

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

**KEYWORD EXTRACTION FROM
SPEECH DATA USING CENTER BASED CLUSTERING**

HAKKI YAVUZ ERZURUMLU
A THESIS SUBMITTED FOR THE DEGREE OF
MASTER OF SCIENCE
DEPARTMENT OF COMPUTER ENGINEERING

GEBZE
2022

T.R.
GEBZE TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

KEYWORD EXTRACTION FROM SPEECH
DATA USING
CENTER BASED CLUSTERING

HAKKI YAVUZ ERZURUMLU
A THESIS SUBMITTED FOR THE DEGREE OF
MASTER OF SCIENCE
DEPARTMENT OF COMPUTER ENGINEERING

THESIS SUPERVISOR
PROF. DR. YUSUF SİNAN AKGÜL

GEBZE
2022

T.C.
GEBZE TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MERKEZ TABANLI KÜMELEME
KULLANARAK KONUŞMA
VERİLERİNDEN ANAHTAR KELİME
ÇIKARTILMASI

HAKKI YAVUZ ERZURUMLU
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

TEZ DANIŞMANI
PROF. DR. YUSUF SİNAN AKGÜL

GEBZE
2022



YÜKSEK LİSANS JÜRİ ONAY FORMU

GTÜ Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 03/02/2022 tarih ve 2022/06 sayılı kararıyla oluşturulan jüri tarafından 21/02/2022 tarihinde tez savunma sınavı yapılan Hakkı Yavuz ERZURUMLU'nun tez çalışması Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

JÜRİ

ÜYE

(TEZ DANIŞMANI)

: Prof. Dr. Yusuf Sinan AKGÜL

ÜYE

: Dr. Burcu YILMAZ

ÜYE

: Doç Dr. Hidayet TAKCI

ONAY

Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun

...../...../..... tarih ve/..... sayılı kararı.

İMZA/MÜHÜR

SUMMARY

In this work, one of the main subjects of natural language processing, the conversion of speech to text and the extraction of keywords from the text were studied. It is aimed to convert the audio files taken from speech-based applications into text and then automatically label the content by making keyword analysis on the obtained texts. The developed system is online end-to-end and includes speech-to-text model training, data collection and labeling, the model that provides the extraction of keywords from the obtained text, and systems that run this model end-to-end. Transformer-based custom models, pre-trained models (VOSK, Wav2Vec 2.0), online model (Google Speech 2 Text) models were used and compared with each other for speech-to-text conversion. For the keyword extraction model, the RAKE, YAKE!, TF/IDF models that work online and produce Turkish results are compared with the keyword finding model we have developed and the results are examined. The main contributions of this work are, instead of using raw content, use of word roots before calculating any scores, penalization of verbs and verb groups, a naïve dynamic cluster selection algorithm is used to select multiple clusters. The effects of the different speech-to-text model scores used on the keyword extraction model are shown. The keyword extraction model we developed produces better results than other methods in low-score speech texts. Also, a benchmarking dataset was created which was labeled by humans. All services developed for the system to be scalable have been written in accordance with virtualization so that they will run over docker.

Key Words: Natural Language Processing, Speech to Text, Keyword Extraction, Artificial Intelligence, Deep Learning, Unsupervised Learning, Data Labeling.

ÖZET

Bu çalışmada doğal dil işlemenin temel konularından olan sesin metne dönüşümü ve metin üzerinden anahtar kelimelerin bulunması üzerine çalışılmıştır. Konuşma tabanlı uygulamalardan alınan ses dosyalarının metne çevrilmesi ve sonrasında elde edilen metinler üzerinden anahtar kelime analizi yapılarak içeriğin otomatik olarak etiketlenmesi amaçlanmıştır. Geliştirilen sistem uçtan uca çevrimiçi olup konuşmadan metne model eğitimi, veri toplanması ve etiketlenmesi, elde edilen metin üzerinden anahtar kelimelerin çıkarılmasını sağlayan modelin ve bu modeli uçtan uca çalıştıran sistemleri kapsamaktadır. Konuşmadan metne çevrimi için, transformer tabanlı özel modeller, önceden eğitilmiş modeller (VOSK, Wav2Vec 2.0), çevrimiçi online model (Google Speech 2 Text) modelleri kullanılmış ve birbirleriyle kıyaslanmıştır. Anahtar kelime çıkarma modeli için, çevrimiçi çalışan ve Türkçe sonuç üretebilen RAKE, YAKE!, TF/IDF modelleri ile kendi geliştirdiğimiz anahtar kelime bulma modeli kıyaslanmış ve sonuçları irdelenmiştir. Bu çalışmanın ana katkıları, kelime içeriğini olduğu gibi kullanmak yerine, kelime köklerinin hesaplanması sonrasında puanlama yapılması, çokça geçen fiillerin ve fiil gruplarının cezalandırılması, birden fazla aday kümede dinamik olarak işlem yapabilmek için yeni bir dinamik küme seçim algoritması kullanılmasıdır. Kullanılan farklı konuşmadan metne model skorlarının anahtar kelime çıkarma modeline etkileri gösterilmiştir. Geliştirdiğimiz anahtar kelime çıkarma modeli düşük skorlu konuşma metinlerinde de diğer yöntemlere göre iyi sonuçlar üretmektedir. Ayrıca, sistemin kıyaslanması için insanlar tarafından etiketlenen bir veri seti oluşturuldu. Sistemin ölçeklenebilir olması için de geliştirilen tüm servisler sanallaştırmaya uygun olarak yazılmış böylece docker üzerinden çalışacak hale gelmiştir.

Anahtar Kelimeler: Doğal Dil İşleme, Anahtar Kelime Çıkarma, Konuşmadan Metne Dönüşüm, Yapay Zeka, Derin Öğrenme, Gözetimsiz Öğrenme, Veri Etiketleme.

ACKNOWLEDGEMENTS

I would like to express my deep and sincere gratitude to my supervisor, Prof. Yusuf Sinan AKGÜL, who not only shared his profound scientific knowledge with me but also taught me great lessons in life. His support, suggestions, encouragement, and patience gave me the drive and will to complete this work.

I wish to express my warm and sincere thanks to ARDIC R&D for their collaboration with this research. Tunç Kahveci (CEO), Haluk Tüfekçi (CTO) and AI Team always believed me and supported me no matter what.

I am grateful to my parents for their love and support. Especially Mom, I couldn't be here if you didn't support me.

Finally, I want to thank Ayşe Gül TÜRK ERZURUMLU, who showed me the beauty of love. She always supported me with everything she got.

I am dedicating this thesis to my beloved son Atilla Hakan.

TABLE OF CONTENTS

	<u>Page</u>
SUMMARY	v
ÖZET	vi
ACKNOWLEDGEMENTS	vii
TABLE OF CONTENTS	viii
LIST OF ABBREVIATIONS AND ACRONYMS	x
LIST OF FIGURES	xi
LIST OF TABLES	xii
1. INTRODUCTION	1
1.1. Motivation	1
1.2. Natural Language Processing	3
1.2.1. Key Applications of Natural Language Processing (NLP)	3
2. DATA COLLECTION	5
2.1. ArVoice	5
2.2. ArdicSRTCCollector	11
3. DATA EMBEDDING	13
3.1. Word2Vec	13
3.2. Global Vectors for Word Representation (GloVe)	14
3.3. FastText	14
3.4. BERT	14
4. SPEECH TO TEXT	16
4.1. Dataset	16
4.1.1. Common Voice (ENG, TR)	17
4.1.2. ArdicSRTCCollector (YouTube Videos)	17
4.1.3. ArVoice	17
4.2. Model	18
4.3. Experiments on Speech to Text	24
4.3.1. Evaluation Metrics	24

4.3.2. Wav2Vec2	26
4.3.3. VOSK	26
4.4. Speech to Text Results	27
5. KEYWORD EXTRACTION	28
5.1. Dataset	29
5.2. Model	31
5.3. Experiments on Keyword Extractions	32
5.3.1. TF/IDF	35
5.3.2. RAKE – Rapid Automatic Keyword Extraction	35
5.3.3. YAKE! – Collection Independent Automatic Keyword Extractor	35
5.3.4. KeyBERT	36
5.4. Keyword Extraction Results	37
6. SYSTEM DEPLOYMENT	43
7. CONCLUSIONS & FUTURE WORKS	48
7.1. Tags Done By Humans	50
7.2. Overall Results Keyword Scores of Each Video	59
7.3. STT Keyword Scores of Each Benchmark Video	65
REFERENCES	70
BIOGRAPHY	74
APPENDICES	75

LIST OF ABBREVIATIONS AND ACRONYMS

<u>Abbreviations and Acronyms</u>	<u>Explanations</u>
ASR	: Automated Speech Recognition
BERT	: Bidirectional Encoder Representations from Transformers
CBOW	: Continuous Bag-of-Words
BERT	: Bidirectional Encoder Representations from Transformers
FFN	: Feed Forward Network
GELU	: Gaussian Error Linear Unit
GloVe	: Global Vectors for Word Representation
MLM	: Masked Language Model
NLP	: Natural Language Processing
NMT	: Neural Machine Translation
PE	: Positional Encoding
POS	: Part-of-Speech
PostLN	: Post Layer Normalization
RELU	: Rectified Linear Unit
SPA	: Single Page Application
STT	: Speech to Text

LIST OF FIGURES

<u>Figure No:</u>	<u>Page</u>
1.1: Growth of Video Conferencing Applications after COVID-19.	1
1.2: High-level system flow.	2
2.1: ArVoice High-Level Blocks.	5
2.2: ArVoice Recording Tab.	6
2.3: ArVoice Recording Tab – After a Record.	7
2.4: ArVoice Listening Tab.	8
2.5: ArVoice Listening Tab – After Playback.	9
2.6: ArVoice Backend MicroService.	10
2.7: ArdicSRTCollector YouTube Data Generation Flow.	12
4.1: Calculating Positional Encoding.	19
4.2: Multi-Head Representations.	20
4.3: Encoder (Left) and Decoder (Right) stack.	22
4.4: Transformer Model for Speech Recognition.	23
4.5: Dataset Preprocessing Steps.	24
4.6: Wav2Vec2 Encoder Block.	26
5.1: IDF Calculation.	32
5.2: Cluster size grid search.	33
5.3: Histogram of the remaining input words.	39
5.4: Total cluster votes.	40
5.5: Calculating word votes.	41
5.6: Distribution of the cluster votes.	41
5.7: Votes and scores of the words.	42
6.1: Deployed System Components.	43
6.2: Sample output from Zemberek’s lemmatization function.	44
6.3: Architecture of services and communications between them.	45
6.4: Keyword Extractor Client Raw and Word Cloud Responses.	46
6.5: Real Time STT and Keyword Results.	47

LIST OF TABLES

<u>Table No:</u>	<u>Page</u>
4.1: Dataset List.	17
4.2: Speech to Text Results.	27
5.1: Benchmark Dataset Samples.	30
5.2: Training Dataset.	31
5.3: Overall Results.	37
5.4: Keyword extraction results with different STT models.	37
5.5: Keyword results for word embedding method with relevant models.	38
7.1: Tags done by humans.	50
7.2: Overall keyword results of each benchmark video.	59
7.3: STT Algorithm Based Scores of Each Video.	65

1. INTRODUCTION

Keyword extraction is one of the most fundamental problems of natural language processing. Being able to summarize a text with a few words plays a key role in solving many problems such as categorizing, summarizing, labeling these texts, and interactive searches through these tags.

1.1. Motivation

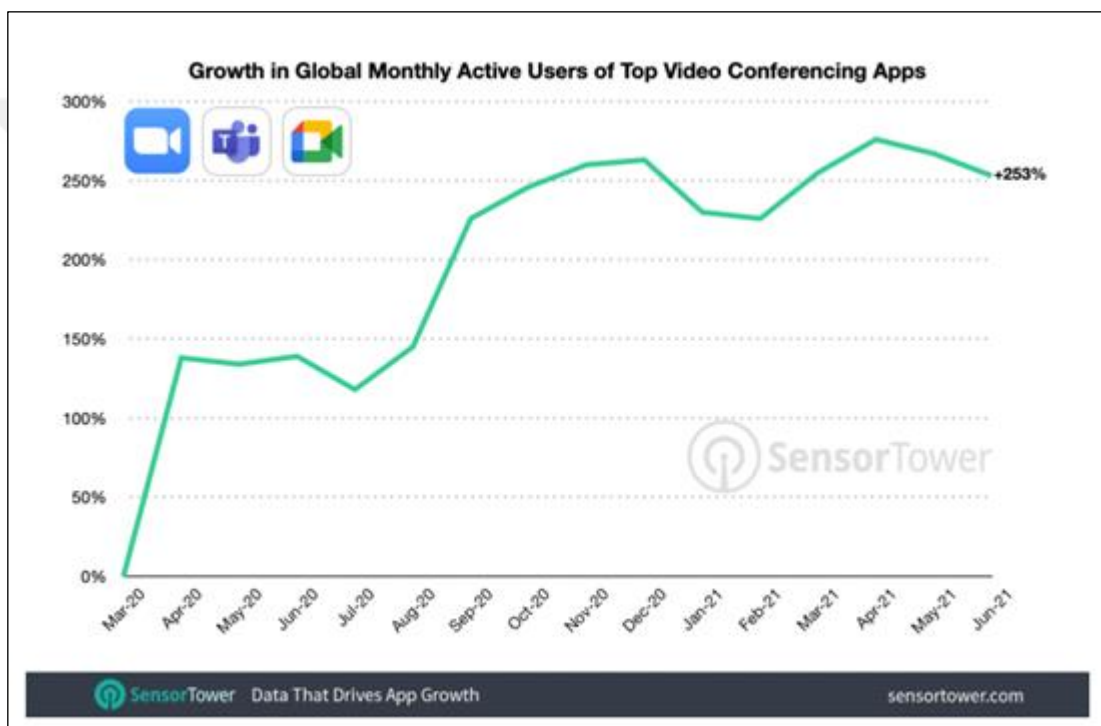


Figure 1.1 Growth of Video Conferencing Applications after COVID-19.

Keyword extraction is one of the most fundamental problems of natural language processing. Being able to summarize a text with a few words plays a key role in solving many problems such as categorizing, summarizing, labeling these texts, and interactive searches through these tags.

Today, due to COVID-19, remote working has increased [Brynjolfsson, 2020], and people have started to meet with their loved ones through video conferencing applications [Web 1, 2022] due to restrictions. Summaries of these meetings or meetings are generally not kept or summarized by one person.

In order to extract keywords from audio files, the audio content must first be translated into text. Translating speech into text is another important problem of natural language processing. In other words, in order to solve this problem, firstly the voice should be converted into a meaningful text. Afterward keyword extraction should be done on this text.

Keywords can also provide information about the content of the conference taking place at that time, as well as the extraction of summaries after the meeting. Instead of manually tagging the live broadcasts that people attend, these tags can be created automatically, so that you can be informed about the content without attending the conference.

Our main purposes in this study are, extracting keywords from the audio content close to real-time, creating tagged content on speech-based applications. The hybrid system we have developed gives significantly better results when compared to the online systems developed for Turkish.

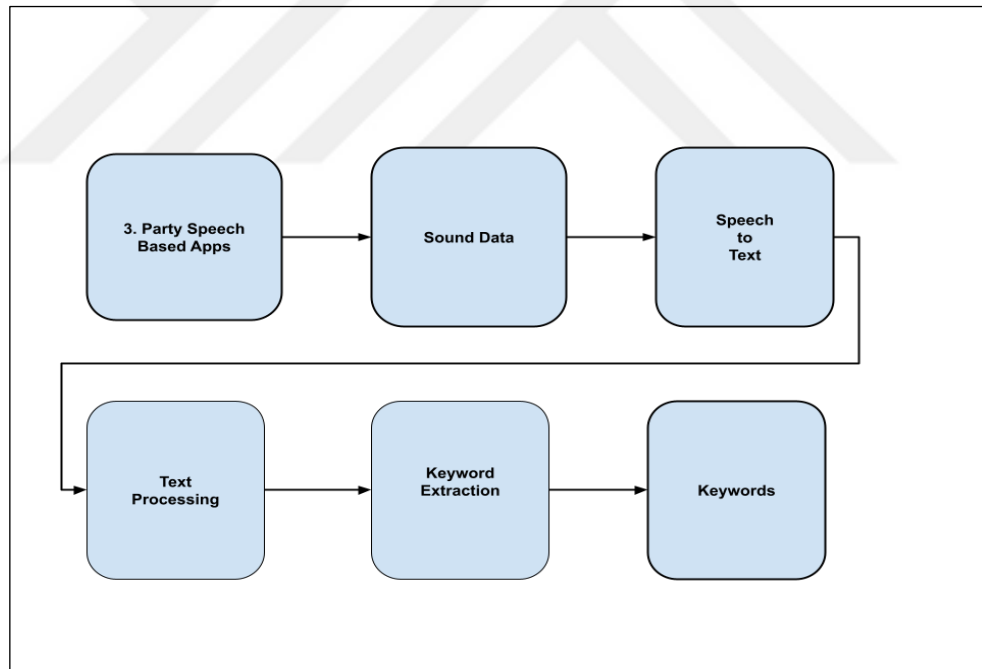


Figure 1.2 High-level system flow.

We can summarize the NLP problems that need to be solved as follows, converting the audio file to a vector, training a speech-to-text conversion model, pre-processing the obtained text, converting text input to vectors, training the keyword extraction model.

1.2. Natural Language Processing

Natural language, linguistics is any language that has evolved naturally in humans through use and repetition without conscious planning or premeditation [Winograd, 1972]. Natural language can differ like speech or singing. They are different from human-constructed formal languages such as programming languages.

Natural language processing (NLP) is the manipulation of natural language by computers.

1.2.1. Key Applications of Natural Language Processing (NLP)

Natural languages have common features such as grammar, syntax, and context. There are several tasks to solve and explain each of these features.

Machine translation, the basic task is converting one natural language to another natural language. There are different methods to accomplish this task such as rule-based [Torregrosa, 2019], example-based [Carl, 1999], statistical [Koehn, 2009], neural networks [Bahdanau, 2014. Singh 2017. Cho, 2014]. Google Translate is a great example of this task. Current state-of-the-art models are using neural machine translation (NMT) [Allison, 2018] which is based on deep learning models.

Text summarization, getting a summarized version of a document is a common task of every natural language [Allahyari, 2017]. The summarized document should include the key ideas of the content. There are two general approaches: abstractive [Nallapati, 2016] and extractive [Miller, 2019] summarization. In abstractive summarization, the NLP model creates new sentences that might not be included in the text. In extractive summarization, the model extracts phrases and sentences from existing text.

Sentiment analysis is a special version of text categorization which tries to detect whether the sentiment of text is positive, negative, or neutral. It is a very common task to analyze customer feedback, movie reviews, market research, tweet analysis [Rao, 2016], etc. to ensure customer satisfaction and service quality [Yue, 2019].

Automated Speech Recognition (ASR), most of the NLP algorithms run with text, but it can also be applied to speech input. ASR systems transcribe speech data to a stream of words.

Keyword (Key-phrase) extraction, finding keywords in a document is another important area [Rose, 2010] of NLP as it opens a way to summarization, indexing, and categorizing the text.

Named-Entity-Recognition (NER), the task [Nadeau and David, 2007] involves extracting names, places, people, and things from input data. It helps to solve open domain question answering problems [Eiselt and Figueroa, 2013]. It also helps to tag them as a person, company, city or celebrity, etc.

Part-of-Speech (POS) Tagging, to build exact relations between words, the NLP model should find the defined tag set [Petrov and Dipanjan, 2011] and tag the POS. Common POS tagging methods include lexical, rule-based, probabilistic models, recurrent neural networks.

Question answering [Hirschman and Gaizauskas, 2001] systems is another critical NLP task. The model must get reading comprehension ability. Then it can answer the questions. Understanding context in the text is difficult for the machines. Also, question answering requires both natural language understanding and general knowledge [Quarteroni and Silvia, 2009]. We can count an example: reading Wikipedia articles to answer open-domain questions [Chen and Danqi, 2017].

Currently, deep learning [Goodfellow and Bengio, 2016] techniques [Goldberg, 2016] became popular to solve these NLP problems. Previous techniques require a feature engineering or feature preparation step. Deep learning techniques handle features engineering inside of themselves.

2. DATA COLLECTION

Data has critical importance to any kind of deep learning algorithm. In order to solve the speech-to-text problem, we need lots of labeled Turkish audio and its corresponding transcription.

To improve our dataset, we developed a web application (ArVoice) that allows end-users to confirm them by listening to audio recordings, a python library (ArdicSRTCCollector) that downloads videos with subtitles from YouTube and trims them according to the time intervals in the subtitles and determines the corresponding text.

2.1. ArVoice

ArVoice is a data collector, single page application (SPA) that allows users to record and validate their voice records.

