PARAGRAPH AND SENTENCE LEVEL SEMANTIC TEXTUAL SIMILARITY MEASUREMENT TECHNIQUES: AN APPLICATION ON SOLVING OSYM EXAM QUESTIONS

ONUR AÇIKGÖZ
B.S., Computer Engineering, IŞIK UNIVERSITY, 2015

Submitted to the Graduate School of Science and Engineering
in partial fulfillment of the requirements for the degree of
Master of Science
in
Computer Engineering

IŞIK UNIVERSITY
2019

IŞIK UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

PARAGRAPH AND SENTENCE LEVEL SEMANTIC TEXTUAL SIMILARITY MEASUREMENT TECHNUQUES: AN APPLICATION ON SOLVING OSYM EXAM QUESTIONS

ONUR AÇIKGÖZ

APPROVED BY:

Prof. Dr. Olcay Taner YILDIZ (Thesis Supervisor)	Işık University
Assist. Prof. İlknur KARADENİZ EROL	Işık University
Assoc. Prof. Arzucan ÖZGÜR	Boğaziçi University

APPROVAL DATE: 06/09/2019

PARAGRAPH AND SENTENCE LEVEL SEMANTIC TEXTUAL SIMILARITY MEASUREMENT TECHNIQUES: AN APPLICATION ON SOLVING OSYM EXAM QUESTIONS

Abstract

An Application on Solving ÖSYM Exam Questions Semantic textual similarity is a well-known natural language processing (NLP) task which aims to measure the degree of similarity of two texts in terms of meanings. In this thesis, our goal is to investigate best semantic textual similarity measurement modeling techniques for the Turkish language at paragraph-to-sentence and sentence-to-sentence levels. Our plan is to exploit morphological knowledge of the Turkish language as a prior input, by using morphological disambiguation toolkit of our study group which automatically annotates morphological tags of words (word, syllable, roots, etc.) in morpheme-level while disambiguating possible parse-trees at the sentence-level.

As an application, we proposed statistical models challenging to solve two special types of official ÖSYM multiple-choice exam questions, which examine comprehension ability of students on textual meanings at sentence-to-sentence and paragraph-to-sentence levels. We constructed a question dataset for evaluation that covers official ÖSYM exams with varying degrees of difficulties such as ÖYS, ÖSS, DGS, TEOG, SBS, etc.

Keywords: Keywords: Similarity, NLP

PARAGRAF VE CÜMLE DÜZEYİNDE ANLAMSAL METİNSEL BENZERLİK ÖLÇME TEKNİKLERİ: OSYM SINAV SORULARI ÇÖZEN UYGULAMA

Özet

İBu uygulamanın amaçı iki metin arasındaki benzerlik derecesini bularak ÖSYM de cıkmıs soruların anlamsal benzerliklerini bulup cevaplandırmaktır. Bu tez çalısmamızda, Türk dili için en iyi anlamsal metinsel benzerlik ülçüm modelleme tekniklerini paragraftan cümleye ve cümle cümle seviyesinde incelemektir. Planımız Türkçenın morfolojik bilgisini bir girdi olarak kullanmaktır, morfolojik belirsizlik giderme araçlarını kullanarak otomatik olarak kelimelerin morfolojik etkilerini ekleyen (kelime, hece, kök vb.) morfem düzeyınde olası ayrıstırma ağaçlarını belisizleştiren çalısma grubumuz olusturulmustur.

ÖYS, ÖSS, DGS, TEOG, SBS gibi cesitli zorluk derecelerini kapsayan değerlendirme için bir soru veri seti olusturduk. Öğrencilerin cümle-cümle ve paragraph-cümle seviyelerinde metin anlama kabiliyetleri inceleyen iki özel soru tipi üzerinde ÖSYM coktan secmeli soru tipi istatistiksel modeler öğrendik.

Anahtar kelimeler: Büyük Veri, DDİ,Özetleme

Acknowledgements

I am grateful to the experts who made personal and educational contributions to me during my undergraduate and graduate studies.Thank you for sharing their knowledge and experience with me as much as possible.

I would like to thank my esteemed professor OLCAY TANER YILDIZ, who has been standing by me since the beginning of preparing her graduate thesis and continues to help me patiently in my mistakes.

I am thankful and fortunate enough to get constant encouragement, support and guidance from Ali Tunca Gürkan and Kübra Açıkgöz.

To my family…

Table of Contents

List of Tables

List of Figures

List of Abbreviations

NLP	**N**atural **L**anguage **P**rocessing
ÖYS	**Ö**ğrenci **Y**erleştirme **S**ınavı
ÖSS	**Ö**ğrenci **S**eçme **S**ınavı
ÖSYM	**Ö**ğrenci **S**eçme **Y**erleştirme **M**erkezi
DGS	**D**ikey **G**eçiş **S**ınavı
SBS	**S**eviye **B**elirleme **S**ınavı
TEOG	**T**emel **E**ğitimden **O**rtaöğretime **G**eçiş
TOEFL	**T**est of **E**nglish as a **F**oreign **L**anguage
ALES	**A**kademik **P**ersonel ve **L**isansüstü **E**ğitimi **G**iriş **S**ınavı
LES	**L**isansüstü **E**ğitim **G**iriş **S**ınavı
LYS	**L**isans **Y**erleştirme **S**ınavı
YGS	**Y**ükseköğretime **G**eçiş **S**ınavı
YÖS	**Y**abancı **U**yruklu **Ö**ğrenci Sınavı
OCT	**O**ptical **C**haracter **I**dentification
GLD	**G**enerally **L**evenshtein **D**istance
XML	**E**xtensible **M**arkup **L**anguage
MSR	**M**achine **S**tress **R**ating
OVL	**O**verlay **F**ile
DDM	**D**ividend **D**iscount **M**odel
IMDM	**I**nternet **M**ovie **D**atabase
CBOW	**C**ommon **B**ag **O**f **W**ords
AESA	**A**ssociation **O**f **E**ducational **S**ervice **A**gencies
SNS	**S**ocial **N**etworking **S**ervice
RNN	**R**ecurrent **N**eural **N**etwork

LTSM	**L**earning and **L**eaching **S**upport **M**aterial
SDAE	**S**pecial Direct Admissions Exercise

Chapter 1

Introduction

In natural language processing applications, it is necessary to calculate the similarities between short texts. These short texts can be like sentences. This is a necessary entailment lately. Folksy, the technique of determining resemblance between documents has been developed based on the analysis of shared words. Often these methods are enough to deal with long documents because we need to consider the number of words that appear to identify text similarities. Word similarity in short sentences may be rare or not at all. This is because people want to use different sentences to explain similar meanings. In short documents such surface information is limited. Therefore, this problem has made it difficult to perform calculation methods. The main part of this study is to calculate the similarity, especially between short texts. Sentence similarities are used in many interesting applications.

This application, which was prepared to solve the ÖSYM exam questions, was aimed to measure the semantic similarity between the two texts. In other words, it was aimed to see the degree of similarity between the two texts. Natural language processing (NLP) was used to achieve these objectives. In this thesis, the best semantic measurement text models were determined first. These were measured from paragraph to sentence to sentence to sentence levels. In this thesis, the morphological factors of the words (speech, time, root, etc.) were used as a priority input by using the morphological uncertainty removal tools of our study group to identify the possible decomposition tools to sentence level. In this application, it is necessary to examine the sentence and sentence comprehension skills of the students. For this, compelling statistical models were proposed to measure the multiple choice exam questions. A question data set was created in exams such as ÖYS, ÖSS, DGS, TEOG, SBS.

