and TRP-QAS extensions together on success for QA tasks. Thus, it is thought that success will be increased even more with two QA extensions that do not require retraining. Finally, it is aimed to apply this system to different languages by researching how to apply NLP-QAS and TRP-QAS on QA. For this, it is aimed to initially create a SQuAD for Turkish and to train LMs for this dataset. Thus, it is considered to create a benchmark platform for Turkish QA.

REFERENCES

Abadani, N., **Mozafari, J.**, **Fatemi, A.**, **Nematbakhsh, M.A.**, and **Kazemi, A.**, 2021, ParSQuAD: Machine Translated SQuAD dataset for Persian Question Answering, in 2021 7th International Conference on Web Research, ICWR 2021, ISBN 9781665404266, doi:10.1109/ICWR51868.2021.9443126.

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.**, and **Zheng, X.**, 2016, TensorFlow: A system for large-scale machine learning, in Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2016, ISBN 9781931971331.

Abdullah Alfaries, A., **Hamad Aljably, R.**, and **Saleh Al-Razgan, M.**, 2017, Modeling the NLP Research Domain using Ontologies: An Ontology Representation of NLP Concepts from a Research Perspective, in Future Technologies Conference.

Adhikari, A., **Ram, A.**, **Tang, R.**, and **Lin, J.**, 2019, DocBERT: BERT for document classification.

Al-Garadi, M.A., **Yang, Y.C.**, **Cai, H.**, **Ruan, Y.**, **O'Connor, K.**, **Graciela, G.H.**, **Perrone, J.**, and **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, ISSN 14726947, doi:10.1186/s12911-021-01394-0.

Allam, A. and **Haggag, M.**, 2012, The Question Answering Systems: A Survey., International Journal of Research and Reviews in Information Sciences (IJRRIS), ISSN 2046-6439.

Angeli, G., **Premkumar, M.J.**, and **Manning, C.D.**, 2015, Leveraging linguistic structure for open domain information extraction, in ACL-IJCNLP

2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference, ISBN 9781941643723, doi:10.3115/v1/p15-1034.

Ballı, S. and **Sağbaş, E.A.**, 2018, Diagnosis of transportation modes on mobile phone using logistic regression classification, IET Software, ISSN 17518806, doi:10.1049/iet-sen.2017.0035.

Beltagy, I., **Lo, K.**, and **Cohan, A.**, 2019, SCIBERT: A pretrained language model for scientific text, in 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, ISBN 9781950737901, doi:10.18653/v1/d19-1371.

Carleo, G., **Cirac, I.**, **Cranmer, K.**, **Daudet, L.**, **Schuld, M.**, **Tishby, N.**, **Vogt-Maranto, L.**, and **Zdeborová, L.**, 2019, Machine learning and the physical sciences, Reviews of Modern Physics, ISSN 15390756, doi:10.1103/RevModPhys.91.045002.

Carrino, C.P., **Costa-Jussà, M.R.**, and **Fonollosa, J.A.**, 2020, Automatic Spanish translation of the SQuAD dataset for multilingual question answering, in LREC 2020 - 12th International Conference on Language Resources and Evaluation, Conference Proceedings, ISBN 9791095546344.

Chakraborty, S., **Bisong, E.**, **Bhatt, S.**, **Wagner, T.**, **Elliott, R.**, and **Mosconi, F.**, 2021, BioMedBERT: A Pre-trained Biomedical Language Model for QA and IR, doi:10.18653/v1/2020.coling-main.59.

Chen, D. and **Yih, W.t.**, 2020, Open-Domain Question Answering, doi:10.18653/v1/2020.acl-tutorials.8.

Chernyavskiy, A., **Ilvovsky, D.**, and **Nakov, P.**, 2021, Transformers: "The End of History" for Natural Language Processing?, in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial

Intelligence and Lecture Notes in Bioinformatics), ISBN 9783030865221, ISSN 16113349, doi:10.1007/978-3-030-86523-8_41.