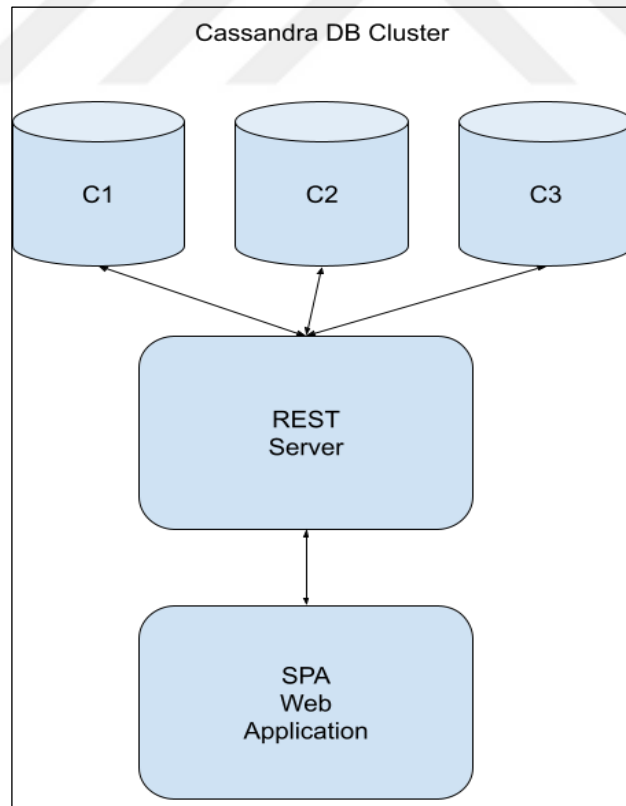


Figure 2.1 ArVoice High-Level Blocks.

ArVoice SPA application contains two main screens:

- Voice recording screen: The user is prompted a random sentence. S/he reads the sentence and sends the recorded voice datum to the database. Users can skip or retry recording voice. The system has also a flag for reporting inappropriate sentences.
- Voice listening screen: The user receives random recorded audio with a corresponding text. If audio matches with the text then s/he can validate it as true, otherwise false. Like a voice recording screen, users can skip listening and receive another random sentence.

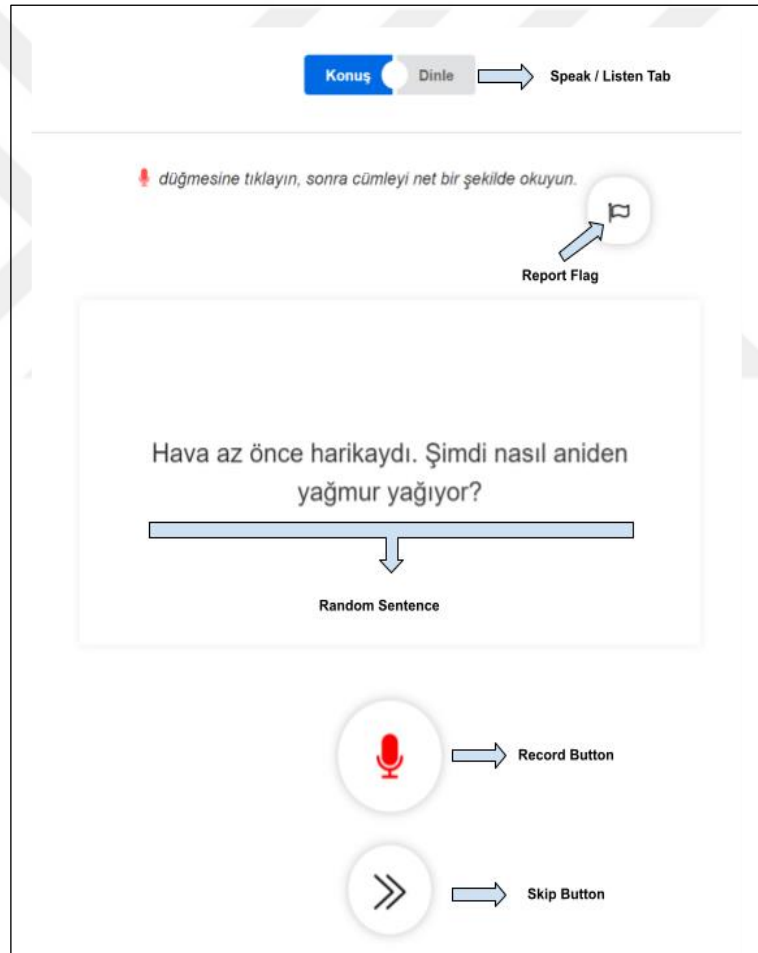


Figure 2.2 ArVoice Recording Tab.

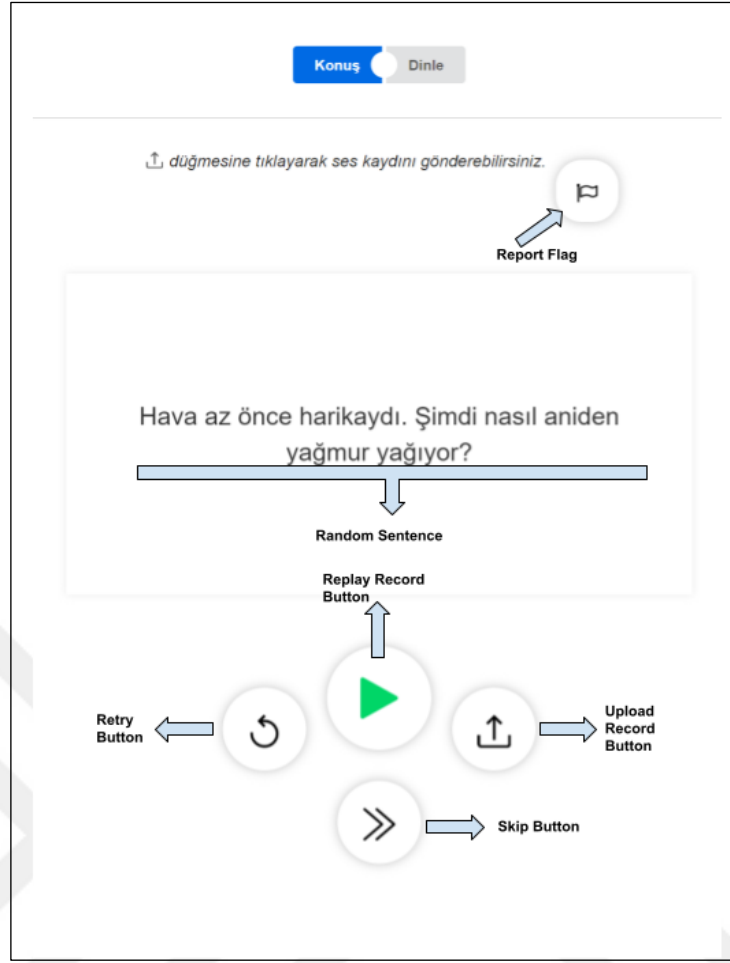


Figure 2.3 ArVoice Recording Tab – After a Record.

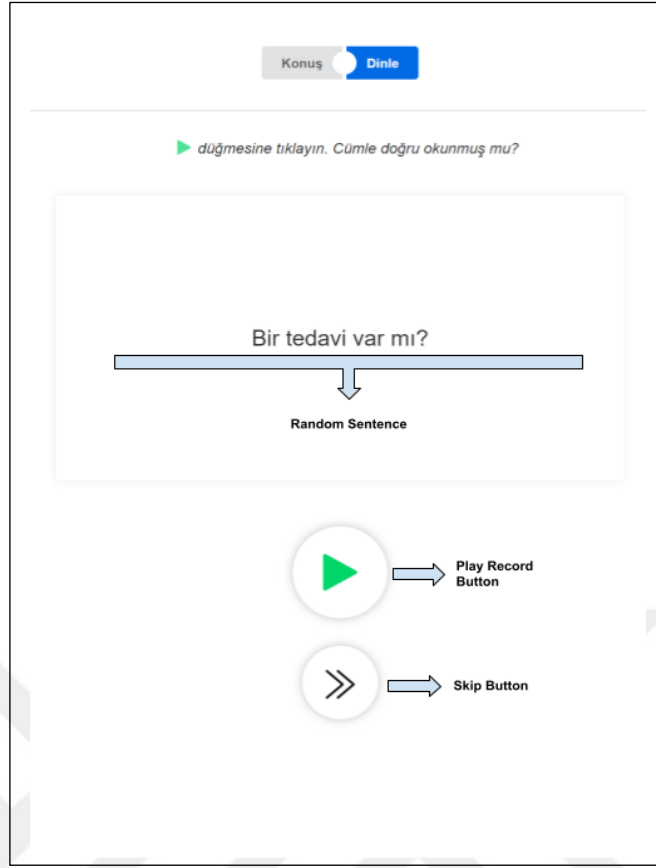


Figure 2.4 ArVoice Listening Tab.

The Listening tab returns a pre-recorded audio file to the user. Users can listen or skip the current sentence. After listening if it is okay, they can upvote the content. Otherwise, they can use the negative vote to it.



Figure 2.5 ArVoice Listening Tab – After Playback.

This SPA client application is written with React Native. React Native is a powerful, cross-platform application development framework.

ArVoice's backend consists of a microservice for REST API calls. This microservice is locally connected to a Cassandra database cluster that has three nodes. This microservice is written in Go programming language.

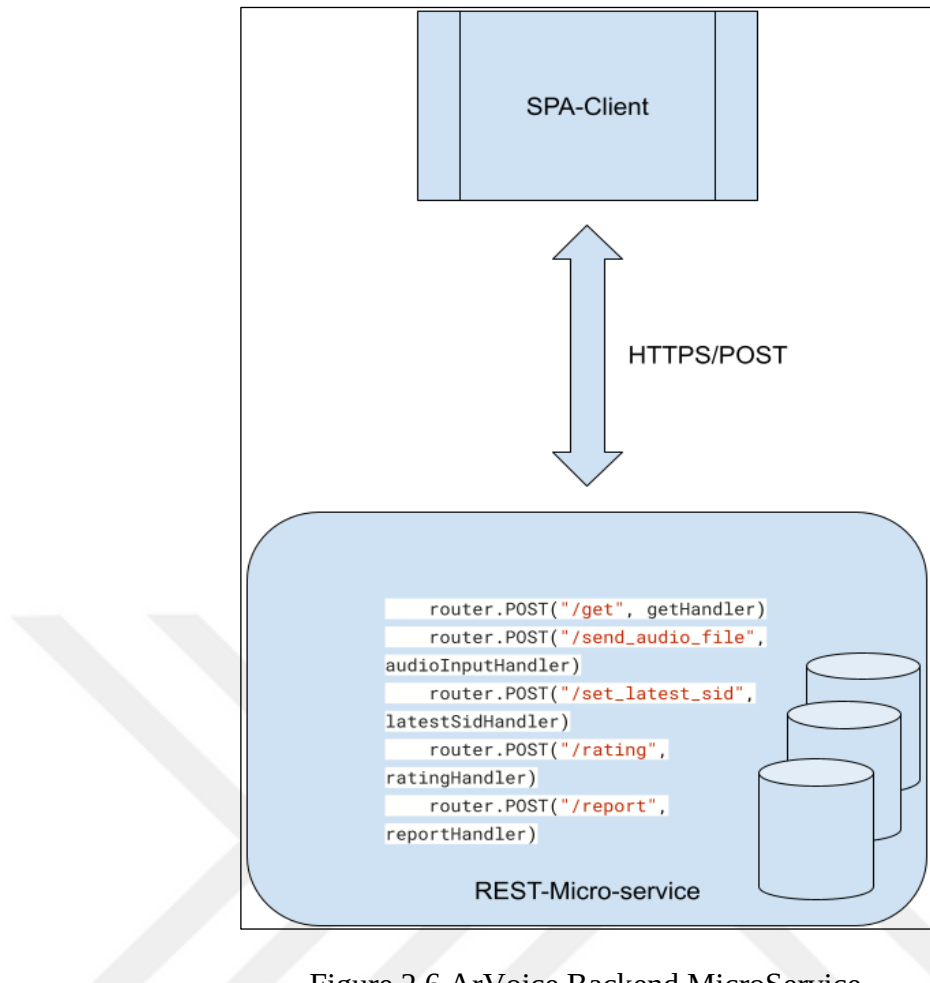


Figure 2.6 ArVoice Backend MicroService.

/get: It gets a random sentence or random voice clip according to the client's tab.

/send_audio_file: uploads recorded voice clip to database.

/set_latest_sid: After every recording on the client-side, this gets called to update latest_sid. This is done so that audioInputHandler can insert into the validation table the proper sentence id.

/rating: Calls updateRating after decoding the JSON response into its rating type and UUID.

/report: Increments the count for the number of reports for a given sentence with a UUID.

ArVoice random sentence database is contains 710326 unique sentences which uses tur_sentences [Tatoeba, 2021] database.

2.2. ArdicSRTCollector

ArdicSRTCollector is a Python library that extracts audio from YouTube videos and crops that audio according to its subtitle file. It generates cropped audio and text couples. That is a great source for speech-to-text training models.

The library can be installed on PyPI [Web 2, 2022] (The Python Package Index) [PyPI]. It uses FFmpeg [Web 3, 2022] to encode, crop, and convert video to audio files. It is fully open-source and available on GitHub [Web 4, 2022].

ardicsrtcollector [-h] [-sv SAVE_PATH] -ufp URL_FILE_PATH

To convert the Youtube URL to mp3 and srt file.

optional arguments:

- *h, --help* show this help message and exit
- *sv SAVE_PATH, --save_path SAVE_PATH*
- *ufp URL_FILE_PATH, --url_file_path URL_FILE_PATH*

The library can convert any YouTube videos which have an auto-generated or manual Turkish subtitle. It works flawlessly with manual written subtitles. One should be careful with using auto-generated subtitles. Some crops may have a little time-shift in them. We can fix it by tweaking crop parameters, but it affects manual subtitles too. That's why we should not forget to undo all settings to default.

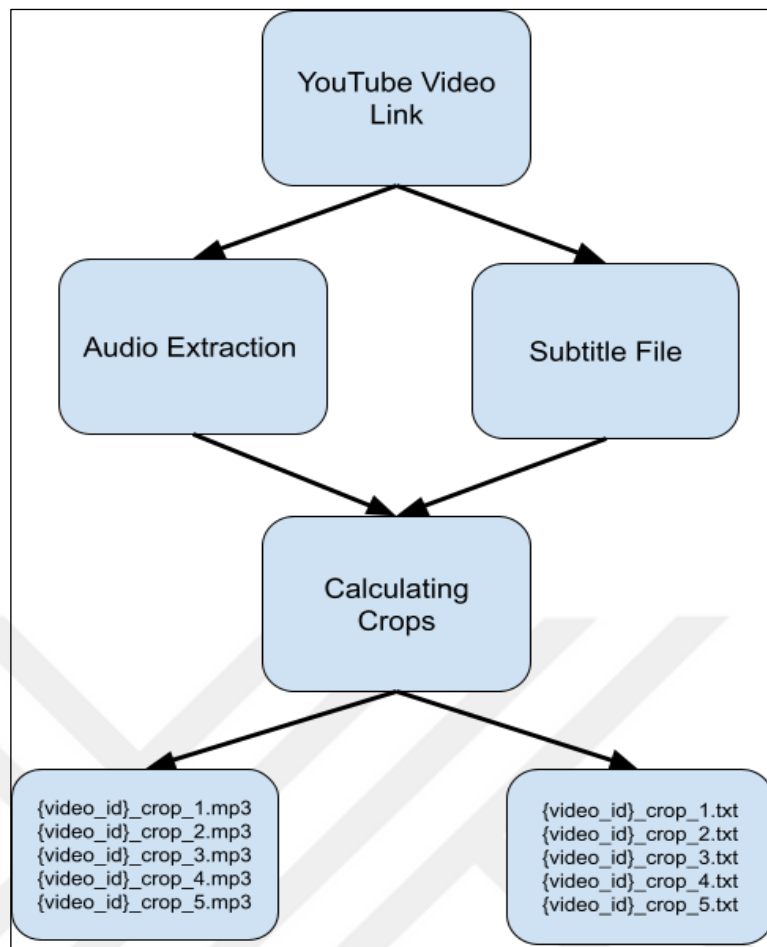


Figure 2.7 ArdicSRTCCollector YouTube Data Generation Flow.

3. DATA EMBEDDING

Word embedding is a way of representing text input with vectors. The goal is to make semantically, and synthetically similar words are also mathematically similar too. That gives us the base building block of a language model.

In this approach, every word is represented as n-dimensional vectors. Each dimension acts as a feature of that word. That's why semantically similar words must be vectorially similar to each other.

There are several word embedding techniques that we have used in this study. These are:

- Word2Vec
- GloVe
- FastText
- BERT

3.1. Word2Vec

Word2Vec is a vector embedding model algorithm that uses a neural network to learn word relations from a large corpus. It uses two new model architectures for learning distributed representations of words that try to minimize computational complexity. These models are Continuous Bag-of-Words Model (CBOW) and Continuous Skip-gram Model.

CBOW [Mikolov, 2014] is a log-linear classifier that classifies middle words using four succeeding and four preceding words. It uses continuous representations of the context.

The continuous Skip-gram [Mikolov and Sutskever, 2013] model is similar to CBOW, but it attempts to detect the same word elsewhere in the sentence.

These two models each use one hidden layer. CBOW predicts the word in the middle, while skip-gram predicts the context from the input word. Linear operations can be done on the resulting vectors.

3.2. Global Vectors for Word Representation (GloVe)

GloVe [Pennington et al., 2014] is an unsupervised learning algorithm that obtains vector representations for words. Training is performed on aggregated global word-word occurrence statistics from the corpus. It tries to calculate how many times a word is shown in the same context. Training GloVe is faster than Word2Vec because it can be easily parallelized. The training objective of GloVe is to learn word vectors such that their dot product equals the logarithm of the words' probability of co-occurrence. Since the logarithm of a ratio equals the difference of logarithms, this objective associates (the logarithm of) ratios of co-occurrence probabilities with vector differences in the word vector space. Because these ratios can encode some form of meaning, this information is encoded as vector differences as well. GloVe vector dimensions differ 50-300 according to training.

3.3. FastText

FastText [Bojanowski, 2017] is based on CBOW. When compared to other algorithms, FastText decreases training time and maintains performance. The previous algorithms Word2Vec and GloVe cannot produce word representations for out of vocabulary (OOV) words.

FastText model solves the issues mentioned above. FastText breaks a word in terms of n-grams instead of full words for feeding into a neural network, which can acquire the relationship between characters and pick up the semantics of words. This model also gives better results by having better word representation, primarily in the case of rare words. Facebook has presented pre-trained word embeddings for 294 different languages, trained on Wikipedia using FastText embedding on 300 dimensions, and utilized word2Vec skip-gram model with its default parameters.

3.4. BERT

Bidirectional Encoder Representations from Transformer (BERT) [Devlin, 2018] uses an attention mechanism to word representations. One of the key innovations of BERT is using bidirectional transformer blocks for creating language

models. The model can learn contextual relations between words or subwords in text. The transformer comes with two separate mechanisms; an encoder reads the input data and a decoder that predicts the result. BERT uses word masking in the corpus. 15% of tokens are masked randomly. The model attempts to predict masked words. That masked language model (MLM) helps to learn contextual representations. The percentage of word masking is important. If 100% of tokens are replaced with masks, then the model could not produce good token representations for non-masked words. Model just tries to optimize masked words on data; it doesn't focus on non-masked tokens.



4. SPEECH TO TEXT

Speech to text (STT), also known as speech recognition, is an NLP task that aims to convert human speech data to text sequence.

Before Deep Neural Networks (DNN), Hidden Markov Model [Nilsson, 2002] (HMM) based architectures were popular. DNN based models quickly outperformed the traditional models and are widely in use. DNN technique enables end-to-end automatic speech recognition (ASR) where all speech recognition components such as pronunciation, acoustic, and language model learn from a single place called the speech recognizer model. In the traditional methods of language models, acoustic components must be trained separately.

The first attempt at an end-to-end ASR was Connectionist Temporal Classification [Graves,2014]. The model is based on recurrent neural networks (RNN) and a CTC layer. This model learns pronunciation and acoustic models together, but it is not a highly effective language learning model due to conditional independence assumptions. CTC models can directly map speech acoustics to letters that may cause spelling mistakes due to pronunciation. Therefore, CTC-based approaches need a good language model to clean up transcripts.

Alternatively, current state-of-the-art approaches are attention-based models. Attention-based models do not have conditional independence and can learn all the components of a speech recognizer such as pronunciation, acoustic, and language model directly.

There are wide application areas of ASR, virtual assistant (Siri, Google Assistant, Alexa), in-car systems, language learning, home automation, video games, automatic subtitle generation, automatic translation, robotics, security.

4.1. Dataset

Several datasets were used in training the speech-to-text model. We combined open-source datasets and our dataset collector applications ArVoice and ArdicSRTCCollector.

4.1.1. Common Voice (ENG, TR)

Common Voice [Ardila and Branson, 2020] is an open-source dataset provided by Mozilla. It contains multilingual speech data donated by humans.

The Common Voice corpus is a massively-multilingual collection of transcribed speech intended for speech technology research and development. Common Voice is designed for Automatic Speech Recognition purposes but can be useful in other domains such as language identification. To achieve scale and sustainability, the Common Voice project employs crowdsourcing for both data collection and data validation.