An Application on Solving ÖSYM Exam Questions Semantic textual similarity is a well-known natural language processing (NLP) task which aims to measure the degree of similarity of two texts in terms of meanings. We developed an application which have similarity methods. We calculated similarity scores and, we wrote the percentage of accuracy in the results. We discussed the purpose of the study in Chapter 1. We NLP techniques literature in Chapter 2. We show how to prepare data of work Chapter 3, similarities in Chapter 4 and give experiments Chapter 5 and given in conclusion Chapter 6.

Chapter 2

Literature Review

In this chapter, we tackle some fundamentals of the big data infrastructure, how to relate big data with NLP and explain summarization techniques.

Non-supervised methods for distributed learning represent NLP research everywhere today. However, little is known about the best ways to learn a distributed sentence or sentence representations from unlabeled data [1]. This article is a systematic comparison of such models. In this article, we determined that the optimal approach is critically dependent on the intended application. More complex models are preferred for demonstrations to be used in supervised systems. Shallow log-bilinear models give the best results to create areas represented by distance metrics. It has been determined that there may be two new unsupervised representation-learning designed to increase the balance between training time, domain portability and performance. Distributed representations real value vectors encoding the meanings of linguistic units are used in NLP research.

For single words or word-like entities, there are ways of achieving such representations from naturally occurring (unlabeled) training data that are based on relatively agnostic targets (such as predicting adjacent words). These methods are empirically understood. Word representation areas show consistent aspects of human conceptual organization. It can be added as a feature to contribute to the language processing system. There is little agreement between methods of learning the dispersed representations of sentences or sentences. With the use of deep language processing techniques, it has become common for models to start using sentences as a valuable vector. Although it is not official, it is seen that internal sentence representations can show semantic intuitions in the best way. The language processing system is used to find out which architectures and targets provide the best impression. In this article, a systematic comparison of methods to learn the representations of sentences is made.

They use unlabeled methods for use in training models because these methods are low in cost and are suitable for use between languages. In this article, it was determined that there were differences in the lake due to the nature of the evaluation criterion. More detailed and mixed models need more resources to train and more time is spent. Except for shallow loboid models, they perform best in controlled environments. Although SkipThought Vectors show the best performance in most audited evaluations, SDAEs show their best performance for password definition.

In comparison to the SICK sentence, FastSent performs better than any other model. Among the audited and uncontrolled criteria, the most powerful performance is the bag-of-words model, which is trained to create word placement by using dictionary definitions. When examined together, these are important and guiding in the representation and learning of sentences or sentences in the grammar comprehension system. In this article, it has been found that many architectures are used by NLP researchers to learn deep learning algorithms, software advances, hardware data, and distributed sentence representations. We have systematically compared the first of these methods (our knowledge). In the performance of the approaches we have obtained from this comparison, it was concluded that there are differences that cannot be ignored in various evaluations. It was found that the most appropriate approach was critically linked to whether or not the representations would be applied in critical or uncontrolled environments. FastSent and Sequential Denoising have proposed two new alternatives to Autoencoders. DictRep is the best choice for those who don't. This article discusses models using training data. It is necessary to use what is necessary to give a human touch in the language technology that is needed to use future goals as an alternative method to train supervised architectures in multiple audit tasks.

One of the recent greatest achievements of uncontrolled word burials in a large number of applications has raised a problem that similar methods can be obtained to systematically improve the word series. In this article, they aim to obtain a simple and effective and also uncontrolled method to educate distributed representations of sentences [2]. With this method, the robustness of the evaluation sentence was taken into consideration and the latest technology used in most of the reference tasks left behind unsupervised models. Developing unsupervised learning, using it for the use of machine learning methods, is of great importance to be used as educational resources and to unlock access to an unlimited amount of data. Exception and cautionary text come in the form of word inscriptions that are trained in an unsupervised manner from the natural language processing area.

These word presentations based on the Matrix factoring model are trained according to the raw text data and have been made available from anywhere. It is difficult to use semantic placements to produce longer pieces of text, such as sentences, paragraphs, or whole documents, even if the words are understandable and usable. Teaching these general-purpose screenings in an unsupervised manner will continue to be an important goal. Nowadays, two contradictory research tendencies have emerged. The strong trend for in-depth learning for NLP has led to the development of more powerful complex models such as RNN, LSTM and even Neural Turing. It is more powerful than others, and at the same time increasing model complexity makes these models run slower on large data sets.

Simple shallow models such as matrix factorizations may benefit from working on very large data sets, which are more advantageous in uncontrolled environments.

LTSMs have been shown to perform better in order to obtain flat centring, with average word vectors on average, to generate sentence placements. This sample demonstrates the potential to use text-to-simplicity from tradeoff using model complexity with scalable algorithms about the ability to process text.

As a result of this change, the unsupervised sentence is better communicated. The proposed model can be assumed as another branch of CBOW to train the sentence instead of placing the word. They conclude that the resultant sentence is an action that exceeds the complexity of education and inference, while maintaining the simplicity of art and educational complexity. In this article, a new method that can be used as an uncontrolled and computationally effective method to educate and understand sentence placements is revealed. This method was found to perform better than all other methods except SkipThought vectors. In addition, SkipThought vectors are extremely weak in sentence similarity method. The model found in this article is generally the most ideal for such determinations. In the future, efforts can be made to increase the model to take advantage of ordered sentence data. In later periods, they planned to conduct studies to investigate the capability of models such as pre-trained burials to enable watershed transfer learning tasks.

In machine learning, inputs must be used as a fixed-length vector. When it comes to text, one of its fixed dimensions is bag-of-word. Although it is very popular, the bag-of words feature has two basic weaknesses. These are words lose their order and ignore the meaning of words. In the algorithm in this article, the algorithm uses each document-trained vectors to estimate the words in the document [3]. In this algorithm, bag-to-bag models come from the weaknesses of the weaknesses.

According to the empirical results, it is seen that it performs better than other techniques for text presentation as well as word expression models. Finally, with this algorithm, they achieved the best results in various text classification and sensitivity analysis tasks. Text classification and clustering play an important role in many applications such as document retrieval, web search, spam filtering. In these applications, there are the algorithms of reaction or machine learning algorithms such as Kmeans. These algorithms must represent text inputs as a fixed-length vector. The most commonly used fixed-length vector model for texts is the n-gram bag. This is due to its simplicity, efficiency and surprisingly correct labels. The sequence word is lost, so different sentences will have the same meaning as long as they are used with the same words.

Although the grammatical word appears in a short context, there are problems such as lack of data and high size. The bag of words gives little information about the division of words. They give information about distances between words. In this article, we have information about Paragraph Vector. Texts are varied and have different lengths. Paragraph A vector name is used to emphasize that a sentence in different lengths of text can be applied to a large document. The model vector representation is designed to be useful in predicting words in a paragraph. In other words, it combines the vector of the paragraph with a few words from a paragraph and tries to guess the following word in this way. Vocabulary vectors and paragraph vectors are trained with static gradient descent and backpropagation. The word vectors are shared between them, even though paragraph vectors and paragraphs are not the same.

At the time of the estimation, paragraph vectors use the vocabulary vectors to extract the new paragraph vector according to the most accurate. This method is trying to identify vector representations of words using neural networks. Each word in the methot is combined with other vocabulary vectors in a context, or the resulting vector is used to determine other words in the context.

As a result, after the model has been trained, the vectors match the similar vector representation of semantically similar words. The researchers then attempted to move the models further than the word level to obtain sentence-level or sentence-level representations. Another approach is to attempt to combine the word vectors using the matrix tree method in the order given by the discrete tree of a sentence. The first approach, which focuses on the weighted average of word vectors, loses word order as in word sequence models. Because the method of parsing is based solely on research, it only benefits sentences. The paragraph vector represents the representation of the input strings of different length. Unlike previous methods, the text of any length can be applied in general. No task-specific adjustment is required for the weight word and is not dependent on the parsing trees.