Chowdhary, K.R., 2020, Natural Language Processing, in Fundamentals of Artificial Intelligence, pages 603–649, Springer India, New Delhi, doi:10.1007/978-81-322-3972-7_19.

Clark, K., **Luong, M.T.**, **Brain, G.**, **Le Google Brain, Q.V.**, and **Manning, C.D.**, 2020, ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators, doi:10.48550/arxiv.2003.10555.

Decker, S., **van Harmelen, F.**, **Broekstra, J.**, **Erdmann, M.**, **Fensel, D.**, **Horrocks, I.**, **Klein, M.**, and **Melnik, S.**, 2000, The Semantic Web: The Roles of XML and RDF, IEEE Internet Computing.

Devlin, J., **Chang, M.W.**, **Lee, K.**, and **Toutanova, K.**, 2018, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, doi:10.48550/ARXIV.1810.04805.

Django, 2020, The web framework for perfectionists with deadlines | Django, `https://www.djangoproject.com/`.

Dong, L., **Mallinson, J.**, **Reddy, S.**, and **Lapata, M.**, 2017, Learning to paraphrase for question answering, in EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings, ISBN 9781945626838, doi:10.18653/v1/d17-1091.

Dwivedi, S.K. and **Singh, V.**, 2013, Research and Reviews in Question Answering System, Procedia Technology, ISSN 22120173, doi:10.1016/j.protcy.2013.12.378.

Esposito, M., **Damiano, E.**, **Minutolo, A.**, **De Pietro, G.**, and **Fujita, H.**, 2020, Hybrid query expansion using lexical resources and word embeddings for sentence retrieval in question answering, Information Sciences, ISSN 00200255, doi:10.1016/j.ins.2019.12.002.

Fellbaum, C., 2005, WordNet and wordnets, in Encyclopedia of Language and Linguistics.

Gardner, M., **Grus, J.**, **Neumann, M.**, **Tafjord, O.**, **Dasigi, P.**, **Liu, N.F.**, **Peters, M.**, **Schmitz, M.**, and **Zettlemoyer, L.**, 2019, AllenNLP: A Deep Semantic Natural Language Processing Platform, pages 1–6, doi: 10.18653/v1/w18-2501.

Giray, G. and **Ünalir, M.O.**, 2013, A method for ontology-based semantic relatedness measurement, Turkish Journal of Electrical Engineering and Computer Sciences, ISSN 13000632, doi:10.3906/elk-1108-68.

Gupta, P. and **Gupta, V.**, 2012, A Survey of Text Question Answering Techniques, International Journal of Computer Applications, doi:10.5120/8406-2030.

Gutierrez, C., **Hurtado, C.A.**, and **Vaisman, A.**, 2007, Introducing time into RDF, IEEE Transactions on Knowledge and Data Engineering, ISSN 10414347, doi:10.1109/TKDE.2007.34.

Guven, Z.A. and **Unalir, M.O.**, 2022, Natural language based analysis of SQuAD: An analytical approach for BERT, Expert Systems with Applications, ISSN 09574174, doi:10.1016/j.eswa.2022.116592.

He, J., **Zhao, L.**, **Yang, H.**, **Zhang, M.**, and **Li, W.**, 2020, HSI-BERT: Hyperspectral Image Classification Using the Bidirectional Encoder Representation from Transformers, IEEE Transactions on Geoscience and Remote Sensing, ISSN 15580644, doi:10.1109/TGRS.2019.2934760.

Hu, M., **Peng, Y.**, **Huang, Z.**, **Qiu, X.**, **Wei, F.**, and **Zhou, M.**, 2018, Reinforced Mnemonic reader for machine reading comprehension, in IJCAI International Joint Conference on Artificial Intelligence, ISBN 9780999241127, ISSN 10450823, doi:10.24963/ijcai.2018/570.