4.1.2. ArdicSRTCCollector (YouTube Videos)

Through the development process, we have downloaded 1237 different Turkish YouTube [Web 5, 2022] videos. All of these videos cropped according to their subtitle timestamps. Thus, we got 817,020 audio-text pairs.

4.1.3. ArVoice

With the ArVoice [Web 6, 2022] project, we got 134 validated audio-text pairs. Data may be small in size, but we are sure that every part of the data is clean and of good quality as all data is validated by humans.

Table 4.1 Dataset List.

Dataset	Total Unique Clips	Size
Common Voice (TR)	36,441	4.3 GB
ArdicSRTCCollector (TR)	817,020	71 GB
ArVoice (TR)	134	40 MB
Common Voice (ENG)	75,879	65 GB

4.2. Model

Transformer, [Vaswani, 2017] contains a total of 12 layers. These 12 layers consist of a 6-layer encoder and a 6-layer decoder.

Attention mechanism in transformer neural networks replaces the recurrence. Thus, it greatly optimizes the operation performance. Because transformers do not suffer from long dependency issues. All attention operations can be done in parallel, unlike RNNs. The attention mechanism is a word-to-word operation. Attention runs dot products between word vectors to determine the strongest relation against each other. That mechanism provides a deep relationship between words and produces better results. Attentions can be stacked and run in parallel. The original transformer model runs eight attention mechanisms in parallel. Multiple attention provides a broader in-depth analysis of sequences, reduced calculation operations due to not using recurrence, can be run in parallel, reducing training time each attention block learns different features of the same sequence.

The encoder stack contains, input embedding, positional encoding (PE), multi-head attention, post layer normalization (PostLN), feed forward network (FFN).

Input Embedding: We can use any word embedding method here. For example, if we use skip-gram with 512 dimensions, it will produce word vectors that have a dimension of 512. Also, closer words will have higher cosine similarity because of skip-gram. Skip-gram provides information about words but vector representations don't have any information about their position on the sequence. Transformer fixes this problem with positional encoding.

Positional Encoding (PE): Instead of using a separate position vector, the transformer embeds the position information to the input embedding vector. The original transformer model [Vaswani, 2017] provides sine and cosine functions to generate different frequencies for positional encoding. For 512 dimensions of the word embedding vector:

$$PE_{(pos2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{512}}}\right) \quad (4.1)$$

$$PE_{(pos2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{512}}}\right) \quad (4.2)$$

Adding a positional encoding vector to the embedding vector is straightforward. Positional encoding(PE) is calculated using pre-calculated embedding word vector $y1(word)$.

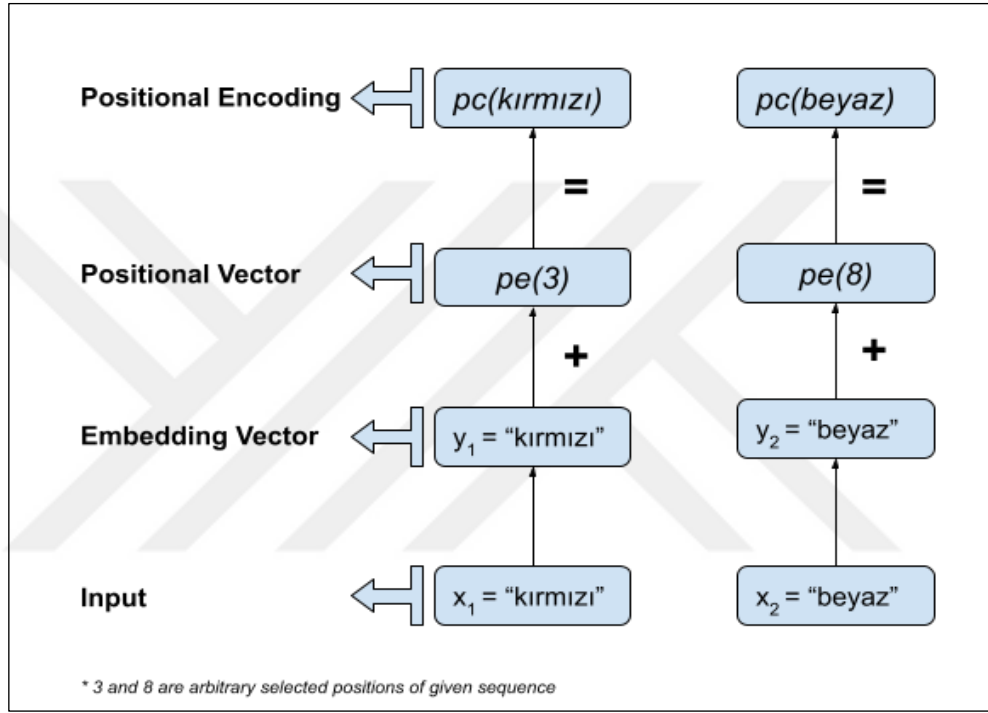


Figure 4.1 Calculating Positional Encoding.

Adding two vectors directly might cause losing information on the word embedding side. A positional vector might minimize it. Instead of using $y1$ directly, adding an arbitrary constant to $y1$ helps to keep word embedding information.

$$y1 = y1 * \sqrt{embedding_dim} \quad (4.3)$$

$$pc(kırmızı) = y1 + pe(3) \quad (4.4)$$

The output of positional encoding is the input of the multi-head attention sublayer.

Multi-Head Attention: The multi-head attention layer contains 8 heads and these heads are followed by PostLN. The “attention architecture” starts here. The input of the attention layer is a vector that contains the embedding and the positional encoding vector of each word.

The dimension of each word x_n of an input sequence is $d_{model_dim} = 512$:

$$pe(x_n) = [d_1 = 9.09297407e - 01, d_2 = 9.09297407e - 01, \dots, d_{512} = 1.00000000e + 00] \quad (4.5)$$

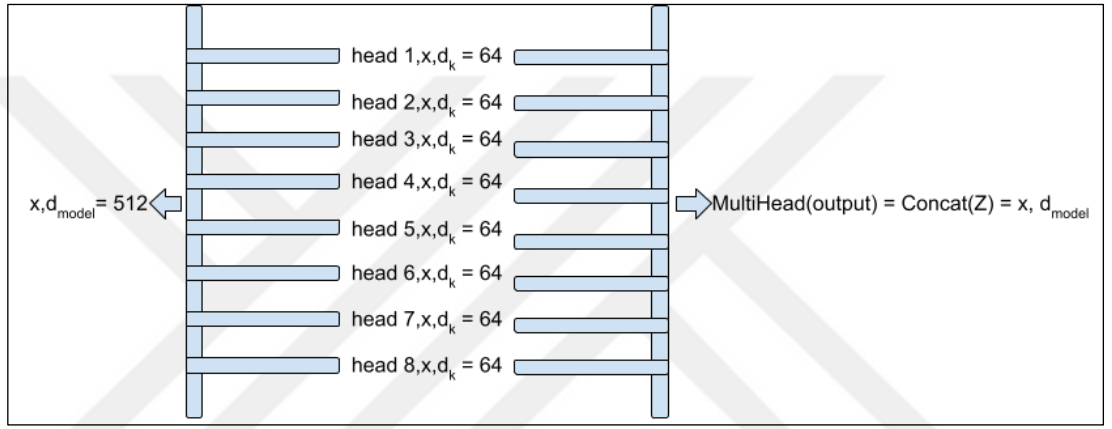


Figure 4.2 Multi-Head Representations.

These 8 heads can be run in parallel. The output of each head is a matrix z_i with a shape of $x * d_k$. The output of a multi-head Z is:

$$Z = (z_0, z_1, z_2, z_3, z_4, z_5, z_6, z_7) \quad (4.6)$$

All subparts of the Z are calculated parallelly and concatenated at the end.

$$MultiHead(output) = Concat(z_0, z_1, z_2, z_3, z_4, z_5, z_6, z_7) = x, d_{model} \quad (4.7)$$

Each head h_n of the multi-head attention, each word has three Q, K, V representations:

- Query vector (Q) has a dimension of $d_q = 64$, activated and trained when a word vector x_n seeks all key-value pairs of other vectors, including itself in self-attention.
- Key vector (K) has a dimension of $d_k = 64$ which will be trained to provide attention value.
- Value vector (V) has a dimension of $d_v = 64$ which will be trained to provide another attention value.

Attention is “*Scaled Dot Product Attention*” is represented in the following equation:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (4.8)$$

Post Layer Normalization: Each attention sub-layer and each FFN sub-layer followed by Post-LN. Post-LN contains an add function and a normalization process.

$$LayerNorm(x + Sublayer(x)) \quad (4.9)$$

$$LayerNorm(v) = \gamma \frac{v - \mu}{\sigma} + \beta \quad (4.10)$$

μ is the mean of v of dimension d :

$$\mu = \frac{1}{d} \sum_{k=1}^d v_k \quad (4.11)$$

σ is the standard deviation v of dimension d :

$$\sigma^2 = \frac{1}{d} \sum_{k=1}^d (v_k - \mu)^2 \quad (4.12)$$

γ is the parameter for scaling and β is bias vector.

Feedforward Network (FFN): FFN sub-layer between encoder and decoder are dense. FFN contains two layers and uses ReLU as an activation function.

The input and output of FFN are 512 but the inner layers are 2048. The output of FFN goes to Post-LN and connects to the multi-head attention layer of the decoder stack.

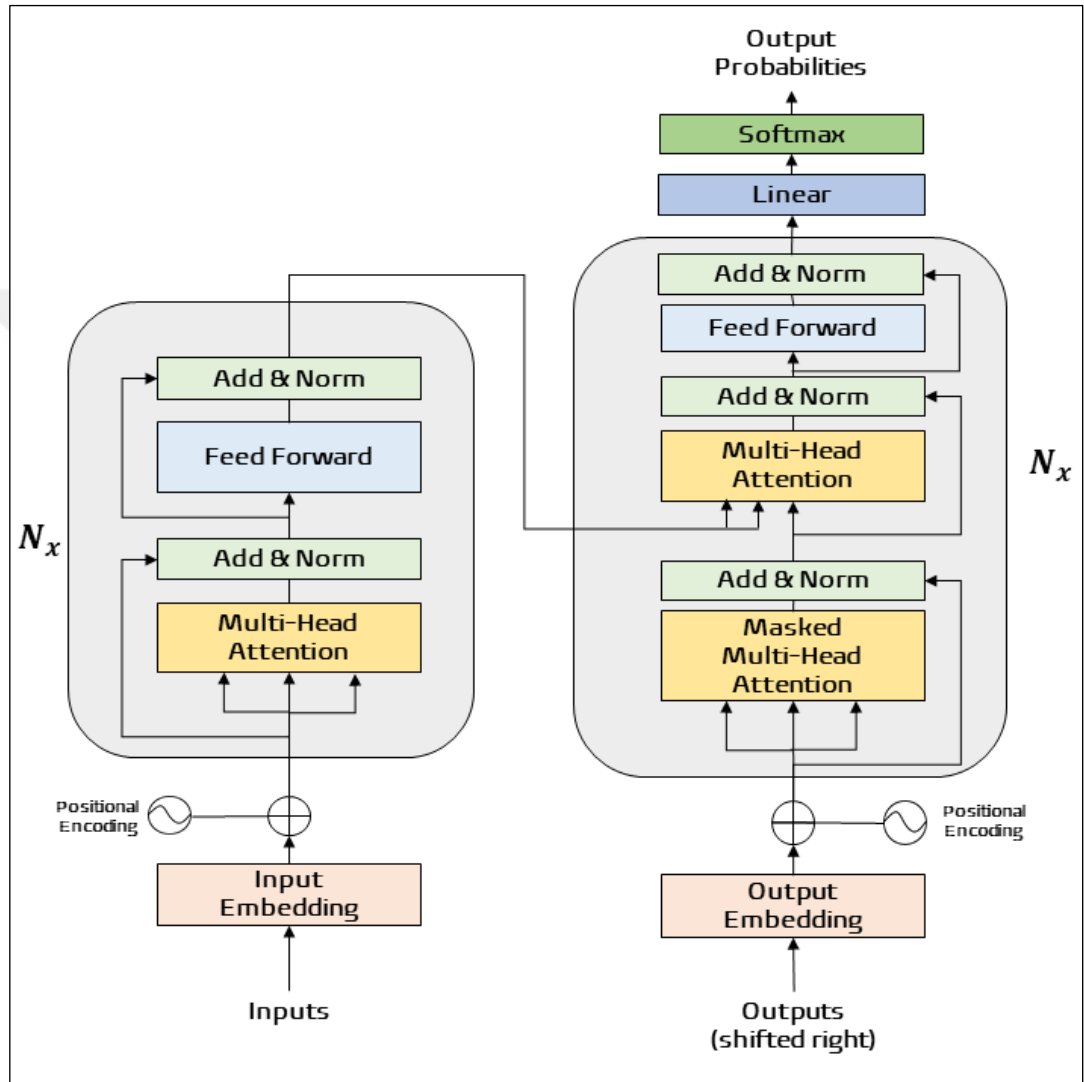


Figure 4.3 Encoder (Left) and Decoder (Right) stack.

The decoder stack contains, output embedding, positional encoding, masked multi-head attention, post-LN, feed forward network.

The structure of the sublayer of the decoder is nearly the same as the encoder. Output embedding and the position encoding are the same. The masked multi-head attention sub-layer-1 only lets attention apply to the positions up to and including the

current position. The future words are hidden from the Transformer, and this forces it to learn how to predict.

The multi-head attention sub-layer-2 also only attends to the positions up to the current position the Transformer is predicting to avoid seeing the sequence it must predict. FFN and Post-LN sublayers are similar to the encoder stack.

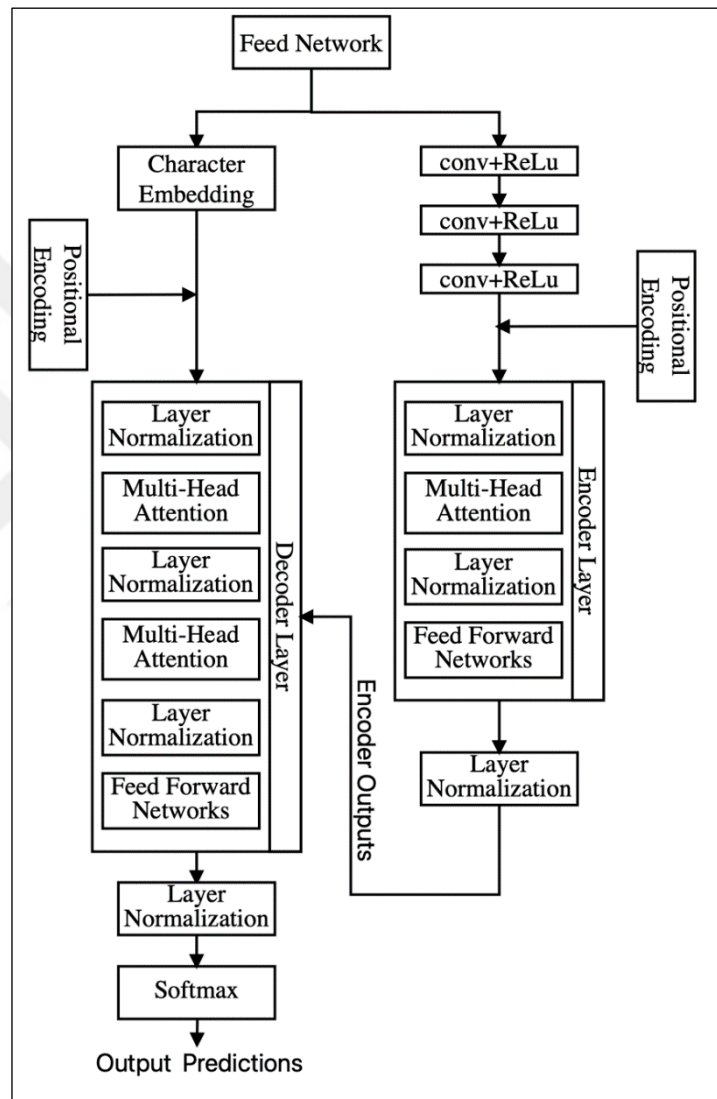


Figure 4.4 Transformer Model for Speech Recognition.

We need to preprocess YouTube data. This preprocessing contains converting Mp3 to WAV using FFmpeg, changing numbers with character representations, removing punctuations, spell correction and removing any speaker ID from text.

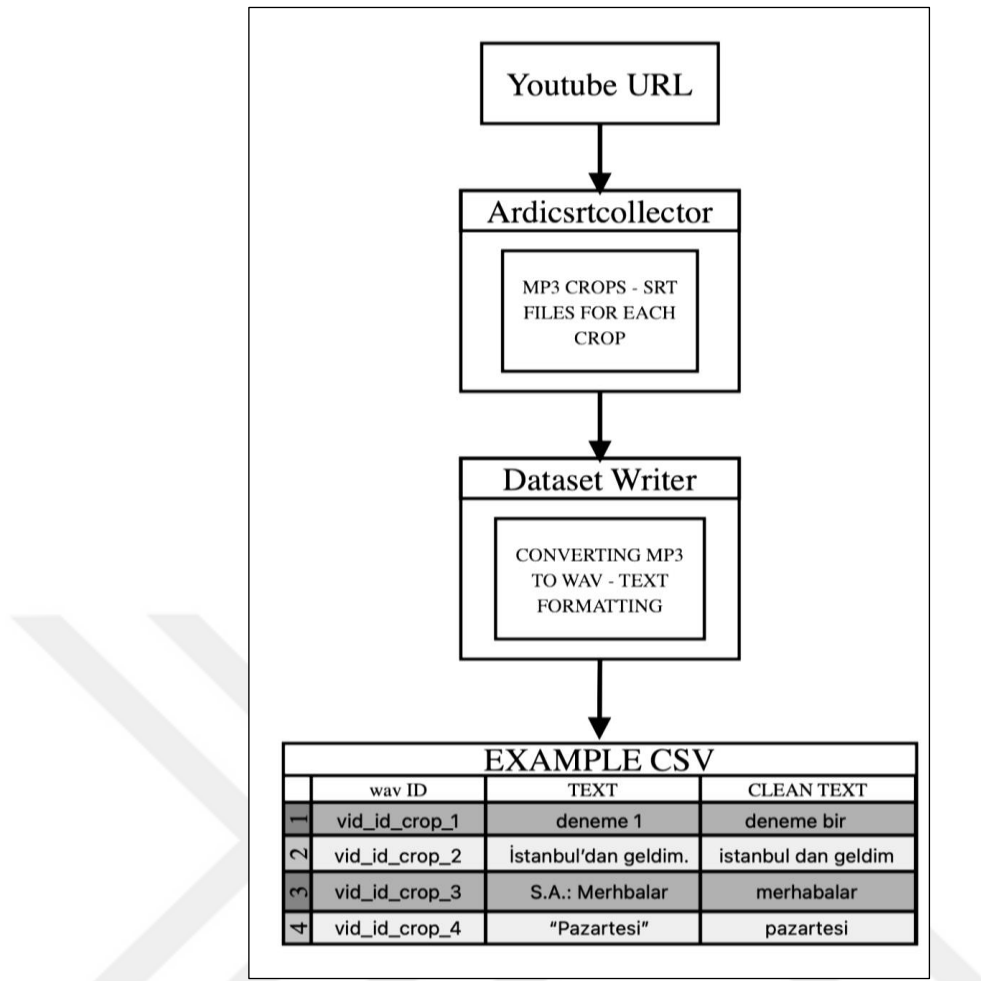


Figure 4.5 Dataset Preprocessing Steps.

4.3. Experiments on Speech to Text

We have used character error rate (CER) [Morris, Andrew, and Maier, 2004] and word error rate (WER) [Morris, Andrew, and Maier, 2004] as speech evaluation metrics.

4.3.1. Evaluation Metrics

Character Error Rate (CER): The character error rate is a common performance metric for ASR systems.

$$CER = \frac{(S + D + I)}{(S + D + C)} \quad (4.13)$$

- S is the number of character substitutions
- D is the number of character deletions
- I is the number of character insertions
- C is the number of correct characters

Word error rate (WER) is a good metric for evaluating the performance of an ASR system. The WER is derived from the Levenshtein distance [Miller, Frederic P, 2009], working at the word level instead of the phoneme level. The WER is a valuable tool for comparing different systems, as well as for evaluating improvements within one system. This kind of measurement, however, provides no details on the nature of translation errors and further work is therefore required to identify the main source(s) of error and to focus any research effort. This problem is solved by first aligning the recognized word sequence with the reference (spoken) word sequence using dynamic string alignment. Examination of this issue is seen through a theory called the power law that states the correlation between perplexity and word error rate. Word error rate can then be computed as shown in equation (14). Where S is the number of substitutions, D is the number of deletions, I is the number of insertions, C is the number of correct words, N is the number of words in the reference ($N=S+D+C$). This value indicates the average number of errors per reference word. The lower the value, the better the performance of the ASR system, with a WER of 0 being a perfect score.

$$WER = \frac{(S + D + I)}{(S + D + C)} \quad (4.14)$$

- S is the number of word substitutions
- D is the number of word deletions
- I is the number of word insertions
- C is the number of correct words

We have trained our ARSpeech model. We fine-tuned the Wav2Vec 2.0 model and used VOSK as a pre-built model.