In this article, Paragraph Vector is defined as an uncontrolled learning algorithm that learns the vector representation of variable length text pieces such as sentences and documents. Stanford Treebank and IMDB sensitivity analysis datasets, it has been determined that this method can be a competitor to the best methods. The benefits of the vector have shown good performance in capturing the meaning of the paragraph, and we conclude that this method performs well. The focus of this work is to represent texts. But we can apply this method to learn impressions for sequential data. In areas where it is not possible to parse, ie in non-text areas, Paragraph Vector is expected to be used as an alternative to bags of words and grams.

The normalized arrangement of the array offered so far show good performances in some applications. None is acceptable in finding the actual metric result between the string. Because the triangle has inequality [4]. The X and Y strings on a finite alphabet are discussed. The correction distance between X and Y is [X] and [Y]. Levenshtein Distance O([X]. [Y]) calculated the distance using the GLD with complexity. If the weight function is a metric, it is a metric of [0,1]. All add-on removal costs are the same if considered based on basic editing procedures. Similar results are obtained from new distances shown by recognizing handwriting figures in the experiments used in the AESA algorithm. If inequality is violated in a given data set and triangle may perform better. Quantifying is used to understand the similarity between striking character strings. Because the key information can be expressed by symbolic sequences such as text import, signal processing, and biological computation. Although the appropriate distance criteria, however, the best practice was proposed and discussed to find the most appropriate one among them. Generally, the generalized Levenshtein Distance (GLD) is the most suitable one to compare strings with various editing operations. This measure is commonly referred to as editing. A string can be defined as the cost of other operations distance and minimum return along the weighted edit sequence.

The comparison of two incorrect short sequences is more critical than the comparison of long strings. Therefore, in some cases, we use it to normalize GLD. In some cases, we use triangular inequality. Although both offer good performance, in a few practical cases from a theoretical point of view, GLD cannot perform any triangular inequality. On the other hand, normalized metrics for the hand, symmetric set difference, and euclidean metrics are not valid for correcting the distance. Criteria based on Lempel-Zive are complex. A normalized regulation distance that can be considered as the original defined so far remains a problem with the metric that has not been solved between the two arrays. This communication provides a solution to define such a metric as a simple function of string lengths and GLD.

In this paper, the newly normalized editing distance is presented. Levenshtein as a simple function of string lengths and generalized. The main contribution of the article is to use AESA with handwriting, and its alignment to two normalized editing distances is like a degree of better results if the triangle inequality is violated at a certain level. Be metric because there is no other normalized distance. This study is important in this respect. They are planning to use the new distance between the future uses. Like phylogenetic tree making.

With some new applications, sentence similarity measurements have come to the fore. New methods have been created to calculate sentence similarity [5]. In this article, the length of the sentence focuses on the calculation of the similarity between very short texts. An algorithm that takes into account the meaning and structure of the sentence is determined. When calculating the semantic similarity of the two sentences, information from a structured database and corpus statistics is used. The inclusion of sensory information and corpus statistics helps to adapt the method to different areas. Recently, natural language processing applications require an effective method to calculate the similarity between short texts or sentences. Employment of sentence similarity can greatly simplify the knowledge base of the tool using natural sentences.

Sentence similarity is used in some applications on the internet. It shows that the computation of sentence similarity in web page acquisition and text mining has become a general component for the research community related to text presentation and discovery. Computation of sentence similarity seems to be an increasing demand. However, current measures have disadvantages in sentence calculation. Further work will include the construction of a more varied sentence pair dataset.

Social network service contains too many data. This data is analyzed at the sentence level [6]. This article describes the system presented for the task of SemEval2015 semantic textual similarity. The Internet contains a lot of information used for different types of purposes. Especially there is a lot of information on sites that provide social networking services. Vector space model used for natural language processing. This model creates vectors on the basis of frequency of word appearance and co-occurring words. In short texts, the word co-occurrence is scarcely any. The vector space model is not the best option because the average SNS contains mostly short sentences. In the system presented to SemEval2015 mentioned at the beginning, the sentence order is calculated considering the semantic distance between words. If the sentence similarity is to be calculated, the editing distance is used. The word view is also mentioned in context. This article was written to propose a method to measure sentence similarity. Briefly, we have adopted the semantic distance of the word at the editing distance and the word appears within the context. The results of the evaluation show that the use of the word view in the context is an effective element to determine the similarity of the sentence.

To be able to read, write and understand the primitive language of the students' determination of interpretation skills, educational system as a method of primary measurement and selection at almost every stage it is used. Multiple-choice questions are used to determine the level of students in the education system in our country. In this study, we offer 540 data sets with related data extrinsic metering data set was created. In addition, we targeted the data set that we created; These questions of the students who take the exams in difficulty levels we wanted to increase the possibility of comparing models with solving achievements.

All the questions that make up the dataset Republic of Turkey Measurement, Selection and Placement Center (ÖSYM) official exams questions. Two main semantic problems targeted for 2 different question types have been determined. From various sources questions in Xlsx file format. The data set is terminated by converting it to XML format within the structural template that we have specified. We aimed to answer questions automatically. Therefore the Turkish language for vocabulary and sub-word morphology we could hear. We have all the questions and answers sentences morphological analysis and morphological disambiguation. As a result of the study, the XML file contains the morphological analysis results.

Chapter 3

Data

Official multiple choice questions of Turkish tests which are prepared by the Republic of Turkey Student Selection and Placement Center are used as inputs. Two types of questions are selected for two target problems: (i) finding the semantically closest sentence to a given paragraph, (ii) finding the semantically closest sentence for a given sentence. After the data gathering, preparing, morphological tagging, and format conversion phases we end up with 540 questions of a final dataset [7].

The main purpose of natural language processing (NLP) research understanding of natural language, which is a highly complex phenomenon, interpretation, classification, summarizing needing high cognitive abilities partially or completely with computers to be supported. Two important cognitive skills determination of measurement discourse and finding similar elements problems are frequently studied in DDM studies have been tasked.

To be able to read, write and understand the primitive language of the students' determination of interpretation skills, educational system as a method of primary measurement and selection at almost every stage it is used. Multiple-choice questions are used to determine the level of students in the education system in our country. In this study, we offer 540 data sets with related data extrinsic metering data set was created. In addition, we targeted the data set that we created;

These questions of the students who take the exams in difficulty levels we wanted to increase the possibility of comparing models with solving achievements.

All the questions that make up the dataset Republic of Turkey Measurement, Selection and Placement Center (ÖSYM) official exams questions. Two main semantic problems targeted for 2 different question types have been determined. From various sources questions in Xlsx file format. The data set is terminated by converting it to XML format within the structural template that we have specified. We aimed to answer questions automatically. Therefore the Turkish language for vocabulary and sub-word morphology we could hear. We have all the questions and answers sentences morphological analysis and morphological disambiguation. As a result of the study, the Xml file contains the morphological analysis results. In FIGURE 1, The planned stages are outlined.

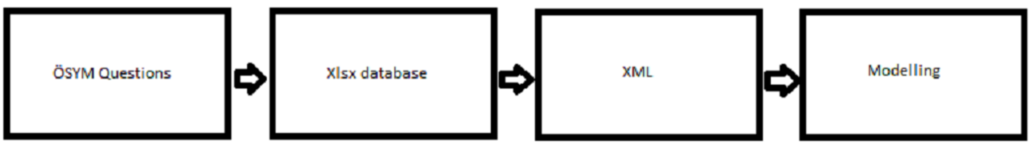


Figure 3.1: Completed and planned stages of study.