Hu, M., **Wei, F.**, **Peng, Y.**, **Huang, Z.**, **Yang, N.**, and **Li, D.**, 2019, Read + Verify: Machine reading comprehension with unanswerable questions,

in 33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, ISBN 9781577358091, ISSN 2159-5399, doi:10.1609/aaai.v33i01.33016529.

Huang, H.Y., **Zhu, C.**, **Shen, Y.**, and **Chen, W.**, 2018, FusionNet: Fusing via fully-aware attention with application to machine comprehension, in 6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings.

HuggingFace, 2021a, huggingface/neuralcoref: Fast Coreference Resolution in spaCy with Neural Networks, `https://github.com/huggingface/neuralcoref`.

HuggingFace, 2021b, Neural Coreference – Hugging Face, `https://huggingface.co/coref/`.

Jayanthi, K. and **Mahesh, C.**, 2018, A Study on machine learning methods and applications in genetics and genomics, International Journal of Engineering and Technology(UAE), ISSN 2227524X, doi:10.14419/ijet.v7i1.7.10653.

Jepsen, T.C., 2009, Just what Is an ontology, anyway?, IT Professional, ISSN 15209202, doi:10.1109/MITP.2009.105.

Jia, Y., **Shelhamer, E.**, **Donahue, J.**, **Karayev, S.**, **Long, J.**, **Girshick, R.**, **Guadarrama, S.**, and **Darrell, T.**, 2014, Caffe, doi:10.1145/2647868.2654889.

Joshi, M., **Chen, D.**, **Liu, Y.**, **Weld, D.S.**, **Zettlemoyer, L.**, and **Levy, O.**, 2020, Spanbert: Improving pre-training by representing and predicting spans, Transactions of the Association for Computational Linguistics, ISSN 2307387X, doi:10.1162/tacl_a_00300.

Khurana, D., **Koli, A.**, **Khatter, K.**, **Singh, S.**, and **Tools, M.**, 2022, Natural language processing: state of the art, current trends and challenges, Multimedia Tools and Applications 2022, pages 1–32, ISSN 1573-7721, doi: 10.1007/S11042-022-13428-4.

Kolomiyets, O. and **Moens, M.F.**, 2011, A survey on question answering technology from an information retrieval perspective, Information Sciences, ISSN 00200255, doi:10.1016/j.ins.2011.07.047.

Kulkarni, A., **Chong, D.**, and **Batarseh, F.A.**, 2020, 5 - Foundations of data imbalance and solutions for a data democracy, in F.A. Batarseh and R. Yang, editors, Data Democracy, pages 83–106, Academic Press, ISBN 978-0-12-818366-3, doi:https://doi.org/10.1016/B978-0-12-818366-3.00005-8.

Kwiatkowski, T., **Palomaki, J.**, **Redfield, O.**, **Collins, M.**, **Parikh, A.**, **Alberti, C.**, **Epstein, D.**, **Polosukhin, I.**, **Devlin, J.**, **Lee, K.**, **Toutanova, K.**, **Jones, L.**, **Kelcey, M.**, **Chang, M.W.**, **Dai, A.M.**, **Uszkoreit, J.**, **Le, Q.**, and **Petrov, S.**, 2019, Natural Questions: A Benchmark for Question Answering Research, Transactions of the Association for Computational Linguistics, doi:10.1162/tacl_a_00276.

Lan, Z., **Chen, M.**, **Goodman, S.**, **Gimpel, K.**, **Sharma, P.**, and **Soricut, R.**, 2019, ALBERT: A Lite BERT for Self-supervised Learning of Language Representations.

Li, M., **Chen, L.**, **Zhao, J.**, and **Li, Q.**, 2021, Sentiment analysis of Chinese stock reviews based on BERT model, Applied Intelligence, ISSN 15737497, doi:10.1007/s10489-020-02101-8.

Liu, R., **Wei, W.**, **Mao, W.**, and **Chikina, M.**, 2017, Phase Conductor on Multi-layered Attentions for Machine Comprehension, doi:10.48550/arxiv.1710.10504.