4.3.2. Wav2Vec2

Wav2Vec2 [Baevski et al, 2020] is an end-to-end speech recognition model that uses labeled CTC [Graves et al., 2006] for loss.

The model uses GELU [Hendrycks and Gimpel, 2016] activation function on the feature encoder side. The total stride of the encoder determines the number of time-steps T which are input to the Transformer [Vaswani et al., 2017] block.

The output of the feature encoder follows the Transformer architecture. Instead of using fixed positional embeddings, a convolution layer acts like PE.

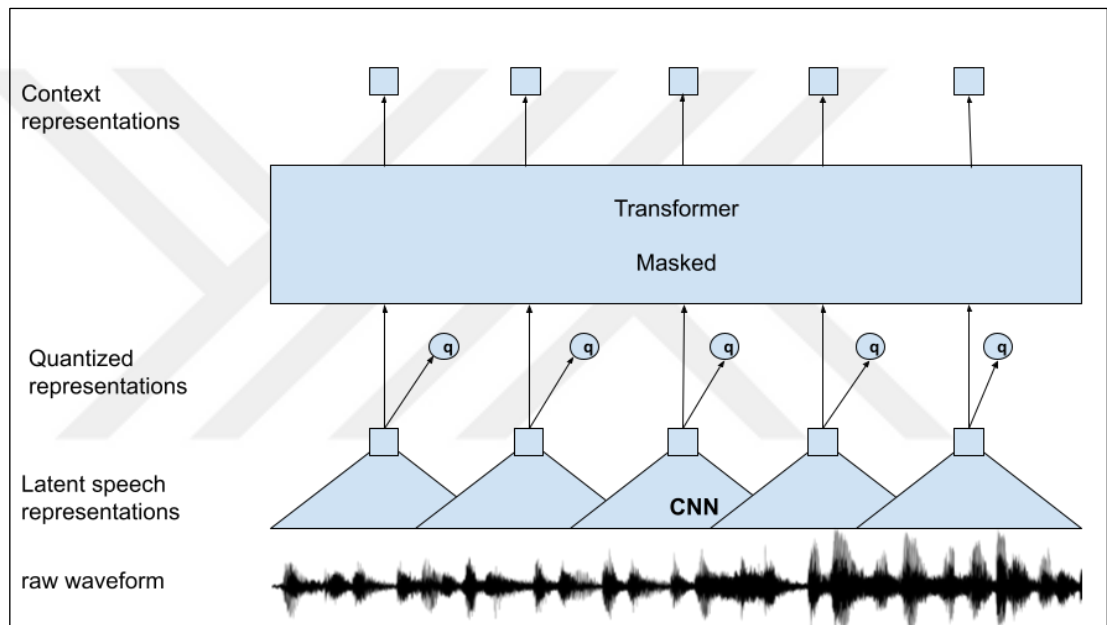


Figure 4.6 Wav2Vec2 Encoder Block.

Pre-trained wav2Vec2 models can be fine-tuned by adding a random linear projection top of the context network with changing vocabulary. Models are optimized by minimizing CTC loss.

Wav2Vec2 is implemented on Hugging Face [Web 7, 2022]. Models can be used pre-trained config or setting a custom config with Wav2Vec2Config class.

4.3.3. VOSK

Vosk is a speech recognition toolkit provided by the Alpha Cephei group. It has prebuilt models for 20+ languages including Turkish. It is running very efficiently

even with low-resource hardware such as Raspberry Pi. It can easily be used with different programming languages such as Java, C#, JavaScript. Portable models are around 50MB and very accurate. Besides these good specs, it is very hard to fine-tune the VOSK model. That's why we have used the prebuilt Turkish model for our benchmarks.

4.4. Speech to Text Results

We have benchmarked all test data which ARSpeech(ours), a finetuned model with our data (Wav2Vec2.0), and a pre-trained model (VOSK). The pre-trained VOSK model got the best results for our Turkish video dataset.

Table 4.2 Speech to Text Results.

Model	WER	CER
ARSpeech	0.861643	0.644888
Wav2Vec 2.0	0.705730	0.316047
VOSK	0.487085	0.239445

5. KEYWORD EXTRACTION

Words are the main phenomena that form a sentence. In terms of understanding the topics in a conversation, one can look at the important words in it. A good keyword extraction model effectively solves problems regarding, text summarization, tagging, indexing, and searching, text categorization.

Increased numbers of video conferences due to the COVID-19 pandemic make this speech and text processing ever so important. Automating the minutes of the meeting, summarizing a video session requires a good keyword extraction model.

In literature, there are several methods for keyword extraction such as TF/IDF [Spärck Jones, 1972], YAKE! [Campos, 2018] [Campos 2020], RAKE [Stuart et al., 2010] which are available online. Some shortcomings of these methods, which will be discussed in detail below, require a new method without these restrictions and shortcomings.

In general keyword extraction methods are statistical, machine learning based, and graph based. Statistical keyword extraction methods aim that word occurrences in documents. TF-IDF (Term Frequency – Inverse Document Frequency) is a good example method for statistical keyword extraction. It works well in common documents but in some professional text like physics or health, the keyword might only be seen once. Machine learning methods uses supervised and unsupervised techniques to describe keywords in the document. Supervised techniques try to optimize the model with a set of keywords. Unsupervised techniques aim to extract features from documents without using any training set or labels. Graph-based models are another good method for keyword extraction. Because graphs can keep a word relationship and information between other words. We can represent words as nodes(vertices), and we can represent relations and other information with links(edge).relationship and information between other words. We can use words as nodes(vertices) and relations and other information with links(edge). TextRank [Rada and Paul, 2004] and TopicRank [Adrien and Florian, 2013] are good example methods for graph based keyword extraction.

Last but not least, if a text is relevant to various topics, these algorithms extract word-wise keywords, but it is desired to represent keywords of topics separately.

As a result of these, in this study, a hybrid model which uses unsupervised learning and TF/IDF scores has been created.

The main contributions of this work are as follows:

- Instead of using raw content, we use word roots before calculating any scores
- According to word type, penalization of verbs and verb groups
- A naive dynamic cluster selection algorithm to select multiple clusters (topic-wise extraction)
- A benchmarking dataset labeled by humans

5.1. Dataset

We have two different datasets. One dataset is for training, calculating cluster origins, and IDF scores. The other is a human-labeled dataset for benchmarking the model's keyword hit and success rate.

Our training dataset is a combination of the Turkish dump of Wikipedia [Wikimedia TR dump, 2021], Kuzgunlar dataset [Kaggle, 2021], and Tur_sentences dataset [Tatoeba, 2021]. These datasets are not ready for direct use. They need to be pre-processed. We can summarize pre-processing steps as, removing stop words, digits, punctuations and extracting word roots.

Stop words are any words in a stop list that are filtered out before or after processing of natural language data[Rajaraman et al., 2011].

After these pre-processing steps, all sentences in the dataset are separated into words, and the word vector of the dataset is created. Then, the basic IDF formula is applied to the word vectors, treating all words as sentences:

$$W_{idf} = \log_{10} \left(\frac{\text{number_of_total_words}}{\text{number_of_existence_of_a_word}} \right) \quad (5.1)$$

Our benchmarking dataset contains 72 different kinds of YouTube video content in Turkish. We shared these videos with five different people. They tagged the videos according to their perspectives. We got their intersection and created the keyword set. Finally, we got 72 different videos tagged by five different people and a total of a thousand keywords to compare.

Table 5.1 Benchmark Dataset Samples.

Video Link	Name	Duration	Human Labelled Keywords
https://www.youtube.com/watch?v=W_IH8c5JZKE	Zaman Nedir? (What is time?)	14:57	zaman(time), teori(theory), newton, dünya(world), merkür(mercury), zihin(consciousness), einstein,güneş(sun), kanun(law), yasa(act), evren(universe), ışık(light), hız(speed), bilim(science), tren(train), deney(experiment), düşünce(thought), uzay(space), galileo, enerji(energy), hareket(movement)
https://www.youtube.com/watch?v=2cIGOUtuqEE	Doların İcadı (The invention of the dollar)	09:29	dollar(dollar), para(money), madeni(brazen), gümüş(silver), peso(pesos), simge(symbol), abd(usa), altın(gold), amerika(america), petrol(oil), enflasyon(inflation), bağımlı(dependant), üretmek(produce), üretim(production), deflasyon(deflation)
https://www.youtube.com/watch?v=7PGLODg3Xg4	İnsansı ve Üreyen Robotlar (Humanoid and Reproducing Robots)	10:47	robot, gelecek(future), dünya(earth), otomasyon(automation), insan(human), ürkütücü(scary), modern, teknoloji(technology), robotic(robotics), biyoloji(biology), kurbağa(frog), evrim(evolution), algoritma(algorithm), canlı(alive), molekül(molecule), yapay(artificial), zeka(intelligence), kök(root), hücre(cell), biyoteknoloji(biotechnology)

Table 5.2 Training Dataset.

Dataset	Sentence Count	Word Count	Unique Word Count
Kuzgunlar	50.098.586	488.796.556	5.710.315
Wikipedia TR	3.598.631	47.205.966	2.430.906
Turkish Sentences	713.344	4.972.279	939.193

5.2. Model

The model contains two stages:

- Estimation of IDF values;
- Clustering cleaned, fastText [Bojanowski et al., 2016] embedded vectors using K-Means Clustering Algorithm [in X., Han J., 2011].

On corpus before pre-calculating IDF scores, we are removing stop words, punctuations, and digits. After that, we are getting the roots of every unique word in the corpus. That allows us to process on a simplified dataset. Every word root has a unique score based on IDF also a word embedding representation on the database.

We have used several word embedding techniques. We got the best representations on fastText word embedding. Using these word embeddings we have trained an unsupervised model using K-Means Clustering Algorithm.

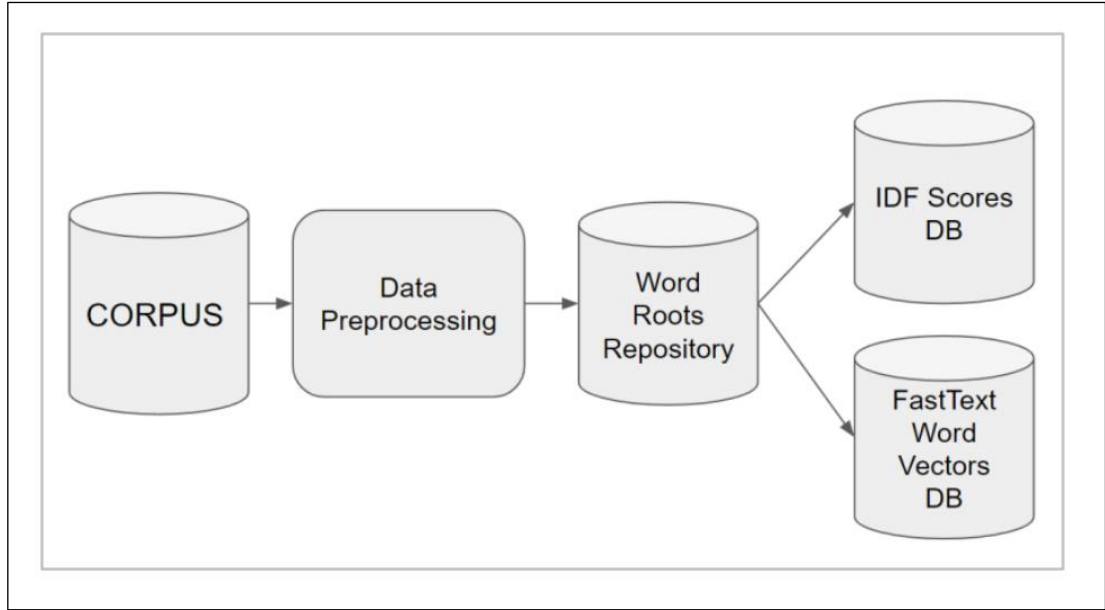


Figure 5.1 IDF Calculation.

FastText is a library for learning word embeddings and text classification created by Facebook's AI Research (FAIR) lab[Wikipedia, 2021]. We have used FastText for word embeddings because FastText handles OOV words extremely well. Every embedded vector has 300 dimensions. Instead of using raw word embeddings, we only used word embeddings of word roots. That gave us a well-separated clustering process.

5.3. Experiments on Keyword Extractions

GloVe [Pennington et al., 2014] is a word embedding model which has a vector size of 300 to represent a word. Referring to the results labeled GloVe in table 5.3, the best GloVe result was missing 81 keywords from the best FastText keyword result. The main reason for this is that GloVe does not represent new words other than the words in its dictionary. Therefore, word votes for selected clusters can only be determined by words from GloVe's own dictionary. Since the new vector representations are not known, the K-Means algorithm cannot predict the clusters to which the new words belong.

Word2Vec [Mikolov et al., 2013] is another word embedding model that we tried, with a vector size of 400 to represent a word. As we can see from table 5.3, the best results of word2vec are worse than GloVe when compared to the FastText. At the same time, as a result of the clustering of the vectors represented by the word2vec with

the K-Means algorithm, the dispersion did not result very well. Also, like GloVe, Word2Vec cannot represent new words.

We decided to use FastText when we saw the restrictions by GloVe and Word2Vec. In addition to being able to represent new words, FastText can also represent misspelled words with vectors close to them by associating them with the correct one. So, this was a very effective feature for our clustering model.

A grid search has been done for deciding the cluster number. We trained the system and changed the cluster size parameter between 50 and 1000. We got the best results at a cluster size of 150.

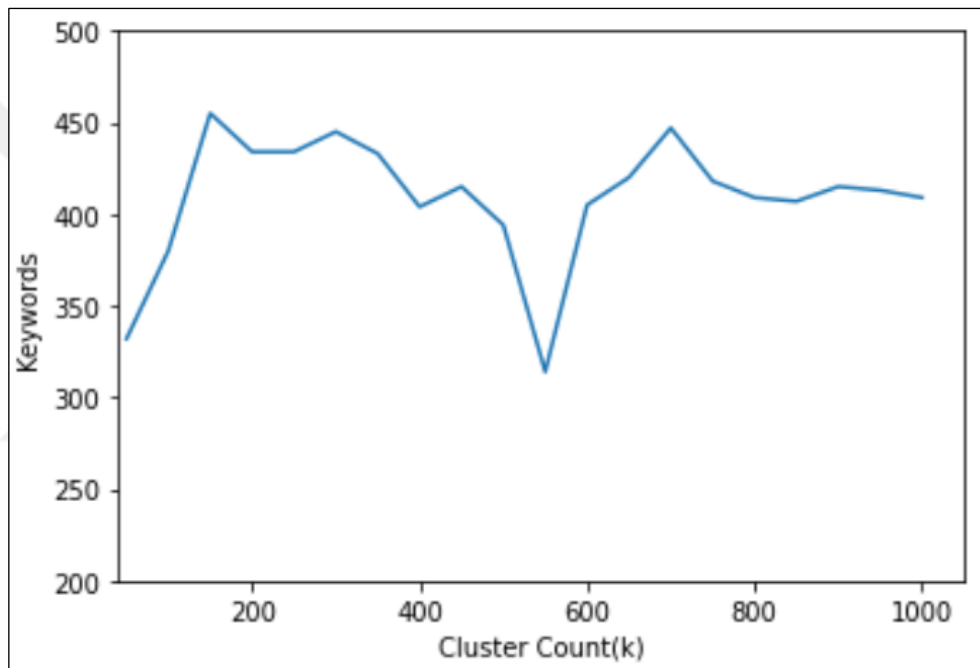


Figure 5.2 Cluster size grid search.

Due to the dataset's origin and content, verbs are disproportionately represented. We should not count common verbs in Turkish such as “gelmek” (to come), “durmak” (to stop), “gitmek” (to go), etc. as a keyword candidate. If any verb has been a candidate as a keyword we introduce a penalty score for decreasing the priority of it.

We tried two different stemmers: Morse [Akyürek et al., 2019] and Zemberek [Zemberek, 2021]. Zemberek runs faster for our case. We have used the Zemberek stemmer for getting word roots. Using word root instead of raw data gave us a significant advantage for calculating the priority of the same words with different

suffixes like, “algoritmaları, algoritmalarda, algoritmayı, algoritması” these all are counted as one keyword candidate as “algoritma(algorithm)”.

Input sequence votes for clusters by using their distance. We have a dynamic calculation method using standard distribution. This method detects how many candidate clusters should be selected for that input sequence based on the distribution amongst them.

If the votes are evenly distributed among the clusters (low variance case), it means that more than one topic has been mentioned in the conversation. The same is true for high variance. We need to set the threshold value so that the clusters in all the mentioned topics are not selected.

The threshold value is chosen to cover a certain percentage of all votes cast. For this, the logistic regression curve is used. When we test it on texts containing relatively more and fewer keywords, the variance value cannot be used to make comparisons between texts because it is affected by the size of the votes. Coefficient of variation (standard deviation/mean) was used for this. As a result of the trials, it was observed that this value changes between 0.5 and 2.

For high variance, we should choose most of the selected clusters by percentage, and for low variance, we should choose a few percentages. The logistic function where x is std/mean is used to make the selection,

$$threshold = \left(\frac{c}{1 + a * e^{-kx}} \right) + d \quad (5.2)$$

In the threshold equation;

- c is the maximum point of the curve
- a is logistics growth coefficient
- k is logistic function slope
- x is the standard deviation/mean ratio
- d is offset

After the selection of clusters, we selected all the words from chosen clusters except the verbs that are frequently used in daily life, which means the verbs with a low IDF score from the clusters. We mentioned that IDF scores were determined in

the dataset section. Based on these scores, we did not select verbs with a score less than 3.5, the midrange value of IDF scores.

5.3.1. TF/IDF

TF-IDF is a weight factor calculated by the statistical method that shows the importance of a term inside a text. Our algorithm is also based on the TF-IDF but for such a job and for Turkish, our algorithm is more comprehensive and more suitable than the original TF-IDF algorithm.

Term frequency[Luhn, 1958], Hans Peter Luhn, inverse document frequency[Spärck Jones, 1972] is the metrics that emerged with the work of Karen Spärck Jones.

In the TF-IDF algorithm, term frequency means the number of the existence of a particular word within the input text. Inverse document frequency can be calculated by taking the logarithm of the total number of documents divided by the number of documents in which a selected word occurs.

5.3.2. RAKE – Rapid Automatic Keyword Extraction

RAKE [Stuart et al., 2010] means Rapid Automatic Keyword Extraction. Initially, rake splits input text into a set of candidate keywords. According to these splits, RAKE can determine stopwords position and keyphrases. One of the advantages of the RAKE is, it can extract not solely keywords but also keyphrases. RAKE evaluates several metrics for calculating word scores based on word frequency $freq(w)$, word degree $deg(w)$, ratio of degree to frequency. Word that occurs in longer candidate keywords are favored by $deg(w) / freq(w)$.

5.3.3. YAKE! – Collection Independent Automatic Keyword Extractor

YAKE! [Campos, 2018] also splits input text into a set of candidate keywords. Afterward, assigns to all words a score according to their algorithm. These operations can make use of both the left side and the right side of the word. YAKE! can extract key phrases as well as keywords like RAKE. In YAKE!, a lower score corresponds to

the more important keyword. YAKE supports different sizes of text, different languages and domains. It uses an unsupervised approach to build features. YAKE has six main components, text pre-processing, feature extraction, individual terms score, candidate keywords list generation, data deduplication and ranking. Pre-processing steps splits text into individual terms. For feature extracting YAKE uses, casing, word positional, word frequency, word relatedness to context and word disjunctiveness. Casing assumes that relevant keywords often tend to concentrate more at the beginning of the document. Word frequency, checks the frequency of the word, score more those words that occur more often. Word relatedness to context, computes different terms that occur with the left side of the candidate keyword. Word disjunctiveness quantifies how often a candidate word appears in different sentences.

$$S(kw) = \frac{\prod_{w \in kw} S(w)}{TF(kw) * (1 + \sum_{w \in kw} S(w))} \quad (5.3)$$

$S(kw)$ is the score of candidate keyword, word score $S(w)$ is the candidate keywords first term which subsequent scores of the remaining terms. This divides the sum of $S(w)$ scores to average the keyword. $TF(kw)$ term frequency of keyword penalizes less frequent candidates. Lower $S(kw)$ score means important keyword.

5.3.4. KeyBERT

BERT, a bi-directional transformer model that allows us to represent phrases and words as vectors. There are many pre-trained BERT embedding models on Hugging Face [Web 7, 2022]. KeyBERT [Grootendorst, 2020] uses the BERT embeddings to extract relevant keywords and keyphrases from a text or a document. BERT has document-level embeddings as well as word-level embeddings. By using these two embedding methods and cosine similarity, the embedding vector of the entire document compares with the keyword or keyphrase embedding vector. According to this similarity, keywords or keyphrases are extracted.

5.4. Keyword Extraction Results

Table 5.3 Overall Results.

Algorithms	RAKE	YAKE	TF-IDF	ArKeyword
Keyword Hit Count	72	286	290	455
Percentage	7.20%	28.60%	29%	45.50%

Table 5.4 Keyword extraction results with different STT models.