3.1 Measurement with Multiple Choice Questions

The use of multiple choice question datasets as a measurement method is not new. MSR study sentence-blank-dropped word-estimated type five It consists of 1040 questions multiple choice data set. The MSR data set is used as an important supportive measurement method in the external measurement of many continuous and discrete language models and word similarity models. [8, 9]. Real of MSR data set measurement based on human evaluations, the deficiencies of basic (intrinsic) statistical measurement methods such as perplexity. In a similar study, Landauer and his colleagues identified 80 questions of the word test-Turkish as a Foreign Language as the data set [10]. The TOEFL dataset has been used as a comparison tool by researchers for synonymous word-finding models since 1997 [11] . Bullinaria and his colleagues (2012) reported that they can produce models that can solve TOEFL questions in 100 % [10].

In another case study modeled using the NLP and reasoning, it was found that the students were able to achieve average student performance in solving mathematical questions asked in exams for admission to the University of Tokyo reported [12]. The researchers interpreted their findings not as a success of machine learning, but as a result of the lack of creativity in the education system.

3.2 Data Collection

ALES, DGS, LES, LYS, OSS, ÖSS, ÖYS, SBS, TEOG, YGS, YÖS exams are the exams at different levels of difficulty organized by OSYM. The common point of the ÖSYM multiple-choice exams is that they consist of repetitive Turkish question types. Questions that measure information such as literature and grammar are excluded from the study. As a result of our preliminary research 2 of the 32 types of questions were included in the data set. The data set is limited to the quiz questions. The exam questions were obtained from the documents shared by ÖSYM on the website as well as from various websites and the question books. For example, the question book[8], which includes paragraph questions and solutions of the last 52 years, is one of the sources of the study.

3.3 Data Preparation

At this stage, question type structures are prepared in the xlsx file format. In addition, ÖSYM exam types, as well as information about the years of examinations, are marked. The resulting data pool was scanned quickly to determine which question is proportional to which question type and the question of how many questions can be reached from which type of question has been made. The four types of question types that target the two main problems identified in light of the estimated data are described below.

3.4 Question Types

In the first question type, it is desirable to find out which of the following sentences is similar to what a statement within the paragraph or paragraph is meant to be said. This type of question can be asked in different expressions as follows:

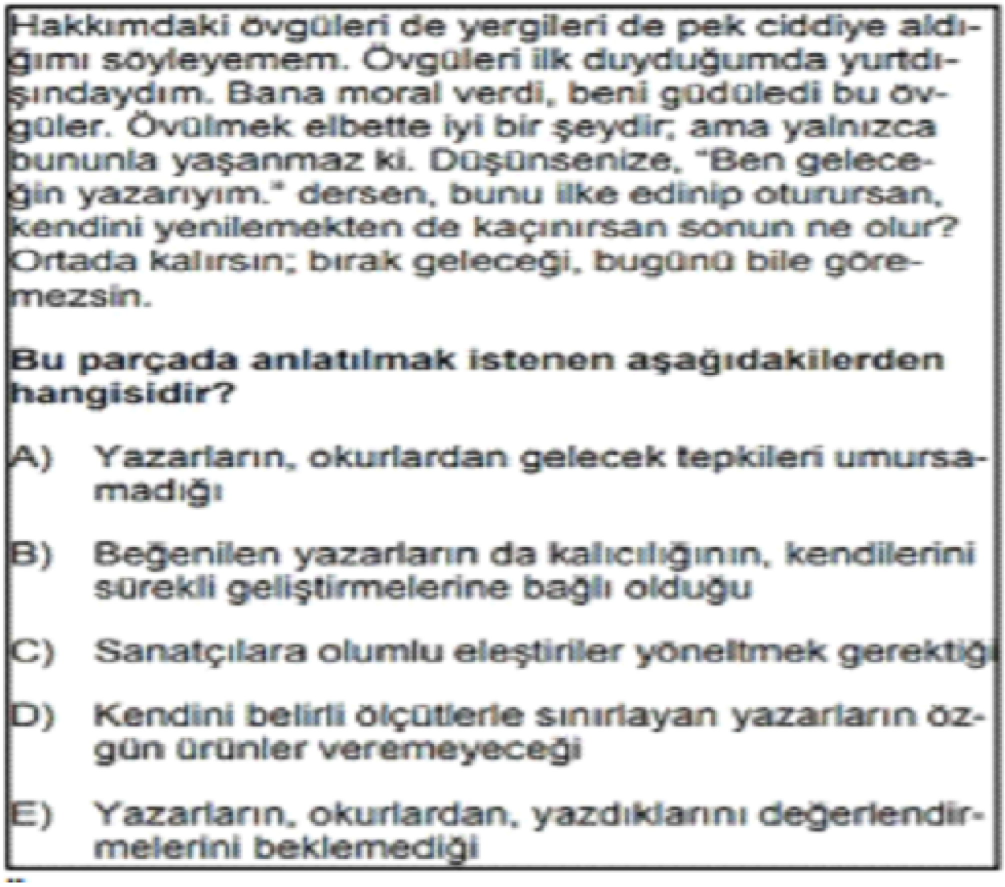
- ...what is meant to be?
- ...what is the subject of the text?
- ...which one can be said / not?
- ...What is the main idea?
- ...is addressed / is not addressed?

  The second question type is aimed at finding sentences that are closest to the sentence.

  Question type below different formats:

- ... is closest to the meaning?
- . ... is not exactly compatible with each other?
- ...find the judiciary that can be removed?
- ... which one has the same meaning as the given sentence?
- What is meant by ...?

Examples of question type 1 and question type 2 are given in Figure 3.2 and Figure 3.3, respectively. For question type 1, paragraph-sentence and sentence



Hakkımdaki övgüleri de yergileri de pek ciddiye aldığımı söyleyemem. Övgüleri ilk duyduğumda yurtdışındaydım. Bana moral verdi, beni güdüledi bu övgüler. Övülmek elbette iyi bir şeydir; ama yalnızca bununla yaşanmaz ki. Düşünsenize, "Ben geleceğin yazarıyım." dersen, bunu ilke edinip oturursan, kendini yenilemekten de kaçınırsan sonun ne olur? Ortada kalırsın; bırak geleceği, bugünü bile göremezsin.

Bu parçada anlatılmak istenen aşağıdakilerden hangisidir?

A) Yazarların, okurlardan gelecek tepkileri umursamadığı

B) Beğenilen yazarların da kalıcılığının, kendilerini sürekli geliştirmelerine bağlı olduğu

C) Sanatçılara olumlu eleştiriler yöneltmek gerektiği

D) Kendini belirli ölçütlerle sınırlayan yazarların özgün ürünler veremeyeceği

E) Yazarların, okurlardan, yazdıklarını değerlendirmelerini beklemediği

Figure 3.2: Example of paragraph to sentence question of the ÖSS 2006 exam (question type 1).

paragraph pairs are at the level. For question type 2, it is predicted that the models which find textual similarity at the sentence-sentence level are suitable. Negligence in question texts expressions such as "incompatible", " not close" like statement changes the correct answer to the problem. Such questions are marked separately as negative questions in the data set.

3.5 Data Input

When producing data, we used digital resources which can be copied on the internet. If the question is not to be copied, the text is entered manually. The questions in the sourcebooks we use are converted to numerical text using an OCT

Buraya geldiğin zaman beni ara.

Aşağıdakilerden hangisi bu cümleyle aynı anlamdadır?