Liu, X., **Shen, Y.**, **Duh, K.**, and **Gao, J.**, 2018, Stochastic answer networks for machine reading comprehension, in ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers), ISBN 9781948087322, doi:10.18653/v1/p18-1157.

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

Ma, T., **Pan, Q.**, **Rong, H.**, **Qian, Y.**, **Tian, Y.**, and **Al-Nabhan, N.**, 2022, T-BERTSum: Topic-Aware Text Summarization Based on BERT, IEEE Transactions on Computational Social Systems, ISSN 2329924X, doi:10.1109/TCSS.2021.3088506.

Mahesh, B., 2020, Machine Learning Algorithms - A Review, International Journal of Science and Research (IJSR), 9, 381–386.

Manning, C., **Surdeanu, M.**, **Bauer, J.**, **Finkel, J.**, **Bethard, S.**, and **McClosky, D.**, 2015, The Stanford CoreNLP Natural Language Processing Toolkit, doi:10.3115/v1/p14-5010.

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

Martinez-Gil, J., **Freudenthaler, B.**, and **Tjoa, A.M.**, 2019, Multiple Choice Question Answering in the Legal Domain Using Reinforced Co-occurrence, in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), ISBN 9783030276140, ISSN 16113349, doi:10.1007/978-3-030-27615-7_10.

Medium.com, 2021, How the Embedding Layers in BERT Were Implemented, https://medium.com/@{_}init{_}/why-bert-has-3-embedding-layers-and-their-implementation-details-9c261108e28a.

Miller, G.A., 1998, WordNet: An electronic lexical database, MIT press.

Möller, Timo; Reina A.; Jayakumar, Raghavan; Pietsch, M., 2020, COVID-QA: A Question Answering Dataset for COVID-19, in In Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020, Association for Computational Linguistics.

MongoDB, 2020, MongoDB: The Developer Data Platform | MongoDB | MongoDB, `https://www.mongodb.com/`.

Noy, N.F. and **McGuinness, D.L.**, 2001, Ontology Development 101: A Guide to Creating Your First Ontology, Technical report, doi:10.1016/j.artmed.2004.01.014.

Otter, D.W., **Medina, J.R.**, and **Kalita, J.K.**, 2018, A survey of the usages of deep learning in natural language processing, doi:10.1109/tnnls.2020.2979670.

Pan, B., **Li, H.**, **Zhao, Z.**, **Cao, B.**, **Cai, D.**, and **He, X.**, 2017, MEMEN: Multi-layer Embedding with Memory Networks for Machine Comprehension, doi:10.48550/arxiv.1707.09098.

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.**, and **Chintala, S.**, 2019, PyTorch: An imperative style, high-performance deep learning library.

Peinelt, N., **Nguyen, D.**, and **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.**, and **Riedel, S.**, 2020, Language models as knowledge bases?, in 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, ISBN 9781950737901, doi:10.18653/v1/d19-1250.

Qu, C., **Yang, L.**, **Qiu, M.**, **Bruce Croft, W.**, **Zhang, Y.**, and **Iyyer, M.**, 2019, BERT with history answer embedding for conversational question answering, in SIGIR 2019 - Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, ISBN 9781450361729, doi:10.1145/3331184.3331341.

Rachiele, G., 2018, Tokenization and Parts of Speech(POS) Tagging in Python's NLTK library.

Rajpurkar, P., 2022, SQuAD - the Stanford Question Answering Dataset, `https://rajpurkar.github.io/SQuAD-explorer/explore/v2.0/dev/Normans.html`.

Rajpurkar, P., **Jia, R.**, and **Liang, P.**, 2018, Know what you don't know: Unanswerable questions for SQuAD, in ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers), ISBN 9781948087346, doi:10.18653/v1/p18-2124.

Rajpurkar, P., **Zhang, J.**, **Lopyrev, K.**, and **Liang, P.**, 2016, SQuad: 100,000+ questions for machine comprehension of text, in EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings, ISBN 9781945626258, doi:10.18653/v1/d16-1264.