Keyword vs STT	ArKeyword	RAKE	YAKE	TF-IDF
ARSpeech	143 / 15.21	16 / 1.70	71 / 7.55	68 / 7.23
Google STT	427 / 45.43	68 / 7.23	220 / 23.40	259 / 27.55
VOSK	342 / 36.38	34 / 3.62	191 / 20.32	236 / 25.11
Wav2Vec2	201 / 21.38	17 / 1.81	106 / 11.28	104 / 11.06

Table 5.5 Keyword results for word embedding method with relevant models.

Word Embedding Model	Speech Model	Overall Keyword Results
Word2Vec - 400D	ARSpeech	33.19% 312/940
Word2Vec - 400D	VOSK	30.96% 291/940
Word2Vec - 400D	Wav2Vec 2.0	16.06% 151/940
Word2Vec - 400D	Google Speech	33.19% 312/940
GloVe - 300D	ARSpeech	15.00% 141/940
GloVe - 300D	VOSK	32.23% 303/940
GloVe - 300D	Wav2Vec 2.0	20.43% 192/940
GloVe - 300D	Google Speech	36.81% 346/940
FastText - 300D	ARSpeech	15.21% 143/940
FastText - 300D	VOSK	36.38% 342/940
FastText - 300D	Wav2Vec 2.0	21.38% 201/940
FastText - 300D	Google Speech	45.43% 427/940

We got the overall best result on the FastText/Google Speech pair. For Word2Vec embedding, ARSpeech and Google Speech showed similar performance. For GloVe embedding, Google Speech showed the best performance. These results prove that FastText has a more proper representation feature than others for our clustering algorithm as explained in section 5.3.

To help clarify the diagram of our keyword extraction algorithm, the below graphs will concrete algorithm steps:

We handled a text about the James Webb space telescope [Web 8, 2022]. After the pre-processing and stemmer steps, the histogram of the remaining words can be seen in Figure 5.3.

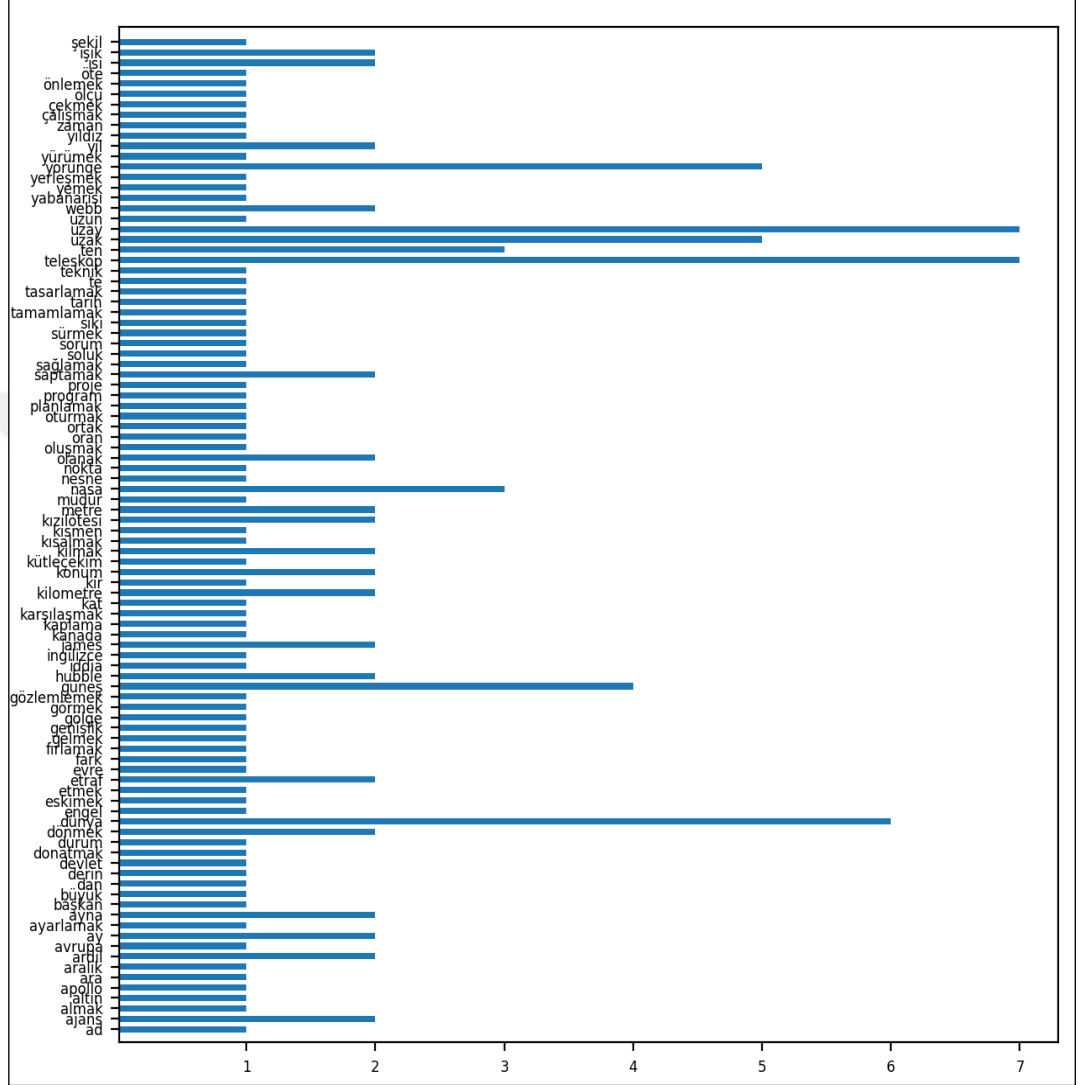


Figure 5.3 Histogram of the remaining input words.

We have a total of 96 unique words. After these words are represented according to the FastText embedding model, our K-Means model predicts by these representations which cluster each word vector belongs to or which cluster it is closer to. One point we would like to mention here is that, if we make the K-Means algorithm predict words one by one, there is a significant decrease in performance. Hence, we decided to have all the word vector representations predicted together, within a NumPy [Harris et al., 2020] array. Thus provided a significant performance improvement.

All remaining unique words voted for their cluster based on the term frequency multiplied by the square of the IDF score. The square of the IDF score is taken there because uniqueness has a more robust effect on cluster choosing. The results of the voting process can be seen in Figure 5.4:

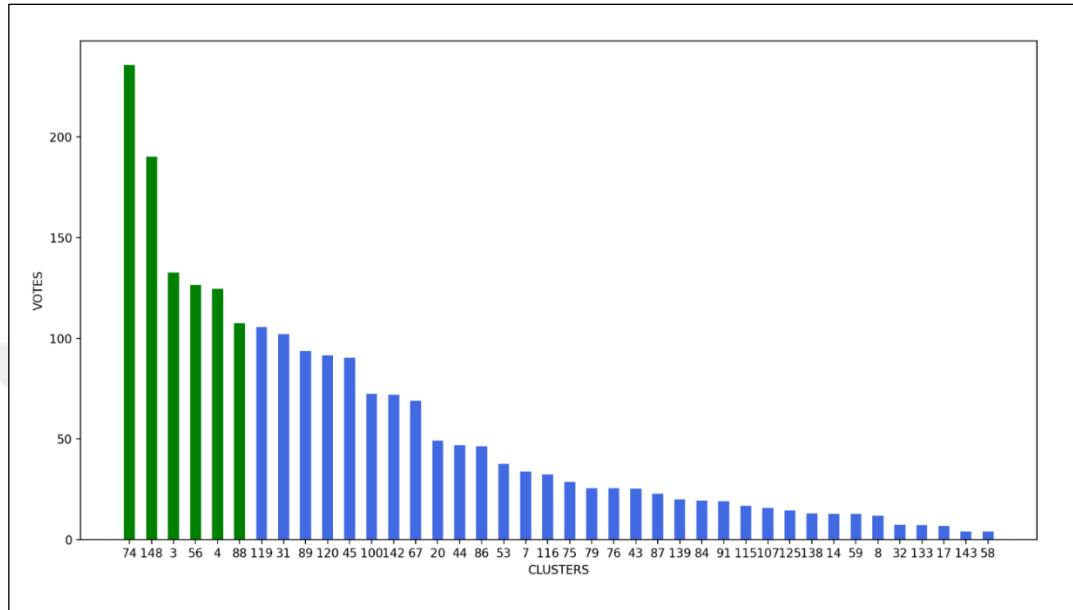


Figure 5.4 Total cluster votes.

In the experiments section, we mentioned the threshold parameter and how we determine it. In this case, the threshold value is equal to 0.3726291279381512. Therefore, the selected clusters will be the smallest representation greater than 37% of all clusters in descending order. This process can be seen in Figure 5.6.

Algorithm 1 Calculating Word Votes

Input: $Word_1 \dots InputWords_N$ **Input:** $VotesMap[]$ **Output:** $VotesMap$ (Words closest cluster count map)

```
1: function LOOP( $InputWords[]$ )
2:    $clusterId \leftarrow 0$ 
3:    $N \leftarrow length(InputWords)$ 
4:   for  $k \leftarrow 1$  to  $N$  do
5:      $clusterId \leftarrow ClosestClusterId(Word_k)$ 
6:      $VotesMap[clusterId] \leftarrow VotesMap[clusterId] + 1$ 
7:   end for
8:   return  $VotesMap$ 
9: end function
```

Figure 5.5 Calculating word votes.

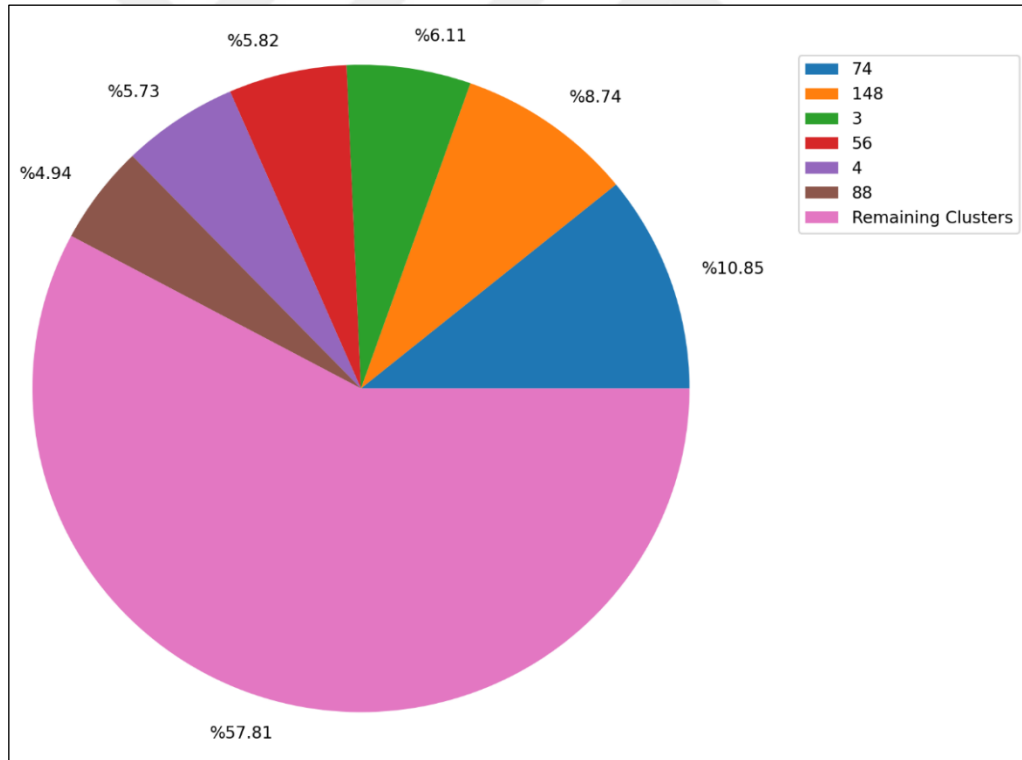


Figure 5.6 Distribution of the cluster votes.

As we can see from Figure 5.6, chosen clusters are 42.19% of the total clusters. Afterward, all the words inside chosen clusters are selected but applied to a limit for frequently used verbs in daily life. That is detailed in the experiments section.

As the last step, if word from the input is in chosen clusters, the term frequency is calculated according to the size of the word vector. Besides, the database has IDF scores and we have the origins of words as a map. If the word is a verb, we penalize the score of the word at a rate of 0.75. The vote of each word and also, the $tf * idf$ score with or without penalization can be seen in Figure 5.7:

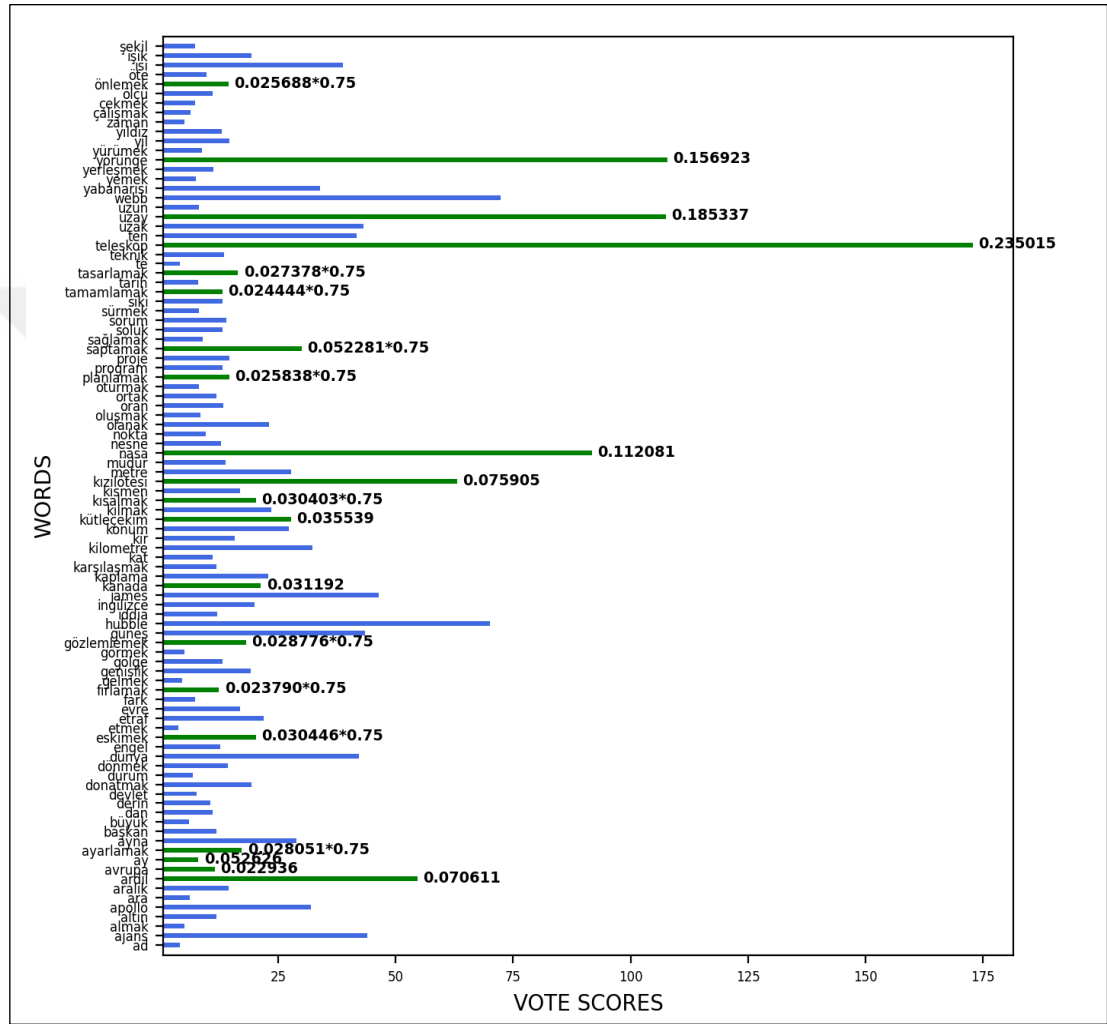


Figure 5.7 Votes and scores of the words.

By using these scores, keywords are extracted in order of importance. The higher a word's score, the more relevant the word is to the input text.

6. SYSTEM DEPLOYMENT

In the deployment step services are separated into four vital services. These services are in order of operation steps;

- API Gateway Service
- Speech-to-text REST API service,
- Keyword REST API service
- Lemmatization Service
- Model Inference Service

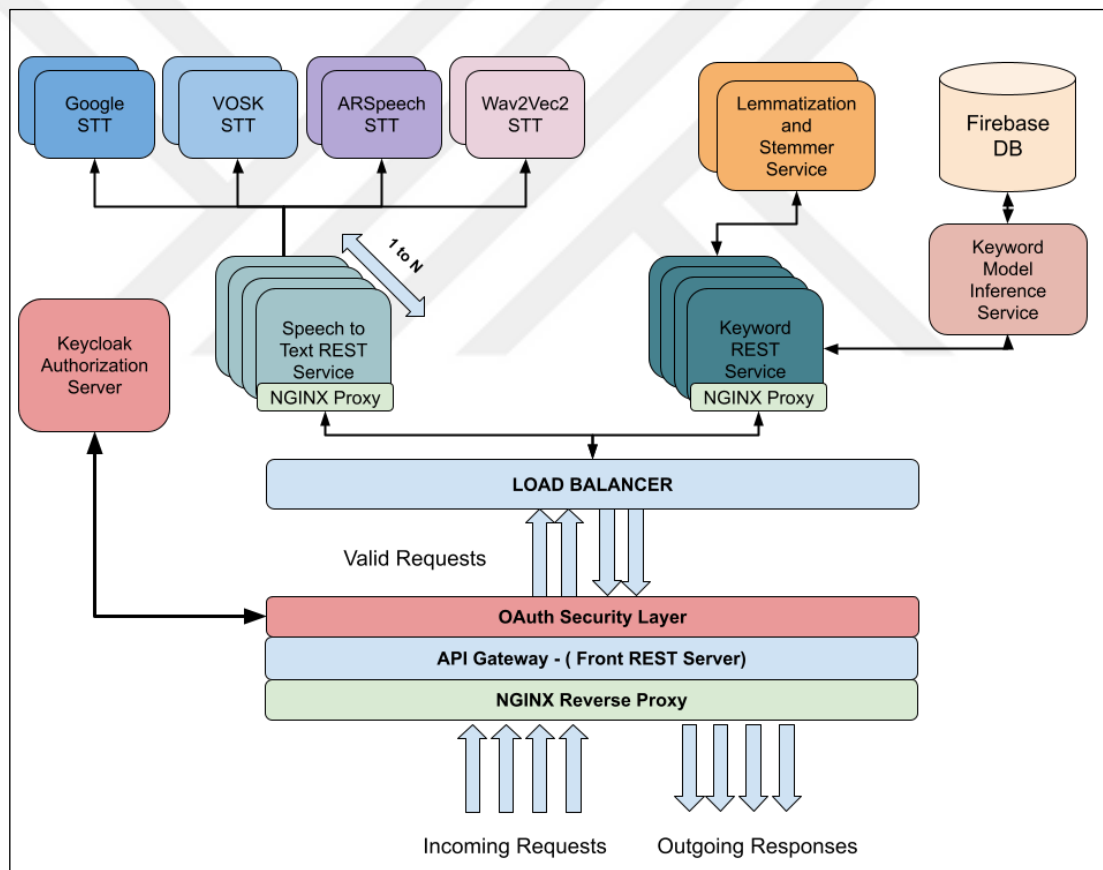


Figure 6.1 Deployed System Components.

As an end-to-end service, the input for keyword extraction service's input is provided by text-to-speech service. Docker swarm was used to increase the control and maintainability of the service scale.

The input can be in MP3, WAV, WEBM, or FLAC format, and the return value can be selected in JSON or text response format according to the user request. It is also possible to select which text-to-speech model the file will be converted to text with. The output of this service is provided as the input of the keyword extraction service.

The Zemberek service is used for lemmatization and finding the type of word. It is used to determine the score to be associated by finding the root of the word and determining whether the word is used as a predicate or a noun. For the sentence; “Dün arkadaşına bir hediye aldım.”, Zemberek's output can be seen in Fig 6.2.

```
[
  {
    "input": "Dün",
    "normalizedInput": "dün",
    "pos": "Noun",
    "analysis": "[dün:Noun,Time] Noun+A3sg",
    "morphemesLexical": "Noun+A3sg"
  },
  {
    "input": "arkadaşına",
    "normalizedInput": "arkadaşına",
    "pos": "Noun",
    "analysis": "[arkadaş:Noun] Noun+A3sg+P1sg+Dat",
    "morphemesLexical": "Noun+A3sg+P1sg+Dat"
  },
  {
    "input": "bir",
    "normalizedInput": "bir",
    "pos": "Det",
    "analysis": "[bir:Det] Det",
    "morphemesLexical": "Det"
  },
  {
    "input": "hediye",
    "normalizedInput": "hediye",
    "pos": "Noun",
    "analysis": "[hediye:Noun] Noun+A3sg",
    "morphemesLexical": "Noun+A3sg"
  },
  {
    "input": "aldım",
    "normalizedInput": "aldım",
    "pos": "Verb",
    "analysis": "[almak:Verb] Verb+Past+A1sg",
    "morphemesLexical": "Verb+Past+A1sg"
  },
  {
    "input": ".",
    "normalizedInput": ".",
    "pos": "Punc",
    "analysis": "[.:Punc] Punc",
    "morphemesLexical": "Punc"
  }
]
```

Figure 6.2 Sample output from Zemberek's lemmatization function.