A) Buraya gelirsen beni ara.

B) Buraya gelirken beni ara.

C) Buraya gelmeden önce beni ara.

D) Buraya gelince beni ara.

E) Buraya geleceksen beni ara.

Figure 3.3: Question of finding meaningful sentences in YÖS 2009 exam (question type2).

Exams	Questions	Questions
Tip	Type1	type2
Ales	69	62
DGS	57	37
DGS	57	37
LES	20	18
ÖSS	73	43
ÖYS	58	20
Diğer	53	29
Toplam	330	209

Table 3.1: Question types and number of questions.

(Optical Character Identification) software. All the questions were taken with a high-resolution phone camera to prepare the data for OBS processing. After the OCT procedure, scanning errors are detected and manipulated by hand.

3.6 Converting XML

The XML format was used to make the compiled questions easier to read from different programming platforms. In the first stage, a dialer program was developed on the Java platform to enable the data set in xlsx format to be converted to XML format. The xlsx files are loaded into the memory through the Apache POI component, and each element (element) is exported to the template structure XML file format, which is designated to be represented by a class. The structure of the final output of the data set is shown in pseudo code.

```
<qtypes>
    <instances>
          <etCode>SBS</etCode><id>1.0</id><p>1.0</p><q>4.0</q>
         <question>
      <answer>s3</answer>
      <p1Segment></p1Segment>
      <paragraph>
        <sentence>
                <text>insan taraf[U+FFFD]ndan fark ed{\i}lmed{\i}\u{g}{\i} s\"{u}rece</text>
                      <tokens>
                        <token>
                                <an>{\i}nsan+NOUN+A3SG+PNON+NOM</an>
                                <form>{\i}nsan</form>
                                <pos>NOUN</pos>
                                <text>insan</text>
                        </token>
                      [U+FFFD]
                <token>
```

Figure 3.4: Converting XML Example.

Some important elements (elements) and (attributes) in the last structure of the data set are described below:

- qtypes: The element that groups different types of questions.
- etCode: The exam code to which the problem belongs.
- id: The singular number that is set for each question.
- question: The element that represents the example of each question.
- answer: The correct answer to the problem.

- p1Segment: Represents the word or sentence underlined in the question.
- paragraph: A set of sentences used to answer the question.
- token: The element in which each word is represented by the morphological analysis.

3.7 Morphological Analysis

In the first phase, all paragraphs were passed through the sentence separation algorithm using the NLP library of our study group [13]. Secondly, each sentence of the questions that are divided into sentences is subjected to morphological analysis and candidate analysis branches are obtained. Finally, the candidate analyzes were conducted with the morphological turbidity analysis and the best branch analysis was recorded with the words Tokens. All morphological tags obtained in the formal analysis stages are saved in the XML file structure. The word "edilemediği "in Figure 3.4. A morphological analysis example is given.

Chapter 4

Similarities

4.1 Similarity Types

The purpose of similarity types is to calculate the similarity score between sentences. When calculating the similarity score is to look at the morphological structure of sentences.

4.1.1 Q Gram Similarity

In many applications, it is necessary to measure the similarity of two strings of a limited number of alphabets and symbols. Numerous string similarity measures have been proposed. Well-known criteria are based on the editing distance and the length of the longest common subdirectory. We develop qgram similarity and the concept of distance together. A string length is entered on the characters in a window q length. A string is required to create the 'q' length grams for matching. A match is then rated as the number of q-gram matches in the second string. We have shown and demonstrated that the distance of the editing distance and the length of the longest common subdirectory are the special cases of Qgram distance and similarity respectively. We provide the formal, recursive definitions of Qgram similarity and distance, and the ability to find and use the effective algorithms used to calculate them. We take word similarity measures based on Qgrams. We formulate the appropriate family of them and calculate the test results showing that the new measurements have left their unigram equivalents behind.

4.1.2 Block Distance

Block distance was used by Krause in 1987. Block distance is another name Manhattan Distance. Let's simply define the distance to Manhattan. Gives you the distance you can travel between blocks in the form of land in a city. For example, a taxi is used when calculating how far the road takes using land-like inter-apartment roads. In the K-nn algorithm, euclidean is the most used distance calculation method along with distance. Calculation of the distance you want to measure the combination of the absolute value of the others is collected with other values. The block distance is taken from each other and corresponds to the sum of the resulting results. In the x component x, y is extracted from the y component and collected.

$$BD(A, B) = \sum_{i=1}^{n} |Ai - Bi| \tag{4.1}$$

4.1.3 Levenshtein Distance

The Levenshtein distance is a string metric to measure the difference between the two arrays. The Levenshtein distance between two words is the minimum number of characters required to change one word to another. He received his name from Vladimir Levenshtein, who had reduced this distance in 1965. The adjustment distance is also called. Creates operations such as adding, deleting, modifying or copying a process, or converting one of the specified strings to another. The distance is defined as the minimum number of operations required to change one string to another. The algorithm uses a two-dimensional array (matrix) to increment the value of the letters for different letters to find the change value. In practice, it can be used to conclude word similarities in search results. It converts the block distance value from 1 and converts it to similarity value.

4.1.4 Jaccard Distance

The Jaccard similarity coefficient (Jaccard index) is used to statistically evaluate the similarity between sets. The Jaccard similarity coefficient is obtained by dividing the combination of the number of elements by the number of elements of the intersection of the two sets, as used to obtain the similarity between the two sets. It is the Jaccard index, which is one of the metrics developed to measure the relationship between the texts. The feature extraction is made by dividing the number of properties that are common after the feature extraction is divided by the total number of properties in the two texts. The Jaccard similarity shares the members with two sets to see which members are shared and which are different. As a result of this similarity measure, the data between the two clusters is obtained between 0 % and 100 %. The higher the percentage, the higher the similarity rate. Although it is easy to interpret, it is extremely sensitive to the small size of the frontline. It may produce incorrect results in very small samples or data sets with incomplete observations.

$$JS(A,B) = \frac{S(A \cap B)}{S(A \cap B)} \tag{4.2}$$

4.1.5 Overlap Coefficient

The Overlap coefficient represents the area under the two probability density functions simultaneously. OVL is based on the first days of Karl Pears in different ways. It is being used by the economist Murray Weitzman in the 1970s. It was used to compare income distributions. The overlap coefficient is a measure of similarity with the Jaccard measure that measures the overlap between the two clusters. It is defined as the intersection measure divided by the size of the two sets.

$$OC(A, B) = \frac{S(|A \cap B|)}{S(Min(|A|, |B|))} \tag{4.3}$$

4.2 Heuristic Methods

When we solving problems we developed some other solutions to improving our results. I write a question for calculating scores

ps: Paragraf sentence

chc: Choice

ps1:çözümlemeye kalkısılınca da çesıtlı öğelerın kurduğu bır dünya olarak çıkar karsımıza.

ps3:ıyı bır sır, zengın ayrıntıların, sasırtıcı incelıklerın kaynağıdır.

ps4:Tükenmeyen bir kaynak olan şır, sadece kendısinin olan bir yapıyla ortaya çıkar.

ps5: Kelimeleri bir araya getirme ustası olan ozana göre sır, "sır olduğu için" önemli ve değerlidir.

Question String: Bu parçada asıl söylenmek istenen asağıdakilerden hangisidir?
chc1: Siirin güzelliği, içerdiği ayrıntılardan gelir.

chc2: Sanat Ürünleri arasında en kalıcı olanı siirdir.

chc3: Her ozanın kendine özgü bir siir anlayısı vardır.

chc4: Şiir, kendine özgü özellikleri olan bir sanattır.

chc5: Ozan kendini tümüyle süre adayan kişidir.