Ram, O., **Kirstain, Y.**, **Berant, J.**, **Globerson, A.**, and **Levy, O.**, 2021, Few-shot question answering by pretraining span selection, in ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the Conference, ISBN 9781954085527, doi:10.18653/v1/2021.acl-long.239.

Reshamwala, A., Mishra, D., & Pawar, P., 2013, Review on natural language processing, IRACST Engineering Science and Technology: An International Journal (ESTIJ), 3(1), 113–116.

Salant, S. and **Berant, J.**, 2018, Contextualized Word representations for reading comprehension, in NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, ISBN 9781948087292, doi:10.18653/v1/n18-2088.

Singh, M., **Jakhar, A.K.**, and **Pandey, S.**, 2021, Sentiment analysis on the impact of coronavirus in social life using the BERT model, Social Network Analysis and Mining, ISSN 18695469, doi:10.1007/s13278-021-00737-z.

Spacy, 2021, Spacy 101: Everything you need to know · spacy usage documentation, `https://spacy.io/usage/spacy-101`.

Specifications, A., 2020, Annotation Specifications · spaCy API Documentation, `https://spacy.io/api/annotation`.

Su, D., **Xu, Y.**, **Winata, G.I.**, **Xu, P.**, **Kim, H.**, **Liu, Z.**, and **Fung, P.**, 2019, Generalizing question answering system with pre-trained language model fine-tuning, in MRQA@EMNLP 2019 - Proceedings of the 2nd Workshop on Machine Reading for Question Answering, ISBN 9781950737819, doi:10.18653/v1/d19-5827.

Sukthanker, R., **Poria, S.**, **Cambria, E.**, and **Thirunavukarasu, R.**, 2020, Anaphora and coreference resolution: A review, Information Fusion, ISSN 15662535, doi:10.1016/j.inffus.2020.01.010.

Sun, C., **Huang, L.**, and **Qiu, X.**, 2019, Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence, in NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, ISBN 9781950737130.

Sun, F., **Li, L.**, **Qiu, X.**, and **Liu, Y.**, 2018, U-Net: Machine Reading Comprehension with Unanswerable Questions, doi:10.48550/arxiv.1810.06638.

Torfi, A., **Shirvani, R.A.**, **Keneshloo, Y.**, **Tavaf, N.**, and **Fox, E.A.**, 2020, Natural Language Processing Advancements By Deep Learning: A Survey, NLP ADVANCEMENTS BY DEEP LEARNING, 1, doi:10.48550/arxiv.2003.01200.

Vaswani, A., **Shazeer, N.**, **Parmar, N.**, **Uszkoreit, J.**, **Jones, L.**, **Gomez, A.N.**, **Kaiser, Ł.**, and **Polosukhin, I.**, 2017, Attention is all you need, in Advances in Neural Information Processing Systems, ISSN 10495258.

Wang, C. and **Jiang, H.**, 2020, Explicit utilization of general knowledge in machine reading comprehension, in ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, ISBN 9781950737482, doi:10.18653/v1/p19-1219.

Wang, H., **Ma, C.**, and **Zhou, L.**, 2009, A brief review of machine learning and its application, in Proceedings - 2009 International Conference on Information Engineering and Computer Science, ICIECS 2009, ISBN 9781424449941, doi:10.1109/ICIECS.2009.5362936.

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

WordNet, 2021, WordNet Search - 3.1, `http://wordnetweb.princeton.edu/perl/webwn?s=car{&}sub=Search+WordNet{&}o2={&}o0=1{&}o8=1{&}o1=1{&}o7={&}o5={&}o9={&}o6={&}o3={&}o4={&}h=`.

Xiong, C., **Zhong, V.**, and **Socher, R.**, 2018, DCN+: Mixed objective and deep residual coattention for question answering, in 6th International

Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings.