After the audio file is transcribed, this transcription is cleaned and word roots are extracted. The response is forwarded to the Clustering Service. The Clustering Service returns the clusters of words in response, respectively. When “[dün arkadaş bir hediye almak .]” is given as an input to the Clustering Service, we get the information that the words belong to the clusters [125, 138, 20, 67,142] respectively.

As mentioned in the previous sections; cleaning the entry, voting, and finding the frequency of the words are performed with the Keyword Extractor service. The text data which will be used as input for the Keyword Extractor service is the output of the speech-to-text service.

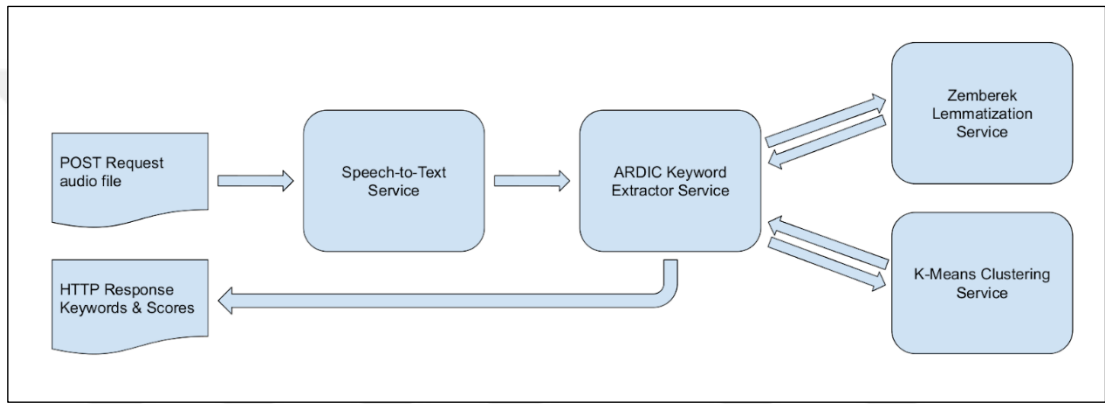


Figure 6.3 Architecture of services and communications between them.

For demonstrating secure HTTPS/POST communication and keyword mechanism, we developed this client application.

ARDIC Keyword Extractor

Input Text

Herkesin merak ettiği bir soru var Pandemi ne zaman ve nasıl bitecek Dünya daha önce birçok kez buna benzer hatta daha büyük sorunlarla karşılaştı İşte birazdan bu felaketlerin nasıl başladığını ve bittiğini göreceğiz Size öyle ilginç detaylar göstereceğim ki gördüğünüzde oldukça şaşıracağınızdan eminim Pandemi başladığı noktadan sonra eğer engellenemezse büyümeye devam ediyor Doğal olarak sosyalleşmeyi seven insanlık bunun büyümesine bir şekilde yardımcı oluyor Sosyalleşme ne kadar fazla olursa virüsün büyüme oranı ve doğal olarak ölen insan sayısı da artıyor Bu ölümler virüsten insan bağışıklığı değişkeninden ve çöken sağlık sistemlerinden dolayı meydana geliyor İnsanların çoğu yavaş yavaş enfekte oluyor Bazıları hayatlarını kaybederken bazıları bağışıklık sistemleri sayesinde hayatta kalıyor

Response time 370 ms. 1009/10000

Youtube Pandemi ▼

Get Keywords

Figure 6.3 Keyword Extractor Client UI.

Keyword Extractor client is a web application written in Flutter/Dart. It uses secure POST API calls to ArKeywordService. It takes raw text input to generate keyword sequences. Also with this client, we test the response time of our model in real-time.

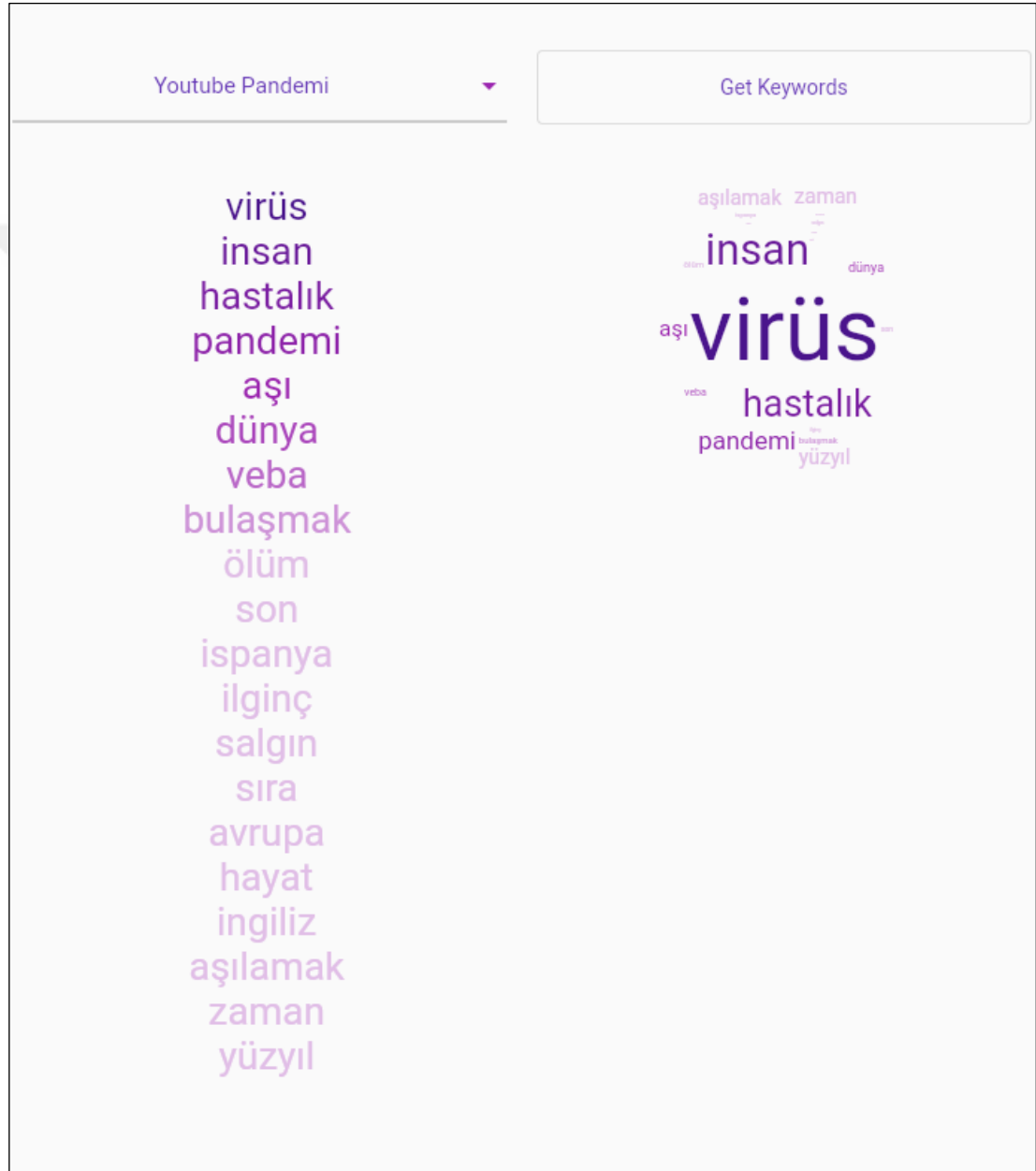


Figure 6.4 Keyword Extractor Client Raw and Word Cloud Responses.

ARSpeech Client Application: For end-to-end testing, we developed an STT client application. This application can generate nearly real-time keywords based on crops. Also, it has different configurations for several scenarios. For example, if a session started and we just want to get only current keywords system could calculate it near real-time. Another example might be we have some previous speeches in the same room with different kinds of topics. And we might want to get all keywords in it. The keyword extractor system allows to get keywords in given intervals.

In the example below, I have read Battle of Manzikert (Malazgirt Meydan Muhaberesi) and system-generated keywords step by step near real-time. STT applications cropped sound blobs every 15 sec which can be configurable on the settings screen and appended the incoming results as speech results. At the same time keyword service is triggered using this speech result.

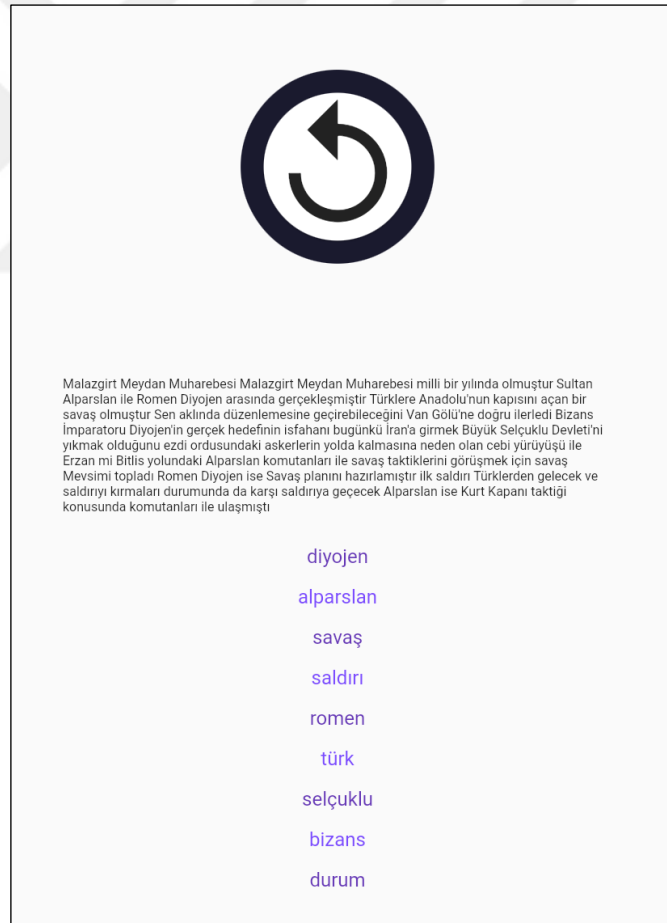


Figure 6.5 Real Time STT and Keyword Results.

7. CONCLUSIONS & FUTURE WORKS

We developed an end-to-end system that includes transcribing audio and extracting keywords over that text. All system components were developed as microservices that allowed us for easy scaling.

Training speech-based models takes so much time. That is a big restriction for tweaking with hyper-parameters. That's why our WER/CER scores are a little bit higher than we expected. Even so, the keyword model performed well.

We developed ArVoice for data labeling, but the usage of the website is not very high. Integrating the ArVoice with gamification and prizing system allows users to have fun and gain income at the same time.

Currently, our keyword model uses FastText embeddings. It is a good solution for handling OOV but still, it is not contextual. Using contextual embeddings may improve performance on synonym words.

Cluster selection algorithms hyper-parameters are set statically. Using dynamic parameters might improve performance.

BERT is a contextual word embedding algorithm, but it is giving different embedding a word for each sentence. Different BERT word embeddings for each sentence make word clustering impossible. Using a BERT-like semantic-based embedding technique will improve the overall performance.

Instead of K-Means, which approaches clustering problem very naively in our case, using HDBSCAN [Malzer, 2020] or other, similar and highly efficient hierarchical clustering algorithms might improve results even further. Since our word vectors are not spherically distributed, similar in distribution density in a low dimensional space, using any clustering algorithm straightaway might be problematic without inspecting the data first. Unfortunately inspecting a 300-dimensional vector proved to be hard and since most of the clustering algorithms struggle to meaningfully represent and cluster data in high dimensions; we suggest reducing dimensions (with t-SNE [van der Maaten, 2008] or UMAP [McInnes, 2018]) of word vectors, in information retaining way first and then run an appropriate clustering algorithm on dimensionally reduced word vectors. Since dimension reduction methods, such as UMAP, will only retain available information about known word vectors, OOV vectors might not belong to the same cluster.

One cannot easily make assumptions about topics and distribution among them. To give an example; should the sports category contain all sports-related terms or should be separate clusters for each sport there is? Such questions may be easy to answer for specific topics, but it is ambiguous for topics that we have no knowledge of or knew even existed. To make such clustering decisions easier, we suggest developing a keyword rating system for end-users.

Selecting a threshold value for cluster selection is also a non-intuitive process. Both in terms of selecting appropriate values for logistic function and of applying for different cluster count distributions. Despite the fact that low variance means roughly equal distribution of votes between clusters it doesn't contain any information about how many clusters have been voted to. It is highly important to realize that low variance with a high number of cluster selection means that all of the topics were equally important but to ensure small and concise output, only the most important words should be selected for given clusters. In a low variance and a low number of selected clusters scenario; the output should be optimized for giving all the clusters equal representation for the desired output size. This also applies to high variance situations.

7.1. Tags Done By Humans

Full human-tagged sample data can be seen in the table below. Fifty videos are tagged by five different humans. These results are the intersect of the tag of humans.

Video durations and video types are selected randomly. All videos are containing Turkish subtitle as well. We have used this subtitles to evaluate our speech to text benchmark results.

We tried to select all types of topics such as sports, politics, art, vlog, nature, computer science etc.

Table 7.1 Tags done by humans.

VIDEOS	VIDEO_NAME	TIME	KEYWORDS
https://www.youtube.com/watch?v=W_IH8c5JZKE	Zaman Nedir?	14:57	zaman, teori, newton, dünya, merkür, zihin, einstein, güneş, kanun, yasa, evren, ışık, hız, bilim, tren, deney, düşünce, uzay, galileo, enerji, hareket
https://www.youtube.com/watch?v=2cIGOUtuqEE	DOLARIN İCADI DOLAR NEDEN DÜNYANIN EN GÜÇLÜ PARASI?	09:29	dolar, para, madeni, gümüş, peso, simge, abd, altın, amerika, petrol, enflasyon, bağımlı, üretmek, üretim, deflasyon
https://www.youtube.com/watch?v=O_m7qtYMLH8	GEÇMİŞTEKİ PANDEMİLER NASIL BAŞLAMIS VE BİTMİŞTİ?	11:04	pandemi, virüs, bulaşmak, hastalık, çiçek, aşı, ölüm, ıspanya, grip, salgın, yayılmak, veba, avrupa, aşılama, insan, dünya
https://www.youtube.com/watch?v=lwzrGdjBRI8	AY'IN İKİYE BÖLÜNMESİ GERÇEKLERİ! BİLİMSEL KANITLAR VE NASA AÇIKLAMALARI	12:04	ay, mekke, islamiyet, peygamber, ayet, yarılmak, nasa, yüzey, çizgi, uydu, bilim, sihir, mucize, islam, inanmak, veri, cassini, yarmak, kıyamet
https://www.youtube.com/watch?v=Br04w4Gk4s	Suudi Arabistan Hakkında Bilmediğiniz	11:44	gökdelen, suudi, arabistan, arap, ezan, idam, kadın, erkek, yasak, turist, ceza, ülke

Table 7.1 Continued.

VIDEOS	VIDEO_NAME	TIME	KEYWORDS
https://www.youtube.com/watch?v=2XKzp0zZzQA	Eski Hacker'a Sorduk! - "Bilgisayarı Kapatsanız Bile Ben Hala Arka Planda Aktifim "	19:10	program, şifre, hacklemek, bilgisayar, para, indirmek, dosya, hack, hacker, kredi, kart, deep, web, veri, mahkeme, haram, grup, illegal, tövbe, site
https://www.youtube.com/watch?v=CcCbIKp3Pzc	Geleceğin Otomobili Nasıl Olmalı	12:37	elektrik, otomobil, gelecek, tasarım, araç, dijital, dairesel, düşünme, bmw, geridönüşüm, bulut, döngü
https://www.youtube.com/watch?v=7PGL0Dg3Xg4	İnsansı ve üreyen robotlar	10:47	robot, gelecek, dünya, otomasyon, insan, ürkütücü, modern, teknoloji, robotik, biyoloji, kurbağa, evrim, algoritma, canlı, molekül, yapay, zeka, kök, hücre, biyoteknoloji
https://www.youtube.com/watch?v=zjsW3gqurc	Bulut Bilişimi (Cloud Computing) Anlamak	11:07	bilgisayar, şirket, işlem, bulut, bilişim, google, yazılım, web, hizmet, internet, mobil, reklam, facebook, instagram, youtube, bilgi
https://www.youtube.com/watch?v=tBxUZrw_OLc	Yoksulluk ve Yardım Etme İkilemi Üzerine	08:45	çocuk, yardım, hayat, yoksul, yaşam, sorumluluk, deney, düşünce, ölmek, kaynak
https://www.youtube.com/watch?v=5j8A8YwxKf8	Lokum Gibi Arnavut Ciğeri Tarifi Sırrını VERİYORUM	08:02	yemek, tarif, ciğer, arnavut, hızlı, kolay, lezzetli, kızartma, kokusuz, piyaz, lavaş
https://www.youtube.com/watch?v=G6hn_Q680M	10° KARAVANDA KIŞ HAYATI - Webasto İşe Yarıyor mu ?	16:49	soğuk, kış, kamp, karavan, kar, yalıtım, çay, dağ, şelale, çamur, yürüyüş, mangal, webasto, ısıtma, ısıtıcı, yaban, güzel
https://www.youtube.com/watch?v=D1Uo7v0Sl1w	Türkiye'nin nükleer macerası: Akkuyu Nükleer Santrali hakkında her şey	17:16	nükleer, santral, akkuyu, mersin, iş, çevre, zarar, kanser, sağlık, enerji, elektrik, enerji, atık, turizm, inşaat
https://www.youtube.com/watch?v=6mQyq7B8yWU	Yapay zeka nedir? Yapay zekanın yaptığı inanılmaz şeyler!	08:23	yapay, zeka, bilgisayar, robot, canlı, gelecek, makine, öğrenme, komut, program, satranç, insan, tedirgin, kısıtlı,

Table 7.1 Continued.

VIDEOS	VIDEO_NAME	TIME	KEYWORDS
https://www.youtube.com/watch?v=hGZDLa8u15c	ahmet nur çebi - sergen yalçın görüşmesinin perde arkası	24:15	spor, direktör, futbol, takım, istifa, görüşme, kadro, futbolcu, sezon, sakat, genç, galibiyet, mağlubiyet, ayrılık, sergen, yalçın, beşiktaş
https://www.netflix.com/watch/81083005?trackId=13752289&tctx=0%2C0%2C0d630a4be5dc ee58a5803b60b376 ad8728983fd9%3Ab859a83438f36fd2 eb453ca1e866240a c49dd756%2C0d630a4be5dcee58a58 03b60b376ad8728 983fd9%3Ab859a8 3438f36fd2eb453c a1e866240ac49dd7 56%2Cunknown% 2C%2C%2CtitlesR esults	ZORBA NASIL OLUNUR? - Gerçekleri Manipüle Et	45:00	stalin, diktatör, bilgi, kontrol, medya, açlık, sovyet, güç, halk, komünist, parti, savaş
https://www.youtube.com/watch?v=WzNA02gjcPE	Kodlayarak Nasıl Sanat Yapabiliriz?	09:41	tablo, görsel, kuş, ses, sanat, fırça, kod, eser, bilgisayar, sanatçı, yazılım, açık, kaynak, tuval, kodlamak
https://www.youtube.com/watch?v=JPGqc5OzjuY	Hiçbir Taksici Benim Gibi Davranmamış (İsmail Küçükkaya)	13:48	taksi, muharrem, ince, haber, sabah, fox, paris, fransız, fransa, avrupa, kafa, muhabbet, ismail, küçükkaya
https://www.youtube.com/watch?v=wEJfcNbnw7c	İlber Ortaylı - Polonyalıların ve Almanların Türkler hakkındaki görüşleri	07:05	Polonya, Türk, Türkiye, mülteci, kriz, Suriye, göçmen, ülke, Alman, tarih, Trabzon, savaş, dünya, akademik, gaz, oda
https://www.youtube.com/watch?v=uB1GX6LP0l0	Geçinemeyenler: "Kölelikten daha kötü bir durumdayız" [7. bölüm]	11:18	çocuk, çalışmak, pazar, kız, maaş, emek, meslek, kadın, okumak, baba

Table 7.1 Continued.