Similarity Scores:

S11 = ps1 - chc1=0.2

S12 = ps2 - chc1=0

S13 = ps3 - chc1=0

S14 = ps4 - chc1=0

S15 = ps5 - chc1=0

S21 = ps1 - chc2=0

S22 = ps2 - chc2=0

S23 = ps3 - chc2=0

S24 = ps4 - chc2=0

S25 = ps5 - chc2=0

S31 = ps1 - chc3=0.125

S32 = ps2 - chc3=0.125

S33 = ps3 - chc3=0.125

S34 = ps4 - chc3=0.125

S35 = ps5 - chc3=0.125

S41 = ps1 - chc4=0.14

S42 = ps2 - chc4=0.14

S43 = ps3 - chc4=0.28

S44 = ps4 - chc4=0.42

S45 = ps5 - chc4=0.42

S51 = ps1 - chc5=0.0

S52 = ps2 - chc5=0.0

S53 = ps3 - chc5=0.0

S54 = ps4 - chc5=0.0

S55 = ps5 - chc5=0.0

4.2.1 Max-Max

MaxMax methods we developed for question type-1, first of all, we find max values of our choices after that we accept to right choice maximum values max value. When we look our example we get maximum value chc4 so we get the correct choice is chc4.

4.2.2 Max-Product

MaxProduct methods we developed for question type-1 we found all similarity scores paragraph to sentence after that we product all similarities we accept to correct choice maximum product score.

S11 * S12 * S13 * S14 * S15

= 0.2*0*0*0*0 = 0

S21* S22* S23* S24* S25

= 0*0*0*0*0 = 0

S31* S32* S33* S34* S35

= 0.125*0.125*0.125*0.125*0.125= 0.00003051757

S41* S42* S43* S44* S45

= 0.14*0.14*0.28*0.42*0.42 = 0.0009680832

S41* S42* S43* S44* S45

= 0*0*0*0*0*0 = 0

When we look scores we get maximum scores as chc4 so we accept right choice as chc4.

4.2.3 Max-Average

MaxAverage methods we developed for question type-1 we found all similarity scores paragraph to sentence after that we sum all sentence to sentence scores and we divided it sentence number of paragraph.

(S11 + S12 + S13 + S14 + S15)/5

= (0.2+0+0+0+0)/5 = 0.04

(S21+ S22+ S23+ S24+ S25)/5

= (0+0+0+0+0)/5 = 0

(S31+ S32+ S33+ S34+ S35)/5

= (0.125+0.125+0.125+0.125+0.125)/5= 0.125

(S41+ S42+ S43+ S44+ S45)/5

= (0.14+0.14+0.28+0.42+0.42)/5 = 0.28

(S51+ S52+ S53+ S54+ S55)/5

= (0+0+0+0+0)/5 = 0

When we look scores we get maximum scores as chc4 so we accept correct choice as chc4.

4.2.4 Max-Min

MaxMin methods we developed for question type-1 we found all similarity scores paragraph to sentence after that we found minimum scores after we found all minimum scores maximum score and we accept it the correct choice. We get minimum scores each choice chc1=0, chc2=0, chc3=0.125 ,chc4=0.14 ,chc5=0.0 and we get maximum score chc4 we accept correct choice as chc4.

4.3 Segments

When we calculate scores we used different segments. I separated the sentence which below it ignore uppercase and lowercase. I put a separator which is "+". Sentence: Yazarlar da tıpkı diğer insanlar gibi duygularını düşüncelerini çevrelerinden edinirler.

4.3.1 Word Segment

Calculating similarity scores, first of all, we spared words after that we applied similarity methods.

yazarlar+da+tıpkı+diğer+insanlar+gibi+duygularını+düşüncelerini+çevrelerinden+ edinirler

4.3.2 Syllable Segment

Calculating similarity scores first of all we spared syllable after that we applied similarity methods.

Ya+zar+lar+da +tıp+kı+diğ+er+in+san+lar+

gibi+duy+gu+lar+ı+nı +düş+ün+ce+ler+i+ni+

çev+re+ler+in+den e+di+nir+ler

4.3.3 Character Segment

Calculating similarity scores first of all we spared character after that we applied similarity methods.

y+a+z+a+r+l+a+r+d+a+t+ı+p+k+ı+d+i+ğ+er+i+n+s+a+n+l+a+r+

g+i+b+i+d+u+y+g+u+l+a+r+ı+n+ı+d+ü+ş+ü+n+c+e+l+e+r+i+n+i+ç

+e+v+r+e+l+e+r+i+n+d+e+n+e+d+i+n+i+r+l+e+r

4.3.4 Root (Morfem) Segment

Calculating similarity scores, first of all, we spared root after that we applied similarity methods.

yazar+tıpkı+diğer+insan+gibi+duygu+düşünce+çevre+edinme

Chapter 5

Experiment

5.1 Paragraph to sentence similarity

We applied similarities Qgram, block distance, Levenshtein distance, Jaccard Distance, Overlap Coefficient. We calculate our similarity scores and we answered our questions different ways. First of all we get classic method we calculated similarity scores We accept right choice. After that we calculated our similarity score and we accept highest first two similarity score right. Also we calculated our similarity scores each similarity method and We have accepted the most accurate result as the correct answer. We also improve some heuristic methods such as MaxMax, MaxProduct, MaxAverage and MaxMin. Before we applied our similarities I removed punctuation, removed stop words and I used lower case characters. According to our algorithms, some result shown below:

5.1.1 Word Segmenter

Segment type(word)	**Max Max**	**Max Product**	**Max Average**	**Max Min**
Qgram Similaryty(k=3)	22.324	19.266	25.993	25.076
Block Distance	22.629	22.935	22.629	22.935
Levenshtein Distance	26.911	23.853	24.464	24.464
Jaccard Distance	22.324	25.382	22.324	25.382
Overlap Coefficient	22.324	25.382	22.324	25.382

Table 5.1: Results of paragraph to sentence similarity word segmenter.

We calculated our results using the word segment. According to results, we get maximum results for MaxMax heuristic method using Levistain Distance 26.911 percentage. It is also the best score for word segmenter. After that looking results for MaxProduct, we get best result 25.382 percentage Jaccard distance and overlap coefficient. We also try MaxAverage heuristic method and we get the best result Qgram similarity (k=3) 25.993 percentage. Finally we find our results MaxMin heuristic method 25.382 percentage.

Segment type(word)	**Max Max**	**Max Product**	**Max Average**	**Max Min**
Qgram Similaryty(k=3)	**49.541**	40.366	**49.847**	**46.483**
Block Distance	36.669	36.391	35.779	35.168
Levenshtein Distance	48.623	42.201	44.036	44.036
Jaccard Distance	42.813	44.342	44.036	44.648
Overlap Coefficient	40.366	**44.648**	44.648	44.648

Table 5.2: Results of paragraph to sentence similarity accept higest two answers correct word segmenter.

We calculated our results using the word segment and accept higest two answers correct. According to results, we get maximum results for MaxMax heuristic method using Qgram similarity 49.541 percentage. After that looking results for MaxProduct, we get best result 44.648 percentage overlap coefficient. We also try MaxAverage heuristic method and we get the best result Qgram similarity (k=3) 49.847 percentage. It is also the best score for word segmenter. Finally we find our results MaxMin heuristic method 46.483 percentage.

	Max Max	**Max Product**	**Max Average**	**Max Min**
Segment Type(Word)	25.993	25.076	24.159	24.770

Table 5.3: Results of paragraph to sentence similarity ensemble method word segment.