Yamada, I., **Asai, A.**, **Shindo, H.**, **Takeda, H.**, and **Matsumoto, Y.**, 2020, LUKE: Deep contextualized entity representations with entity-aware self-attention, in EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, ISBN 9781952148606, doi:10.18653/v1/2020.emnlp-main.523.

Yang, W., **Xie, Y.**, **Lin, A.**, **Li, X.**, **Tan, L.**, **Xiong, K.**, **Li, M.**, and **Lin, J.**, 2019a, End-to-end open-domain question answering with BERTserini, in NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Demonstrations Session, ISBN 9781950737161, doi: 10.18653/v1/N19-4013.

Yang, Z., **Dai, Z.**, **Yang, Y.**, **Carbonell, J.**, **Salakhutdinov, R.**, and **Le, Q.V.**, 2019b, XLNet: Generalized autoregressive pretraining for language understanding, in Advances in Neural Information Processing Systems, ISSN 10495258.

Yasunaga, M., **Ren, H.**, **Bosselut, A.**, **Liang, P.**, and **Leskovec, J.**, 2021, QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering, doi:10.18653/v1/2021.naacl-main.45.

Yeh, Y.T. and **Chen, Y.N.**, 2020, QainfoMax: Learning robust question answering system by mutual information maximization, in 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, ISBN 9781950737901, doi:10.18653/v1/d19-1333.

Yoon, W., **Lee, J.**, **Kim, D.**, **Jeong, M.**, and **Kang, J.**, 2020, Pre-trained Language Model for Biomedical Question Answering, in Communications in

Computer and Information Science, ISBN 9783030438869, ISSN 18650937, doi:10.1007/978-3-030-43887-6_64.

Zhang, Z., **Wu, Y.**, **Zhao, H.**, **Li, Z.**, **Zhang, S.**, **Zhou, X.**, and **Zhou, X.**, 2019a, Semantics-aware BERT for language understanding, doi:10.1609/aaai.v34i05.6510.

Zhang, Z., **Wu, Y.**, **Zhou, J.**, **Duan, S.**, **Zhao, H.**, and **Wang, R.**, 2019b, SG-Net: Syntax-guided machine reading comprehension, doi:10.1609/aaai.v34i05.6511.

Zhang, Z., **Yang, J.**, and **Zhao, H.**, 2020, Retrospective Reader for Machine Reading Comprehension, 35th AAAI Conference on Artificial Intelligence, AAAI 2021, 16, 14506–14514, doi:10.48550/arxiv.2001.09694.

Zheng, J., **Chapman, W.W.**, **Crowley, R.S.**, and **Savova, G.K.**, 2011, Coreference resolution: A review of general methodologies and applications in the clinical domain, doi:10.1016/j.jbi.2011.08.006.

Zhou, X., **Hu, B.**, **Chen, Q.**, and **Wang, X.**, 2018, Recurrent convolutional neural network for answer selection in community question answering, Neurocomputing, ISSN 18728286, doi:10.1016/j.neucom.2016.07.082.

Zhu, H., **Tiwari, P.**, **Ghoneim, A.**, and **Hossain, M.S.**, 2022, A Collaborative AI-Enabled Pretrained Language Model for AIoT Domain Question Answering, IEEE Transactions on Industrial Informatics, ISSN 19410050, doi:10.1109/TII.2021.3097183.

APPENDIX

As a result of NLP-QAS analysis, sample questions answered with RNP methods for questions answered incorrectly by the BERT model and the applied procedures are shown in Table 8.1, Table 8.2 and Table 8.3 for each question pronoun.