VIDEOS	VIDEO_NAME	TIME	KEYWORDS
https://www.youtube.com/watch?v=St4bC9fcfA	Richard Stallman - Özgür Yazılım	13:39	Bilgisayar, emir, program, Sirket, özgürlük, yazılım, kod, kullanıcı, geliştirici, Microsoft, Windows, güç, gnu, linux, sunucu, web, mühendislik, donanım, zor, okul, kaynak
https://www.youtube.com/watch?v=bA9vScR3E0Q	DİPLOMASIZ YAZILIMCI OLUP İSVEÇ'TE İŞ BULMAK VE TAŞINMAK	13:49	göteborg, anna, isveç, antalya, üniversite, türkiye, mali, müşavir, muhasebe, ingilizce, şirket, yazılım, google, youtube, başvuru, dil, bölüm, karın, yardımcı
https://www.youtube.com/watch?v=3HThqtVx_Y	Kış Şartlarında Araç Kullanma Tekniklerinin 10 KURALI - Kışın Araba Sürmenin 10 Kuralı	08:50	araç, kış, lastik, basınç, hava, kar, buz, yokuş, yol, faydalı, cam, antifriz, far, yolculuk, görmek
https://www.youtube.com/watch?v=5Jkm_B1JNrc	Taksidi Bile 4-3-3 Aklıma Geliyor Yılmaz Vural KAFA TAKSİ'de!	18:02	Yılmaz, Vural, kafa, taksi, ofsayt, futbol, galatasaray, fenerbahçe, maç, Erol, evlenmek, anneanne, İstanbul, hoca
https://www.youtube.com/watch?v=3rXPVw_qYqE	KAFAYA TAKMAMA SANATI: NASIL BAŞARIRIM?	11:31	düşünce, hayat, odaklanmak, kafa, takmak, mevzu, beklenti, haber, evlenmek, kapı, düşünmek, duygu, konu, korku
https://youtu.be/TwW95gmYR0s	ŞEKER DOST MU DÜŞMAN MI?	11:01	şeker, zarar, bilim, karaciğer, tatlı, metabolizma, zaaf, kola, glikoz, meyve, fruktoz, meşrubat, doğal, doktor
https://youtu.be/Oo5PretbU0I	Gülten Dayıoğlu: "Çocuk kalmasaydım"	23:25	çocuk, yazar, öğretmen, aile, gülten, dayıoğlu, kitap, okumak, dergi, edebiyat, yetenek, anne, okul, heves
https://youtu.be/9ABaEFhLDII	Devlet Neden Var, Ne İşe Yarar?	22:56	felsefe, maslow, siyaset, sorumluluk, toplum, platon, politik, Hobbes, devlet, Rousseau, kriz, ihtiyaç, mülk, sahip

Table 7.1 Continued.

VIDEOS	VIDEO_NAME	TIME	KEYWORDS
https://youtu.be/g5PuceJKMak	Cache Friendly Programming Teknikleri	47:45	işlemci, bilgi, erişme, çekirdek, mimari, hafıza, okuma, yazma, derleyici, optimizasyon,
https://youtu.be/GBqgEgJTXHM	Açık Mimariler ve Mimari Tasarım Kalıpları	01:20:25	mimari, açık, tasarım, kalıp, gereksinim, katman, protokol, dağıtık, otonom, mikroservis, yazılım
https://youtu.be/Xf5VUvJTiwg	An'da Mutluluk	26:40	mutluluk, serdar, kuzuoğlu, arzu, beyhan, budak, danışan, hayal, geçmiş, fark, kıyas, sosyal, medya, karşılaştırma, an, yaşamak
https://youtu.be/P4B_7Esn1A	Uyku Düzenimizi Değiştirmek Bizim Elimizde!	09:25	bilimsel, çalışma, uyku, düzeni, başarı, meslek, avantaj, pozitif, gece, üretken, ruh, hali, sabah, doktor, sosyal
https://www.youtube.com/watch?v=HcV-bVOxXyY	Elon Musk Neden 35 Metrekarelik Bir Evde Yaşıyor?	10:51	Elon, Musk, zaman, para, ev, tasarruf, hedef, fiyat, lüks, tercih, süre, hayat, fabrika, musk
https://www.youtube.com/watch?v=rueGKU5rbS0	MUTLU İNSANLARIN YAPTIĞI 3 FARKLI ŞEY	05:20	hayat, Martin, Seligman, özellik, zorluk, baş, olay, ilişki, iyimserlik, öğrenilmiş, başarısız, hata, insan, başarı, kötü
https://www.youtube.com/watch?v=T6uyFl8Kd8k	Bu 5 Şeyi Yapan İnsanlar Kendi Kendine Yetiyor	08:14	kendi, yetmek, özgüven, duygu, ilgi, yalnızlık, yalnız, kırılmak, şefkat
https://www.youtube.com/watch?v=efoEm9mq-SQ	Tuna Tavus Belgeseli: Tuna Tavus İçin 'Sıradan Bir Gün' @Tuna Tavus	13:55	tuna, tavus, yemek, vücut, kilo, dambıl, spor, instagram, youtube, öfke, idman, kahvaltı, omlet, powerlift, guru, twitch, parçalamak, video
https://www.youtube.com/watch?v=mbJVBabNbTo	Trata: Gezici Restoran	12:57	restoran, yemek, deneyim, doğa, yeni, tat, kasaba, ortam, mutfak
https://www.youtube.com/watch?v=psVq6ilNv8I	Biraz Ada Biraz Caz: Bozcaada Caz Festivali	13:23	ada, bozcaada, caz, müzik, müzisyen, doğaçlama, tür, deniz, rüzgar, motivasyon, özgürlük, vapur

Table 7.1 Continued.

VIDEOS	VIDEO_NAME	TIME	KEYWORDS
https://www.youtube.com/watch?v=C0HafLzUT4A	Moda (Semt Olan)	14:50	moda, kitle, sokak, taksim, cihangir, mahalle, kadıköy, elit, insan, yeldeğirmeni, barış, manço, komşu, cihangir, taşınmak, yeşil, metrobüs, marmaray, istanbul, mekan, özgürlük, ankara, kültür
https://www.youtube.com/watch?v=4hggU7H5ISk	Mutluluğunu Kendin Engelliyor Olabilirsin	08:31	mutluluk, kıyaslamak, farklı, insan, hata, ruhsağlığı, çevre, dost, kötü, iyi, hissetmek, duygu, sorun
https://www.youtube.com/watch?v=N9oJUuxVukE	Pereira İle Yollar Ayrıldı: Ali Koç'un Önünde 2 Yol Var	11:53	futbolcu, futbol, ali, koç, başkan, yönetim, fenerbahçe, derbi, beşiktaş, takım, hoca, stoper, forvet, santrafor, şampiyon, oyuncu, maç, irfan, eski
https://www.youtube.com/watch?v=RnR_K82Sm64	Kontrast - Araba Tamircisi 21. Bölüm Amerika vs Türkiye	13:50	araba, tamir, meslek, çalışmak, işçilik, fiyat, iş, amerika, türkiye
https://www.youtube.com/watch?v=1wQBqCjDYvA	Kontrast - Eski Mahkûm AMERİKA VS. TÜRKİYE 13. Bölüm	22:29	hapis, hapishane, cezaevi, new, york, kürtçe, dil, silah, suç, sokak, okumak, korku, değiştirmek, geliştirmek, muhalif, örgüt, komün, mahkum, telefon, tecrit, muhalefet
https://www.youtube.com/watch?v=4uSk6uA8ZGs	Kontrast - Tır Şoförü 19. Bölüm Türkiye vs Amerika	09:00	tır, şoför, iş, meslek, hayal, yol, araç
https://www.youtube.com/watch?v=AlamULMxsi0	Hayatını ve Enerjini Organize Etmenin Yolları	10:36	enerji, potansiyel, odaklanmak, dikkat, dağılmak, telefon, bilgisayar, uygulama, zihin, insan, küçük, güç, dinlenme, hayat
https://www.youtube.com/watch?v=QftpzM70cGI	Ruhsal Yorgunluk Nasıl Geçer?	06:39	his, ruh, yorulmak, insan, suç, başarı, yaşam, hayat, ihtiyaç, dinlenmek, kendi, yorgun, yormak, empati
https://www.youtube.com/watch?v=2P0TSE_30ts	Kriz Anında Sakin Kalma Yolları Beyhan Budak Klinik Psikolog	11:01	psikolog, psikoterapi, kriz, zihin, uzaklaştırmak, kaygılanmak, nefes, alarm, korku, yavaşlamak, duygu, olay, yürüyüş, uyku, insan, sakın

Table 7.1 Continued.

VIDEOS	VIDEO_NAME	TIME	KEYWORDS
https://www.youtube.com/watch?v=7ZNF3qqVaQA	Verimli Çalışma İçin Dört Mükemmel Yöntem (Kişisel Gelişim Videoları)	04:15	çalışmak, verim, iş, dikkat, vakit, odaklanmak, fayda, teknik, hedef
https://www.youtube.com/watch?v=tZSPk7kD2yI	Antrenman Programım Haftalık spor rutinim	21:36	antrenman, supplement, program, rmb, kickboks, pazartesi, koşu, idman, snach, kuvvet, juisu, protein, kas
https://www.youtube.com/watch?v=OachmS1dm_s	Alanında iş bulamayanlar: Elektrik ve elektronik mühendisi	08:51	nüfuz, fen, elektrik, elektronik, hacettepe, üniversite, lise, sektör, savunma, sanayii, firma, hedef, başvurmak, istihdam, mezun,
https://www.youtube.com/watch?v=V7Vq-_uWvgg	Alanında iş bulamayanlar: İnşaat mühendisi	06:49	üniversite, inşaat, okul, mühendis, mezun, kariyer, tecrübe, asker, geliştirmek, yetiştirmek,
https://www.youtube.com/watch?v=MSf0YD2Zwa4	İngiltere'de Eğitim Almanın Önemi JANE (Evgiina Shaeva) ile söyleşi	15:16	moskova, amerika, rüya, burs, rus, rusça, başvuru, ıspanya, norveç, mülakat, ingilizce, avrupa, deneyim, eğitim, ingiltere, barselona, londra, üniversite, yurtdışı, işletme, fiyat, staj, pahalı
https://www.youtube.com/watch?v=QQDN-YazbIQ	Kontrast - Taksici 24. Bölüm Türkiye vs Amerika	10:15	taksi, araç, farklı, kazanç, ücret, müşteri, insan, yolcu, taksici, vergi, uber, lift, aile, dersim, mutlu
https://www.youtube.com/watch?v=qS8T17z2540	Gol Sevinçlerinde Yaşanan İlginç Olaylar	10:47	gol, sevinç, sergen, araba, parmak, maç, kutlamak, zemin, top, arkadaş
https://www.youtube.com/watch?v=OGE6xQ-SAXo	Kontrast - Afgan Mülteci 20. Bölüm Amerika vs Türkiye	11:09	afganistan, türkiye, memleket, gurbet, savaş, pakistan, dil, akraba, yardım, amerika, avrupa
https://www.youtube.com/watch?v=C49RDmwfIUQ	Kontrast - Berber 12. Bölüm AMERİKA VS TÜRKİYE	12:14	memnun, berber, tıraş, saç, sakal, tanımak, müşteri, hastalık, olumsuz, meslek, esnaf, sendika, kurum, para, türkiye, çırak

Table 7.1 Continued.

VIDEOS	VIDEO_NAME	TIME	KEYWORDS
https://www.youtube.com/watch?v=33CQd8PXNlo	Kontrast - 1. Bölüm Evsizler AMERİKA VS. TÜRKİYE (Altyazı dilini seçmeyi unutmayınız!)	12:00	uzaklaştırma, evsiz, sokak, tartı, adana, milyoner, milyarder, belediye, araba, yardım, hanım, pepsi, hayat, para, kitap, ırk, din
https://www.youtube.com/watch?v=wJlJWOJdgdQ	İnsanın Fabrika Ayarları: Tükenmişlik Sendromu Sinan Canan DenizBank Deniz Akademi	16:03	tükenmişlik, tükenmek, sendrom, zihin, psikoloji, yorgun, dikkat, beyin, pil, dinlenme, durmak, sinirsel, durum, şebeke, kafa, kabul, kabullenmek, sabır, hayat, motivasyon, ödül, sosyal, destek, insan
https://www.youtube.com/watch?v=nWNY2i3FX14	İnsanın Fabrika Ayarları: Beslenme Sinan Canan DenizBank Deniz Akademi	15:40	beslemek, hastalık, sağlık, insan, yiyecek, açlık, ata, fizyoloji, yemek, uzman, tıbbi, fazla, temel, vücut, karbonhidrat, yağ, protein, şeker, glikoz, tüketmek, gıda
https://www.youtube.com/watch?v=Rcp8kZKvKhU	Google'da çalışmanın dezavantajları	12:02	google, mühendis, madalyon, dezavantaj, teknoloji, proje, takım, şirket, hata, büyük
https://www.youtube.com/watch?v=tLhugizmHdQ	SoruYorum - Yetersizlik Kaygısı / Sosyal Fobi	04:58	geçmiş, gelecek, beyin, stres, amigdala, simülasyon, heyecan, sinyal, beklemek, konuşmak, sonuç, anksiyete, performans, gerginlik, fobi
https://www.youtube.com/watch?v=OtfHOVNNtfY	SoruYorum - Hayat Amacı / Öz	05:26	varoluş, konu, kitap, eğitim, sistem, cesaret, sorun, genç, hayal, anlatmak
https://www.youtube.com/watch?v=3f6q85q3xgA	INSTAGRAMDA SATILAN ÇAKMA AIRPODS	10:40	cihaz, kulak, şarj, instagram, araba, mikrofon, sıkıntı, ses, kalite, duymak
https://www.youtube.com/watch?v=Sle2LW8uGM	En Hızlı Köpüren Sabun Deneyi - Barış Özcan'ın Ses Desteğiyle	06:50	deney, bilim, sabun, hidrojen, potasyum, peroksit, oksijen, bakteri, bileşik, iyodür, mezür, gıda, boya, madde, sıvı, çözmek, katalizör.öğrenmek, eğlenmek

Table 7.1 Continued.

VIDEOS	VIDEO_NAME	TIME	KEYWORDS
https://www.youtube.com/watch?v=aVypVBT7HYY	Görünmezlik Deneyi Yaptım	04:44	deney, gözükme, cam, gliserin, malzeme, beher, kırılma, hava, ışık, hava, indis, kırılmak, yansımak
https://www.youtube.com/watch?v=nzEFWdFzDo	Çok Şaşıracığınız Yanlışlıkla Keşfedilmiş 5 Buluş	12:15	keşif, keşfetmek, cips, amerika, patates, kızartma, mısır, gevrek, bakteri, alexander, fleming, küflenmek, penisilin, percy, mikrodalga, yapıştırmak, yapıştırıcı, spencer, mühendis
https://www.youtube.com/watch?v=ABRQA-G_1M8	Tuzdaki Bomba	05:21	sodyum, metal, patlayıcı, tuz, deney, hidroksit, klor, zehir, patlamak, özellik, bileşik, hidrojen, tepkime, madde, tehlike

7.2. Overall Results Keyword Scores of Each Video

Table 7.2 Overall keyword results of each benchmark video.

Videos / Rate of Algorithms	RAKE	YAKE	TF-IDF /	ArKeyword Extractor
Zaman Nedir?	23.81	23.81	38.10	42.86
DOLARIN İCADI DOLAR NEDEN DÜNYANIN EN GÜÇLÜ PARASI?	26.67	66.67	60.00	66.67
GEÇMİŞTEKİ PANDEMİLER NASIL BAŞLAMIS VE BİTMİŞTİ?	31.25	50.00	50.00	81.25
AY'IN İKİYE BÖLÜNMESİ GERÇEKLERİ! BİLİMSEL KANITLAR VE AÇIKLAMALARI	0.00	5.26	5.21	47.37
Suudi Arabistan Hakkında Bilmediğiniz 27 İNANILMAZ GERÇEK	33.33	58.33	50.00	58.33
Eski Hacker'a Sorduk! - "Bilgisayarı Kapatsanız Bile Ben Hala Arka Planda Aktifim"	10.00	20.00	25.00	55.00
Geleceğin Otomobili Nasıl Olmalı	0.00	0.00	8.33	41.67
İnsansı ve üreyen robotlar	10.00	10.00	15.00	20.00
Bulut Bilişimi (Cloud Computing) Anlamak	18.75	37.50	43.75	68.75
Yoksulluk ve Yardım Etme İkilemi Üzerine	30.00	30.00	40.00	60.00
Lokum Gibi Arnavut Ciğeri Tarifi Sırrını VERİYORUM	27.27	18.18	18.18	18.18
10° KARAVANDA KIŞ HAYATI - Webasto İşe Yarıyor mu ?	17.65	17.65	11.76	17.65

Table 7.2 Continued.

Videos / Rate of Algorithms	RAKE	YAKE	TF-IDF /	ArKeyword Extractor
Türkiye'nin nükleer macerası: Akkuyu Nükleer Santrali hakkında her şey	0.00	33.33	40.00	60.00
Yapay zeka nedir? Yapay zekanın yaptığı inanılmaz şeyler! İNSANLIK 2.0	6.67	26.67	26.67	46.67
AHMET NUR ÇEBİ - SERGEN YALÇIN GÖRÜŞMESİNİN PERDE ARKASI	0.00	17.65	17.65	29.41
ZORBA NASIL OLUNUR? - Gerçekleri Manipüle Et	16.67	16.67	33.33	66.67
Kodlayarak Nasıl Sanat Yapabiliriz?	6.67	53.33	40.00	53.33
Hiçbir Taksici Benim Gibi Davranmamış (İsmail Küçükkaya)	0.00	21.43	42.86	28.57
İlber Ortaylı - Polonyalıların ve Almanların Türkler hakkındaki görüşleri	0.00	37.50	25.00	50.00
Geçinemeyenler: "Kölelikten daha kötü bir durumdayız" [7. bölüm]	10.00	10.00	10.00	40.00
Richard Stallman - Özgür Yazılım	9.52	23.81	23.81	33.33
DİPLOMASIZ YAZILIMCI OLUP İSVEÇ'TE İŞ BULMAK VE TAŞINMAK	0.00	15.79	26.32	42.11
Kış Şartlarında Araç Kullanma Tekniklerinin 10 KURALI - Kışın Araba Sürmenin 10 Kuralı	0.00	33.33	33.33	60.00

Table 7.2 Continued.

Videos / Rate of Algorithms	RAKE	YAKE	TF-IDF /	ArKeyword Extractor
Takside Bile 4-3-3 Aklıma Geliyor Yılmaz Vural KAFA TAKSİ'de!	0.00	28.57	42.86	28.57
KAFAYA TAKMAMA SANATI: NASIL BAŞARIRIM?	7.14	14.29	14.29	42.86
ŞEKER DOST MU DÜŞMAN MI?	7.14	28.57	28.57	64.29
Gülten Dayıoğlu: "Çocuk kalmasaydım bu kadar eser olmazdı"	0.00	42.86	42.86	64.29
Devlet Neden Var, Ne İşe Yarar?	0.00	14.29	14.29	35.71
Cache Friendly Programming Teknikleri	18.18	09.09	09.09	27.27
Açık Mimariler ve Mimari Tasarım Kalıpları	0.00	18.18	27.27	36.36
An'da Mutluluk	12.50	6.25	6.25	12.50
Uyku Düzenimizi Değiştirmek Bizim Elimizde!	0.00	13.33	13.33	20.00
Elon Musk Neden 35 Metrekarelik Bir Evde Yaşıyor?	0.00	14.29	21.43	42.86
MUTLU İNSANLARIN YAPTIĞI 3 FARKLI ŞEY	13.33	26.67	26.67	46.67
Bu 5 Şeyi Yapan İnsanlar Kendi Kendine Yetiyor	0.00	11.11	11.11	33.33
Tuna Tavus İçin 'Sıradan Bir Gün'	0.00	22.22	33.33	44.44
Trata: Gezici Restoran	11.11	22.22	22.22	55.56
Biraz Ada Biraz Caz: Bozcaada Caz Festivali	16.67	33.33	41.67	58.33

Table 7.2 Continued.

Videos / Rate of Algorithms	RAKE	YAKE	TF-IDF /	ArKeyword Extractor
Lastik Pabucun Anatomisi: Sneaker Sevdası	10.00	30.00	30.00	50.00
Moda (Semt Olan)	0.00	17.39	34.78	34.78
Mutluluğunu Kendin Engelliyor Olabilirsin: 9 Sebep	7.69	7.69	0.00	38.46
Pereira İle Yollar Ayrıldı: Ali Koç'un Önünde 2 Yol Var	0.00	26.32	31.58	47.37
Kontrast - Araba Tamircisi 21. Bölüm Amerika vs Türkiye	11.11	33.33	33.33	33.33
Kontrast - Eski Mahkûm AMERİKA VS. TÜRKİYE 13. Bölüm	0.00	14.29	14.29	33.33
Kontrast - Tır Şoförü 19. Bölüm Türkiye vs Amerika	0.00	14.29	14.29	57.14
Hayatını ve Enerjini Organize Etmenin Yolları	0.00	7.14	14.29	35.71
Ruhsal Yorgunluk Nasıl Geçer?	0.00	0.00	0.00	9
Kriz Anında Sakin Kalma Yolları Beyhan Budak [Klinik Psikolog]	0.00	12.50	6.25	37.50
Verimli Çalışma İçin Dört Mükemmel Yöntem	0.00	22.22	33.33	33.33
Antrenman Programım Haftalık spor rutinim	0.00	23.08	38.46	61.54
Alanında iş bulamayanlar: Elektrik ve elektronik mühendisi	0.00	31.25	25.00	43.75
Alanında iş bulamayanlar: İnşaat mühendisi	0.00	09.09	18.18	36.36

Table 7.2 Continued.