We calculated our results and we get best result MaxMax 25.993.

5.1.2 Root Segmenter

Segment type(Root)	**Max Max**	**Max Product**	**Max Average**	**Max Min**
Qgram Similaryty(k=3)	28.134	26.911	28.194	28.134
Block Distance	22.935	22.324	22.935	22.935
Levenshtein Distance	25.382	22.629	25.382	25.382
Jaccard Distance	21.21	21.21	21.21	21.21
Overlap Coefficient	21.21	21.21	21.21	21.21

Table 5.4: Results of paragraph to sentence similarity root segmenter.

We calculated our results using the root segment. According to the results, we get maximum score MaxMax, MaxProduct and MaxMin heuristic methods best score using Qgram similarity which is 28.134 percentage. This score also best score all experiment. Using MaxProduct we also get best results Qgram similarity is 26.911 percentage.

Segment type(Root)	Max Max	Max Product	Max Average	Max Min
Qgram Similaryty(k=3)	**49.235**	44.954	**49.235**	46.483
Block Distance	48.929	**48.929**	48.929	**48.929**
Levenshtein Distance	44.342	43.119	44.342	44.342
Jaccard Distance	47.4	47.4	47.4	47.4
Overlap Coefficient	45.565	45.565	45.565	45.565

Table 5.5: Results of paragraph to sentence similarity accept higest two answers correct root segmenter.

We calculated our results using the root segment and accept higest two answers correct. According to results, we get maximum results for MaxMax heuristic method using Qgram similarity (k=3) 49.235 percentage. After that looking results for MaxProduct, we get best result 48.929 percentage block distance. We also try MaxAverage heuristic method Qgram similarity (k=3) 49.235 percentage. It is also the best score for root segmenter. Finally we find our results MaxMin heuristic method 48.929 percentage.

	Max Max	Max Product	Max Average	Max Min
Segment Type(Root)	22.018	22.018	22.018	22.018

Table 5.6: Results of paragraph to sentence similarity ensemble method root segment.

5.1.3 Syllable Segment

Segment type(Syllable)	Max Max	Max Product	Max Average	Max Min
Qgram Similaryty(k=3)	22.324	19.266	**25.688**	25.076
Block Distance	22.629	22.935	22.629	22.935
Levenshtein Distance	**26.911**	23.547	24.464	24.464
Jaccard Distance	22.018	**25.382**	22.018	**25.382**
Overlap Coefficient	22.018	**25.382**	22.018	**25.382**

Table 5.7: Results of paragraph to sentence similarity syllable segmenter.

We calculated our results using Syllable segment. We calculated our results using MaxMax heuristic method we get our best score using Levenstein distance 26.911 percentage using Levenshtein distance. It is also best score for Syllable Segment. After that we look MaxProduct results we get the best scores using Jaccard Distance and Overlap Coefficient 25.382 percentage. Next we look MaxAverage we get best score Qgram Similarity (k=3) which is 25.688 percentage. Finally we get results MaxMin heuristic method we get best scores Qgram similarity, Jaccard similarity and overlap Coefficient which score is 25.382 percentage.

Segment type(Syllable)	Max Max	Max Product	Max Average	Max Min
Qgram Similaryty(k=3)	**49.541**	40.978	**49.847**	**47.4**
Block Distance	37.003	35.168	35.474	34.862
Levenshtein Distance	48.623	42.813	44.036	44.036
Jaccard Distance	42.201	44.036	43.119	44.342
Overlap Coefficient	40.672	**44.342**	43.425	44.342

Table 5.8: Results of paragraph to sentence similarity accept higest two answers correct syllable segmenter.

We calculated our results using the syllable segment and accept higest two answers correct. According to results, we get maximum results for MaxMax heuristic method using Qgram similarity (k=3) 49.541 percentage. After that looking results for MaxProduct, we get best result 44.342 percentage block distance. We also try MaxAverage heuristic method Qgram similarity (k=3) 49.847 percentage. It is also the best score for syllable segmenter. Finally we find our results MaxMin heuristic method 47.4 percentage.

	Max Max	**Max Product**	**Max Average**	**Max Min**
Segment Type(Syllable)	25.993	25.382	24.159	24.770

Table 5.9: Results of paragraph to sentence similarity ensemble method syllable segment.

We calculated our results and we get best result Max Average 25.993.

5.1.4 Character Segment

Segment type(Character)	**Max Max**	**Max Product**	Max Average	**Max Min**
Qgram Similaryty(k=3)	22.324	19.266	**25.688**	25.076
Block Distance	22.011	20.795	20.183	24.770
Levenshtein Distance	22.629	22.324	24.464	23.853
Jaccard Distance	22.935	23.547	22.935	25.547
Overlap Coefficient	22.0183	25.382	22.0183	25.382

Table 5.10: Results of paragraph to sentence similarity character segmenter.

We calculated our results also Character Segment, first of all, we look MaxMax method we get the best result Jaccard Distance similarity 22.935 percentage. After that we look MaxProduct we get best score 25.382 percentage. Next, we get MaxAverage best score is 25.688 with Qgram Similarity (k=3). Finally we get best score MaxMin heuristic method 25.382 percentage using Overlap Coefficient.

Segment type(Character)	**Max Max**	**Max Product**	Max Average	**Max Min**
Qgram Similaryty(k=3)	49.541	40.978	9.847	47.4
Block Distance	37.308	40.672	35.474	42.507
Levenshtein Distance	44.342	38.837	44.648	43.730
Jaccard Distance	**50.764**	**50.152**	**52.293**	**51.987**
Overlap Coefficient	35.779	33.027	32.110	34.556

Table 5.11: Results of paragraph to sentence similarity accept higest two answers correct character segmenter.

We calculated our results using the character segment and accept higest two answers correct. According to results, we get maximum results for MaxMax heuristic method using Jaccard distance 50.764 percentage. After that looking results for MaxProduct, we get best result 50.152 percentage Jaccard distance. We also try MaxAverage heuristic method Jaccard distance 52.293 percentage. It is also the best score for charachter segmenter. Finally we find our results MaxMin heuristic method 51.987 percentage.

	Max Max	**Max Product**	**Max Average**	**Max Min**
Segment Type(Character)	22.935	23.547	24.159	23.853

Table 5.12: Results of paragraph to sentence similarity ensemble method character segment.

We calculated our results and we get best result Max Average 24.159.

5.2 Sentence To Sentence Similarity

We applied similarities Qgram, block distance, Levenshtein distance, Jaccard Distance, Overlap Coefficient. First of all we get classic method we calculated similarity scores We accept right choice. After that we calculated our similarity score and we accept highest first two similarity score right. Also we calculated our similarity scores each similarity method and We have accepted the most accurate result as the correct answer. Before we applied our similarities I removed punctuation and I used lower case characters. According to our algorithms some result is shown below:

5.2.1 Word Segmenter

Similiraty type	**Qgram Similarity**	**Block Distance**	**Levenshtein Distance**	**Jagard Distance**	**Overlap Coefficient**
	18.536	**23.414**	16.585	19.024	18.536

Table 5.13: Results of paragraph to sentence similarity word segmenter.

We applied our results using word segment we get the best result using Block Distance 23.414 percentage.

Similiraty type	**Qgram Similarity**	**Block Distance**	**Levenshtein Distance**	**Jagard Distance**	**Overlap Coefficient**
	37.073	**37.560**	34.14	35.121	36.097

Table 5.14: Results of sentence to sentence similarity accept higest two answers correct word segmenter.

We applied our results using word segment we get the best result using Block Distance 37.560 percentage.

Similarity Score	19.024

Table 5.15: Results of sentence to sentence similarity ensemble method word segmenter.