Table 8.1: The RC method example for questions answered incorrectly by the 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 8.2: The SNER method example for questions answered incorrectly by the 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: NORP, Ireland: GPE, Pale: ORG, Trim Castle: PERSON, Dublin Castle: 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: DATE, French: NORP, Roman Catholic: NORP, Guyard de Moulin: 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 Answer Tag: How much = QUANTITY or MONEY Sentence NER tags:Victoria: GPE, the decade: DATE, 20 million ounces: QUANTITY, one third: 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: PERSON, Ögedei: GPE, Güyük: GPE, Great Khan: PERSON, 1251: 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 labeled as "chocolate". Answer Tag: Which year = DATE Sentence NER tags: 2003: DATE, Italy: GPE, Italian: 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 8.3: The SPOS method example for questions answered incorrectly by the 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'), ... 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'), ... Answer: northeast

As a result of TRP-QAS analysis statistics with question pronouns for questions that all LMs could not answer correctly are also given in detail in Table 8.4. This table shows that all LMs often incorrectly answer questions containing the question pronoun ”what”.

Table 8.4: Statistics of question pronouns for questions answered incorrectly by LMs

	Answer = N					**Answer = Y**				
Question pronouns	***BERT***	***ALBERT***	***ELECTRA***	***SpanBERT***	***RoBERTa***	***BERT***	***ALBERT***	***ELECTRA***	***SpanBERT***	***RoBERTa***
Who	158	67	118	112	79	159	171	126	95	167
When	100	43	76	80	44	105	162	83	88	132
What	778	339	593	633	319	1234	1584	1150	989	1418
Where	63	27	55	42	23	91	114	88	77	98
Why	15	7	14	15	5	58	65	53	50	68
How	150	60	116	127	63	218	264	185	185	247
Which	38	21	33	33	20	93	114	83	75	106
Others	6	1	12	11	1	34	42	34	31	40

ACKNOWLEDGMENTS

My sincere thanks go to my supervisor Prof. Dr. Murat Osman Ünalır for accepting me as a Ph.D. candidate and for his endless patience and support throughout this process. Many thanks also to Prof. Dr. Serdar Korukoğlu, Prof. Dr. Yalçın Çebi and Asst. Prof. Emine Sezer for their important contributions to the formation of this thesis. In addition, I would like to thank the Ege University Computer Engineering family for their support during my Ph.D. education and thesis studies.

As always, I would like to thank my very precious family, who supported me throughout my education life and helped me reach these days.

26/09/2022

Zekeriya Anıl GÜVEN

CURRICULUM VITAE

Personal Information

Name and Surname: Zekeriya Anıl Güven

Education

Ph.D.: Ege University, Department of Computer Engineering (2018-2022)

Master: Yildiz Technical University, Department of Computer Engineering (2015−2018)

Bachelor: Kocaeli University, Department of Computer Engineering (2010−2015)

Academic Experiences

Research Assistant, Ege University, Department of Computer Engineering (2018−2022)

Visiting Researcher, Aalborg University, Department of Computer Engineering (10.2021-12.2021)

Research Assistant, Recep Tayyip Erdoğan University, Department of Computer Engineering (2017−2018)

Job Experiences

Finnet Elektronik Yayıncılık Data İletişim Ltd.Şti., Software Specialist, Istanbul (08.2015−03.2016)

Terapi Yazılım, Software Specialist, Bursa (09.2016−02.2017)

Etiya, Software Specialist, Istanbul (02.2017−04.2017)

Articles & Papers

SCI or SCI Expanded, SSCI, AHCI journal:

1. Guven, Z. A., & Unalir, M. O. (2022). Natural language based analysis of SQuAD: An analytical approach for BERT. Expert Systems with Applications, 195, 116592. DOI:https://doi.org/10.1016/j.eswa.2022.116592

2. Guven, Z. A. (2022). The Comparison of Language Models with a Novel Text Filtering Approach for Turkish Sentiment Analysis. Transactions on Asian and Low-Resource Language Information Processing. DOI:https://dl.acm.org/doi/10.1145/3557892

3. Güven, Z., Diri, B., & Çakaloğlu, T. (2022).Impact of n-stage Latent Dirichlet Allocation to the Analysis of Headlines Classification. Computer Science-AGH Journal (Just accepted)

4. Güven, Z., Diri, B., & Çakaloğlu, 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, 35(4). DOI:https://doi.org/10.17341/gazimmfd.556104

Other Indexed Journal:

1. Güven, Z. A., Banu, DİRİ., & ÇAKALOĞLU, T. (2019). Emotion Detection with n-stage Latent Dirichlet Allocation for Turkish Tweets. Academic Platform-Journal of Engineering and Science, 7(3), 467-472. DOI:https://doi.org/10.21541/apjes.459447

2. Guven, Z. A., Diri, B., & Cakaloglu, T. (2021). n-stage Latent Dirichlet Allocation: A Novel Approach for LDA. arXiv preprint arXiv:2110.08591.