Videos / Rate of Algorithms	RAKE	YAKE	TF-IDF /	ArKeyword Extractor
İngiltere'de Eğitim Almanın Önemi JANE (Evgiina Shaeva) ile söyleşi	13.04	13.04	34.78	56.52
Kontrast - Taksici 24. Bölüm Türkiye vs Amerika	6.67	46.67	40.00	60.00
Gol Sevinçlerinde Yaşanan İlginç Olaylar: "Parmağı Koptu!"	0.00	20.00	20.00	60.00
Kontrast - Afgan Mülteci 20. Bölüm Amerika vs Türkiye	09.09	45.45	45.45	64.64
Kontrast - Berber 12. Bölüm AMERİKA VS TÜRKİYE	0.00	25.25	25.25	37.50
Kontrast - 1. Bölüm Evsizler AMERİKA VS. TÜRKİYE (Altyazı dilini seçmeyi unutmayınız!)	0.00	23.53	23.53	41.18
İnsanın Fabrika Ayarları: Tükenmişlik Sendromu Sinan Canan DenizBank Deniz Akademi	4.17	16.67	20.83	25.00
İnsanın Fabrika Ayarları: Beslenme Sinan Canan DenizBank Deniz Akademi	4.76	14.29	14.29	42.86
Google'da çalışmanın dezavantajları	10.00	40.00	60.00	80.00
SoruYorum - Yetersizlik Kaygısı / Sosyal Fobi	0.00	20.00	26.67	33.33
SoruYorum - Hayat Amacı / Öz	10.00	20.00	20.00	50.00
INSTAGRAMDA SATILAN ÇAKMA AIRPODS	10.00	10.00	20.00	80.00

Table 7.2 Continued.

Videos / Rate of Algorithms	RAKE	YAKE	TF-IDF /	ArKeyword Extractor
En Hızlı Köpüren Sabun Deneyi - Barış Özcan'ın Ses Desteğiyle	5.56	50.00	50.00	72.22
Görünmezlik Deneyi Yaptım	15.38	23.08	23.08	38.46
Çok Şaşıracaksınız Yanlışlıkla Keşfedilmiş 5 Buluş	0.00	26.32	47.37	52.63
Tuzdaki Bomba	26.67	26.67	33.33	73.33

7.3. STT Keyword Scores of Each Benchmark Video

Table 7.3 STT Algorithm Based Scores of Each Video.

VIDEO_NAME	VOSK (WER)	VOSK (CER)	ARSPEECH (WER)	ARSPEECH (CER)	W2V2 (WER)	W2V2 (CER)
Zaman Nedir?	47.35	18.02	84.67	64.52	62.99	26.77
DOLARIN İCADI DOLAR NEDEN DÜNYANIN EN GÜÇLÜ PARASI?	44.78	18.91	86.54	63.39	63.92	27.96
GEÇMİŞTEKİ PANDEMİLER NASIL BAŞLAMIS VE BİTMİŞTİ?	59.56	33.33	86.52	65.28	77.8	42.61
AY'IN İKİYE BÖLÜNMESİ GERÇEKLERİ! BİLİMSEL	45.34	19.37	84.25	63.86	66.24	25.15
Suudi Arabistan Hakkında Bilmediğiniz 27 Gerçek	47.11	19.38	85.38	63.16	57.74	19.97
Eski Hacker'a Sorduk! - "Bilgisayarı Kapatsanız Bile Ben Hala Arka Planda Aktifim "	55.92	26.18	91.46	70.96	82.23	33.19
Geleceğin Otomobili Nasıl Olmalı	37.82	12.07	80.38	58.59	52.23	17.89
İnsansı ve üreyen robotlar	36.59	10.68	76.11	54.97	53.17	19.21
Bulut Bilişimi Anlamak (Cloud Computing)	40.01	13.54	84.05	67.54	54.6	21.29
Yoksulluk ve Yardım Etme İkilemi Üzerine	30.48	8.56	79.56	59.64	53.92	20.59
Lokum Gibi Arnavut Ciğeri	56.35	25.45	86.32	61.81	69.06	32.95

Table 7.3 Continued.

VIDEO_NAME	VOSK (WER)	VOSK (CER)	ARSPEECH (WER)	ARSPEECH (CER)	W2V2 (WER)	W2V2 (CER)
10° KARAVANDA KIŞ HAYATI - Webasto İşe Yarıyor Mu ?	53.77	28.06	86.49	64.3	83.74	41.35
Türkiye'nin nükleer macerası: Akkuyu Nükleer Santrali hakkında her şey	49.16	21.89	86.48	62.89	69.29	27.56
Yapay zeka nedir? Yapay zekanın yaptığı inanılmaz şeyler! İNSANLIK 2.0	61.46	34.0	87.43	64.45	86.88	49.96
Ahmet Nur Çebi - Sergen Yalçın Görüşmesi	46.19	19.47	84.71	61.95	74.34	30.43
Zorba Nasıl Olunur?	100.0	100.0	100.0	100.0	100.0	100.0
Kodlayarak Nasıl Sanat Yapabiliriz?	33.43	11.57	80.02	59.16	57.29	21.36
Hiçbir Taksici Benim Gibi Davranmamış	86.11	54.85	92.55	72.59	96.54	62.31
Polonyalıların ve Almanların Türkler hakkındaki görüşleri	100.0	100.0	100.0	100.0	98.85	75.79
Geçinemeyenler: "Kölelikten daha kötü bir durumdayız"	58.81	32.16	96.03	73.9	97.27	75.11
Richard Stallman - Özgür Yazılım	100.0	100.0	100.0	100.0	98.44	75.32
DİPLOMASIZ YAZILIMCI	48.94	19.94	96.05	78.65	96.61	74.55
Kış Şartlarında Araç Kullanma Tekniklerinin 10 KURALI - Kışın Araba Sürmenin 10 Kuralı	48.85	27.43	87.39	63.34	73.67	33.99

Table 7.3 Continued.

VIDEO_NAME	VOSK (WER)	VOSK (CER)	ARSPEECH (WER)	ARSPEECH (CER)	W2V2 (WER)	W2V2 (CER)
Takside Bile 4-3-3 Aklıma Geliyor	82.6	56.33	92.74	70.28	95.48	59.65
Kafaya Takmama Sanatı:	32.16	14.06	79.16	60.08	55.99	19.36
Şeker Dost Mu Düşman Mı?	41.76	12.35	82.09	56.2	57.16	21.14
Dayıoğlu: "Çocuk kalmasaydım olmazdı"	48.97	21.65	86.93	60.32	75.86	27.37
Devlet Neden Var, Ne İşe Yarar?	36.35	16.18	82.26	57.99	50.68	14.98
Cache Friendly Programming Teknikleri	51.05	21.0	100.44	81.71	66.91	25.01
An'da Mutluluk	39.0	18.95	81.68	60.77	61.98	20.47
Uyku Düzenimizi Değiştirmek Bizim Elimizde!	40.98	14.27	82.3	62.14	68.52	29.12
Elon Musk Neden 35 Metrekarelik Bir Evde Yaşıyor?	45.18	19.86	84.54	59.92	67.97	33.43
Mutlu İnsanların Yaptığı 3 Şey	31.4	14.94	77.65	57.89	55.02	28.6
Bu 5 Şeyi Yapan İnsanlar Kendi Kendine Yetiyor	34.28	14.94	79.15	59.38	54.95	21.84
Tuna Tavus İçin 'Sıradan Bir Gün'	76.65	63.29	89.58	66.37	87.85	46.72
Trata: Gezici Restoran	65.6	44.58	90.21	67.75	84.74	43.22
Biraz Ada Biraz Caz: Bozcaada Caz Festivali	63.14	42.41	87.66	65.14	83.85	41.81
Lastik Pabucun Anatomisi	68.17	49.06	88.81	67.1	83.99	45.79
Moda (Semt Olan)	51.56	28.3	87.17	64.24	80.87	35.6
Mutluluğunu Kendin Engelliyor Olabilirsin: 9 Sebep	34.75	15.55	82.79	59.85	52.81	18.65

Table 7.3 Continued.

VIDEO_NAME	VOSK (WER)	VOSK (CER)	ARSPEECH (WER)	ARSPEECH (CER)	W2V2 (WER)	W2V2 (CER)
Pereira İle Yollar Ayrıldı: Ali Koç'un Önünde 2 Yol Var	44.58	19.45	84.13	59.86	75.53	29.85
Kontrast - Araba Tamircisi 21. Bölüm Amerika vs Türkiye	54.23	29.89	85.26	60.79	77.57	33.42
Kontrast - Eski Mahkûm	72.28	51.37	92.84	68.98	86.68	50.1
Kontrast - Tır Şoförü 19. Bölüm Türkiye vs Amerika	55.32	26.75	88.16	63.22	82.19	33.82
Hayatını ve Enerjini Organize Etmenin Yolları	35.47	11.28	80.95	59.76	59.19	19.82
Ruhsal Yorgunluk Nasıl Geçer?	47.89	18.72	88.8	66.4	64.11	28.2
Kriz Anında Sakin Kalma Yolları	35.14	14.52	79.53	57.25	50.49	19.27
Verimli Çalışma	33.55	7.4	85.07	64.26	57.95	33.78
Antrenman Programım Haftalık spor rutinim	61.87	39.24	90.48	72.3	78.2	33.81
Alanında iş bulamayanlar: Elektrik ve elektronik mühendisi	33.88	10.98	77.99	58.39	58.03	24.1
Alanında iş bulamayanlar: İnşaat mühendisi	40.45	15.78	80.75	54.69	65.69	28.17
İngiltere'de Eğitim Almanın Önemi	96.67	78.35	96.93	73.25	102.28	82.09
Kontrast - Taksici	63.96	44.07	89.13	64.72	81.18	48.46
Gol Sevinçlerinde Yaşanan İlginç Olaylar:	68.63	46.64	88.52	67.03	95.31	59.14
Kontrast - Afgan Mülteci	79.43	59.1	92.54	67.11	93.11	60.78

Table 7.3 Continued.

VIDEO_NAME	VOSK (WER)	VOSK (CER)	ARSPEECH (WER)	ARSPEECH (CER)	W2V2 (WER)	W2V2 (CER)
Kontrast - Berber 12. Bölüm AMERİKA VS TÜRKİYE	59.31	36.64	88.8	63.64	84.01	35.12
Kontrast - 1. Bölüm Evsizler AMERİKA VS. TÜRKİYE (Altyazı dilini seçmeyi unutmayınız!)	73.61	52.9	92.53	66.12	89.05	49.67
İnsanın Fabrika Ayarları: Tükenmişlik Sendromu Sinan Canan DenizBank Deniz Akademi	29.16	12.17	79.7	61.32	51.87	15.94
İnsanın Fabrika Ayarları: Beslenme Sinan Canan DenizBank Deniz Akademi	30.3	9.79	82.53	61.39	52.42	17.3
Google'da çalışmanın dezavantajları	35.92	13.02	82.61	63.28	57.56	22.77
SoruYorum - Yetersizlik Kaygısı / Sosyal Fobi	44.91	26.69	89.57	69.37	80.79	44.87
SoruYorum - Hayat Amacı / Öz	41.43	21.4	85.95	66.97	71.79	42.12
Instagramda Satılan Çakma Aırpods	64.14	39.92	87.89	65.6	84.9	40.23
En Hızlı Köpüren Sabun Deneyi - Barış Özcan'ın Ses Destegiyle	40.89	15.42	85.62	61.36	66.93	28.73
Görünmezlik Deneyi Yaptım	62.27	35.71	88.65	61.11	78.36	39.75
Çok Şaşıracacağınız Yanlışlıkla Keşfedilmiş 5 Buluş	53.69	21.37	86.19	63.59	74.37	29.82
Tuzdaki Bomba	49.92	26.0	82.02	57.85	80.17	42.45

REFERENCES

- Adrien B., Florian B., Beatrice D., (2013), "TopicRank: Graph-Based Topic Ranking for Keyphrase Extraction", International Joint Conference on Natural Language Processing(IJCNLP), 543-551, Nagoya, Japan, October.
- Allahyari M., Pouriyeh S., Assefi M., Safaei S., Trippe D., Gutierrez B., and Kochut K., (2017), "Text summarization techniques: a brief survey", arXiv e-prints, 1707.02268(1), 1-6.
- Ardila R., Branson M., Davis K., Henretty M., Kohler M., Meyer J., Morais R., Saunders L., Tyers F. M., and Weber G., (2020) "Common Voice: A Massively-Multilingual Speech Corpus", 12th Conference on Language Resources and Evaluation (LREC), 4211-4215, Marseille, France, May.
- Baevski A., Zhou H., Mohamed A., and Auli M., (2020), "wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations", arXiv e-prints, 2006.11477(1), 1-19.
- Bahdanau D., Cho K., and Bengio Y., (2014), "Neural machine translation by jointly learning to align and translate", arXiv e-prints, 1409.0473(1), 1-15.
- Bojanowski P., Grave E., Joulin A., Mikolov T., (2017), "Enriching word vectors with subword information", Transactions of the Association for Computational Linguistics (TACL), 135-146, Prague, Czech Republic, June.
- Brynjolfsson E., Horton J., Ozimek A., Rock D., Sharma G., and TuYe H., (2020), "COVID-19 and Remote Work: An Early Look at US Data", NBER Working Papers, 10.3386/w27344(1), 1-26 June.
- Campos R., Mangaravite V., Pasquali A., Jatowt A., Jorge A., Nunes C., and Jatowt A., (2020), "YAKE! Keyword Extraction from Single Documents using Multiple Local Features", In Information Sciences Journal Elsevier, Vol 509, 257-289.
- Campos R., Mangaravite V., Pasquali A., Jorge A., Nunes C., Jatowt A., (2018), "YAKE! Collection-independent Automatic Keyword Extractor", 40th European Conference on Information Retrieval (ECIR'18), 806-810, Grenoble, France. March.
- Campos R., Mangaravite V., Pasquali A., Jorge A., Nunes C., Jatowt A., (2018), "A Text Feature Based Automatic Keyword Extraction Method for Single Documents", 40th European Conference on Information Retrieval (ECIR'18), 684-691, Grenoble, France, March.
- Carl M., Pease C., Iomdin L., and Streiter O., (1999), "Towards a Dynamic Linkage of Example-Based and Rule-Based Machine Translation", Machine Translation, 15(3), 223-257.

Chen D., Fisch A., Weston J., Bordes A., (2017), "Reading Wikipedia to Answer Open-Domain Questions", 55th Annual meeting of the Association for Computational Linguistics (ACL 2017), 1870-1879, Vancouver, Canada, April.

Cho K., Van Merriënboer B., Bahdanau D., and Bengio Y., (2014), "On the properties of neural machine translation: Encoder-decoder approaches", arXiv e-prints, 1409.1259(1), 1-9.

Devlin J., Chang M. W., Lee K., and Toutanova K., (2018), "BERT: Pre-training of deep bidirectional transformers for language understanding", arXiv e-prints, 1810.04805(1), 1-16.

Eiselt A., Figueroa A., (2013), "A Two-Step Named Entity Recognizer for Open-Domain Search Queries", Proceedings of the Sixth International Joint Conference on Natural Language Processing (IJCNLP), 829-833, Nagoya, Japan, October.

Goldberg Y., (2016), "A Primer on Neural Network Models for Natural Language Processing", Journal of Artificial Intelligence Research(JAIR), 57(1), 345–420.

Goodfellow I., Bengio Y., Courville A., (2016). Deep Learning, 9780262035613, MIT Press.

Graves A., Fernández S., and Gomez F., (2006), "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks", 23rd International Conference on Machine Learning(ICML), 369-376, Pittsburgh, Pennsylvania, USA, June.

Hendrycks D., and Gimpel K., (2016), "Gaussian error linear units (GELUs)", arXiv e-prints, 1606.08415(1), 1-9.

Hirschman L., Gaizauskas R., (2001), "Natural Language Question Answering: The View from Here", Natural Language Engineering, 7(4), 275-300.

Koehn P., (2009), "Statistical Machine Translation", 9780511815829, Cambridge University Press.

Malzer C., Baum M., (2020), "A hybrid approach to hierarchical density-based cluster selection", IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), 223-228, Karlsruhe, Germany, September.

McInnes L., Healy J., and Melville J., (2018), "UMAP: Uniform manifold approximation and projection for dimension reduction", arXiv e-prints, 1802.03426(1), 1-63.

Mikolov T., Chen K., Corrado G., Dean J., (2013), "Efficient estimation of word representations in vector space." arXiv e-prints, 1301.3781(1), 1-12.

Mikolov T., Sutskever I., Chen K., Corrado G. S., Dean J., (2013), "Distributed representations of words and phrases and their compositionality", Proceedings of the 26th International Conference on Neural Information Systems(NIPS 13), 3111-3119, Lake Tahoe, Nevada, United States, December.

Miller D., (2019), "Leveraging BERT for extractive text summarization on lectures.", arXiv e-prints, 1906.04165(1), 1-7.

Nadeau D., Sekine S., (2007), "A survey of named entity recognition and classification", *Linguistic Investigations*, 30(1), 3-26.

Nallapati R., Zhou B., Gulcehre C., and Xiang B., (2016),"Abstractive text summarization using sequence-to-sequence rnns and beyond." arXiv e-prints, 1602.06023(1), 1-12.

Nilsson M., and Ejnarsson M., (2002), "Speech Recognition using Hidden Markov Model (Dissertation)", Degree of Master of Science in Electrical Engineering, Blekinge Institute of Technology.

Pennington J., Socher R., and Manning C. D., (2014),"GloVe: Global Vectors for Word Representation", Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), 1532–1543, Doha, Qatar, October 2014.

Petrov S., Das D., McDonald R., (2011), "A Universal Part-of-Speech Tagset", arXiv e-prints, 1104.2086(1), 1-6.

Quarteroni S., and Suresh M., (2009), "Designing an interactive open-domain question answering system.", *Natural Language Engineering*, 15(1), 73-95.

Rada M., and Paul T., (2004), "TextRank: Bringing Order into Text", In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004), 404–411, Barcelona, Spain. July.

Rao J., Rao S., (2016), "A Survey on Sentiment Analysis and Opinion Mining", Proceedings of the International Conference on Advances in Information Communication Technology & Computing (AICTC '16), 1-5, New York, USA, August.

Rose S., Engel D., Cramer N., Cowley W., (2010), "Text Mining Applications and Theory", 9780470689646, Wiley Press.

Singh S. P., Kumar A., Darbari H., Singh L., Rastogi A., and Jain S., (2017), "Machine translation using deep learning: An overview," International Conference on Computer, Communications and Electronics (Comptelix), 162-167, Jaipur, India, July.

Torregrosa D., Pasricha N., Masoud M., Chakravarthi B. R., Alonso J., Casas N., Arcan M., (2019), "Leveraging Rule-Based Machine Translation Knowledge for Under-Resourced Neural Machine Translation Models", Proceedings of Machine Translation Summit XVII: Translator, Project and User Tracks, 125-133, Dublin, Ireland, August.

van der Maaten L., and Hinton G., (2008), "Visualizing Data using t-SNE", Journal of Machine Learning Research, 9(2605), 2579-2605.

Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser L., and Polosukhin I., (2017), "Attention is all you need", 31st Conference on Neural Information Processing Systems (NIPS), 6000-6010, Long Beach, CA, USA.

Web 1, (2022), <https://sensortower.com/blog/video-conferencing-apps-mau-growth>, (Retrieved 29/01/2022).

Web 2, (2022), <https://pypi.org/>, (Date of Access 29/01/2022).

Web 3, (2022), <https://www.ffmpeg.org/>, (Date of Access 29/01/2022).

Web 4, (2022), <https://github.com/>, (Date of Access 29/01/2022).

Web 5, (2022), <https://www.youtube.com/>, (Date of Access 29/01/2022).

Web 6, (2022), <https://arvoice.iot-ignite.com/>, (Date of Access 29/01/2022).

Web 7, (2022), <https://huggingface.co/>, (Date of Access 29/01/2022).

Web 8, (2022), https://tr.wikipedia.org/wiki/James_Webb_Uzay_Teleskopu, (Retrieved 29/01/2022).

Winograd T., (1972), "Understanding natural language", Cognitive Psychology, 3(1), 1-3.

BIOGRAPHY

Hakkı Yavuz ERZURUMLU graduated from the Department of Computer Engineering, Faculty of Natural Sciences, Karadeniz Technical University. He is currently working as an AI team leader in ARDİC R&D. He is eager to learn new technologies and techniques. During his master's he worked on deep learning and natural language processing. He likes robotics and computer games. His research interests are artificial intelligence, robotics, natural language processing.



APPENDICES

Appendix A: Publications on the thesis

Erzurumlu H.Y., Akgül Y.S., (2022), “Adaptive Keyword Extraction Service for Turkish”, In 2022 11th International Conference on Soft Computing, Artificial Intelligence and Applications (SAI-2022)