We applied ensemble method using word segment and we get result 19.024.

5.2.2 Root Segmenter

Similiraty type	**Qgram Similarity**	**Block Distance**	**Levenshtein Distance**	**Jagard Distance**	**Overlap Coefficient**
	17.056	15.609	11.707	17.056	**17.056**

Table 5.16: Results of sentence to sentence similarity root segmenter.

We applied our results using root segment we get best scores using Qgram similarity (k=3), Jaccard similarity and Overlap Coefficient methods which score 17.56 percentage.

Similiraty type	**Qgram Similarity**	**Block Distance**	**Levenshtein Distance**	**Jagard Distance**	**Overlap Coefficient**
	37.073	36.585	35.609	35.121	**37.073**

Table 5.17: Results of sentence to sentence similarity accept higest two answers correct root segmenter.

We applied our results using word segment we get the best result using Qgram Similarity(k=3) and Overlap Coefficient 37.073 percentage.

Similarity Score	16.097

Table 5.18: Results of sentence to sentence similarity ensemble method root segmenter.

We applied ensemble method using word segment and we get result 19.024.

5.2.3 Syllable Segment

Similiraty Type	Qgram Similarity	Block Distance	Levenshtein Distance	Jagard Distance	Overlap Coefficient
	18.536	**23.414**	17.560	20.487	20.00

Table 5.19: Results of sentence to sentence similarity syllable segmenter.

We applied our results using syllable segment we get the best result using block distance. It is also the best score for sentence to sentence

Similiraty Type	Qgram Similarity	Block Distance	Levenshtein Distance	Jagard Distance	Overlap Coefficient
	38.536	38.536	34.634	36.097	36.585

Table 5.20: Results of sentence to sentence similarity accept higest two answers correct syllable segmenter.

We applied our results using word segment we get the best result using Qgram Similarity(k=3) and Overlap Coefficient 38.536 percentage.

Similarity Score	16.097

Table 5.21: Results of sentence to sentence similarity ensemble method syllable segmenter.

We applied ensemble method using syllable segment and we get result 16.097.

5.2.4 Character Segment

Similiraty Type	Qgram Similarity	Block Distance	Levenshtein Distance	Jagard Distance	Overlap Coefficient
	18.536	18.048	18.536	**19.024**	12.682

Table 5.22: Results of sentence to sentence similarity character segmenter.

We applied our results for character segment we get best result Jaccard similarity.

Similiraty type	Qgram Similarity	Block Distance	Levenshtein Distance	Jagard Distance	Overlap Coefficient
	38.536	**39.024**	36.585	30.731	32.195

Table 5.23: Results of sentence to sentence similarity accept higest two answers correct character segmenter.

We applied our results using word segment we get the best result using Block Distance 39.024 percentage.

Similarity Score	13.658

Table 5.24: Results of sentence to sentence similarity ensemble method character segmenter.

5.3 Test Screen

We developed an application for testing our questions. I created an frontend application with react and backend application with Spring Boot. I used created XML which include our questions. I show our question,result and choices on the screen.

Figure 5.1: Test Screen

Chapter 6

Conclusion

This is an introductory study that aims to measure the degree of similarity of two texts in terms of textual similarity. We developed an application on Solving ÖSYM Exam Questions Semantic textual similarity is a well-known natural language processing (NLP) task which aims to measure the degree of similarity of two texts in terms of meanings.

In this study investigate best semantic textual similarity measurement modelling techniques for the Turkish language at paragraph-to-sentence and sentence-to-sentence levels. We exploit morphological knowledge of the Turkish language as a prior input, by using morphological disambiguation toolkit of our study group which automatically annotates morphological tags of words (words, syllable, roots, character etc.) in morpheme-level while disambiguating possible parse-trees at the sentence-level. Calculating similarity we used some similarity methods such as Q Gram Similarity, Block Distance, Levenshtein Distance, Jaccard Distance, Overlap Coefficient. Heuristic Methods paragraph to sentence and sentence to sentence. We also developed some heuristic methods Max-Max, Max-Product, Max-Average and Max-Min.

We tried to find the semantic similarity between the sentences using the morphological similarities of the sentence. While trying to understand semantic similarity, some situations caused similarity scores to be poor. Morphologically very similar sentences were effective in making the results worse. Proverbs and idioms badly affected results.

We can carry out a study by using the wordnet library and combining it with our current work while finding the similarity score in the next term study.

Question:Aşağıdakilerden hangisi, bu cümleye yakın anlamdadır?

A) Gerçegin yalnız bir parçasını soylemek **gercek** ustune hicbir sey soylememektir. **gercek** degistirilerek anlatılırsa inandiriciligindan cok sey.

B)gercegin yalniz bir parcasini soylemek **gercek** ustune hicbir sey soylememektir. **gercek** ancak butunuyle ortaya konuldugu zaman eksiksiz anlatilmis olur.

C)gerçegin yalniz **bir** parcasını soylemek ustune " hicbir sey soylememektir.

gerçegi butun yonleriyle anlatmak sakincaliysa onun **bir** bolumu anlatilmalidir.

D)gerçegin yalniz bir parcasini soylemek gercek ustune hicbir sey soylememektir gerçegi anlatabilmenin kosulu onu butun yonleriyle bilmektir

E)gerçegin yaliz bir parcasini soylemek gercek ustune hicbir sey soylememektir gercekler gizlenmek isteniyorsa degisik anlatim yolları aranmalıdır.

For example at this example we can see correct answer is B but when we look A there is one more stop word "şey" it makes similarity score higher because of this our application find answer wrong.

References

[1] F. Hill, K. Cho, and A. Korhonen, "Learning distributed representations of sentences from unlabelled data," *arXiv preprint arXiv:1602.03483*, 2016.

[2] M. Pagliardini, P. Gupta, and M. Jaggi, "Unsupervised learning of sentence embeddings using compositional n-gram features," *arXiv preprint arXiv:1703.02507*, 2017.

[3] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International conference on machine learning*, 2014, pp. 1188–1196.

[4] L. Yujian and L. Bo, "A normalized levenshtein distance metric," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 6, pp. 1091–1095, 2007.

[5] Y. Li, D. McLean, Z. A. Bandar, K. Crockett *et al.*, "Sentence similarity based on semantic nets and corpus statistics," *IEEE Transactions on Knowledge & Data Engineering*, no. 8, pp. 1138–1150, 2006.

[6] N. Miura and T. Takagi, "Wsl: sentence similarity using semantic distance between words," in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 2015, pp. 128–131.

[7] G. Ercan, O. Erkek, O. Açıkgöz, R. Özçelik, S. Parlar, and O. T. Yıldız, "Türkçe anlamsal söylem ve cümle benzerligi analizleriiçin veri kümesi olusturma yöntemi."

[8] T. Mikolov, "Statistical language models based on neural networks," *Presentation at Google, Mountain View, 2nd April*, vol. 80, 2012.

[9] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[10] T. K. Landauer, P. W. Foltz, and D. Laham, "An introduction to latent semantic analysis," *Discourse processes*, vol. 25, no. 2-3, pp. 259–284, 1998.

[11] "Toefl synonym questions (state of the art)," https://aclweb.org/aclwiki/TOEFL_Synonym_Questions, accessed: 2019-09-16.

[12] N. H. Arai, "The impact of ai—can a robot get into the university of tokyo?" *National Science Review*, vol. 2, no. 2, pp. 135–136, 2015.

[13] O. Görgün and O. T. Yildiz, "A novel approach to morphological disambiguation for turkish," in *Computer and Information Sciences II*. Springer, 2011, pp. 77–83.