3. Guven, Z. A., Diri, B., & Cakaloglu, T. (2021). Evaluation of Non-Negative Matrix Factorization and n-stage Latent Dirichlet Allocation for Emotion Analysis in Turkish Tweets. arXiv preprint arXiv:2110.00418.

Papers:

1. Guven, Z. A. (2021, September). The Effect of BERT, ELECTRA and ALBERT Language Models on Sentiment Analysis for Turkish Product Reviews. In 2021 6th International Conference on Computer Science and Engineering (UBMK) (pp. 629-632). IEEE. DOI:https://doi.org/10.1109/UBMK52708.2021.9559007

2. Guven, Z. A. (2021, September). Comparison of BERT models and machine learning methods for sentiment analysis on Turkish tweets. In 2021 6th International Conference on Computer Science and Engineering (UBMK) (pp. 98-101). IEEE. DOI:https://doi.org/10.1109/UBMK52708.2021.9559014

3. Guven, Z. A., & Unalir, M. O. (2021, September). Improving the BERT Model with Proposed Named Entity Recognition Method for Question Answering. In 2021 6th International Conference on Computer Science and Engineering (UBMK) (pp. 204-208). IEEE. DOI:https://doi.org/10.1109/UBMK52708.2021.9558992

4. Güven, Z. A., Diri, B., & Çakaloğlu, T. (2019, September). Comparison of topic modeling methods for type detection of Turkish news. In 2019 4th International Conference on Computer Science and Engineering (UBMK) (pp. 150-154). IEEE. DOI:https://doi.org/10.1109/UBMK.2019.8907050

5. Güven, Z. A., Diri, B., & Çakaloğlu, T. (2019). Comparison Method for Emotion Detection of Twitter Users. In 2019 Innovations in Intelligent Systems and Applications Conference (ASYU) (pp. 1-5). IEEE. DOI:https://doi.org/10.1109/ASYU48272.2019.8946435

6. Guven, Z. A., Diri, B., & Çakaloglu, T. (2018, October). Classification of New Titles by Two Stage Latent Dirichlet Allocation. In 2018 Innovations in Intelligent Systems and Applications Conference (ASYU) (pp. 1-5). Ieee. DOI:https://doi.org/10.1109/ASYU.2018.8554027

7. Güven, Z. A., Diri, B., & Çakaloğlu, T. (2018, April). Classification of TurkishTweet emotions by n-stage Latent Dirichlet Allocation. In 2018 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT) (pp. 1-4). IEEE. DOI:https://doi.org/10.1109/EBBT.2018.8391454

Review Tasks

Expert Systems with Applications, SCI, 3 different journal

ACM Transactions on Asian and Low-Resource Language Information Processing, SCI, 3 different journal

Computer Science-AGH, ESCI, 3 different journal

Concurrency and Computation: Practice and Experience, SCI, 2 different journal

Turkish Journal of Electrical Engineering & Computer Sciences, SCI, 2 different journal

Engineering Applications of Artificial Intelligence, SCI, 2 different journal

Information Processing & Management, SCI, 1 journal

Egyptian Informatics Journal, SCI, 1 journal

Soft Computing, SCI, 1 journal

Journal of Artificial Intelligence and Data Science, Other Indexed Journal, 1 journal

Thesis

Master Degree: N-seviyeli gizli Dirichlet ayırımı desteği ile tür ve duygu sınıflandırma (Genre and emotion classification by support of N-stage latent Dirichlet allocation)