AMERICAN UNIVERSITY OF BEIRUT

DIRECT SPEECH TO SPEECH TURKISH TO ARABIC

by

MASSA BAALI

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Science
to the Department of Computer Science
of the Faculty of Arts and Science
at the American University of Beirut

Beirut, Lebanon
April 12, 2021

AMERICAN UNIVERSITY OF BEIRUT

DIRECT SPEECH TO SPEECH TURKISH TO ARABIC

by

MASSA BAALI

Approved by:

Dr. Wassim El Hajj, Associate Professor — Advisor
Computer Science

Dr. Mohammad Nassar, Assistant Professor — Member of Committee
Computer Science

Dr. Ahmad Ali, Principal Engineer — Member of Committee
Qatar Computing Research Institute

Date of thesis defense: April 12, 2021

AMERICAN UNIVERSITY OF BEIRUT

THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name: Baali Massa Amer

Last First Middle

☑ Master's Thesis ○ Master's Project ○ Doctoral Dissertation

☑ I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

□ I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes after: **One ___ year from the date of submission of my thesis, dissertation or project.**
Two ___ years from the date of submission of my thesis , dissertation or project.
Three ___ years from the date of submission of my thesis , dissertation or project.

23-4-2021

Signature Date

This form is signed when submitting the thesis, dissertation, or project to the University Libraries

Acknowledgements

Thanks for helping me move to the next level and for the continuous guidance, patience, and support during the past two years...

Dr. Wassim El Hajj

Thanks for helping me explore all the possibilities in any challenge...

Dr. Ahmed Ali

I would like to express my gratitude to the members of my examination committee especially Dr. Mohammad Nassar.

This work is dedicated to the memory of my role model, Mouafaq Daaboul, who always believed in my ability to be successful. You are gone but your belief in me has made this journey possible.

Dr. Mouafaq Daaboul

You always wanted me to be the best! Thanks for convincing me continue my studies, for believing in my abilities and for the limitless support...

Dr. Fayez Kiwan

I would like to thank my family for the endless support...

Family

I would like to thank my friends for making this journey more enjoyable...

Friends

The biggest thank is for you.

Mom

An Abstract of the Thesis of

Massa Baali for Master of Science

Major: Computer Science

Title: Direct Speech to Speech Turkish to Arabic

Dubbed series are gaining a lot of popularity in recent years with strong support from major media services providers. Such popularity is fueled by studies that showed that dubbed versions of TV shows are more popular than their subtitled equivalents. In this paper, we propose an unsupervised approach to construct speech-to-speech corpus, aligned on short segment level, to produce a parallel speech corpus in the source- and target- languages. Our methodology exploits speech recognition, machine translation and noisy frames removal algorithms, to match segments in both languages. Without losing any generalization, our approach was successfully applied on Turkish-Arabic dubbed series. Out of 36 hours, our pipeline was able to generate 17 hours of paired segments with 70% overall accuracy. The corpus will be freely available for the research community.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Clean and large parallel speech corpora constitute a major building block in developing speech-to-speech translation systems. However, building such big corpora is a very costly and lengthy process that normally relies on manual labor. According to our knowledge, no attempts have been made to create a high quality and large parallel speech corpora in an automatic fashion. In this work, we propose an unsupervised approach that takes as input a dubbed series and produces a Speech-to-Speech Corpus in the respective languages of the dubbed series. In the proposed approach, we address the following major challenges: (1) removing the noisy voice segments, such as Ads, from each dubbed version to produce dubbed versions that have almost the same duration, (2) reducing the effect of the background noise or music when matching the speech segments, and (3) devising a set of rules for segments' matching where hyperparameters can be tuned to optimize the tradeoff between quality and corpus size. Without loss of generality, we demonstrate our pipeline using Turkish series dubbed into Arabic. We picked these languages since dubbed series have shown a lot of attention in the past few years in the Arab region, where more than 85 million Arab viewers watch Syrian-dubbed Turkish series [1]. The original input was 51 hours of Turkish series and the corresponding 54 hours of the dubbed Arabic version (in the Arabic version there were a lot of Ads). After cleaning the versions from the Ads, noise segments, and unrecognized segments, 36 hours of each dubbed version were produced. After applying the segments' matching rules and tuning the hyperparameters to gain good quality parallel corpus, 17 hours of parallel speech corpus was produced. The experiments that relied on random samples annotated by bilingual speakers showed a 70% overall accuracy. The same random samples were also annotated for emotions producing an 80% accuracy.

The project pipeline includes data collection,voice activity detection (VAD), automatic speech recognition (ASR), machine translation (MT), and finally, we consider further paralinguistic features; such as emotion recognition.

Chapter 2

Literature Review

In this section we will present the studies that are related to the corpus creation and the ones that are related to creating Arabic Synthesizer which is one of the main components in the application presented. Creating a parallel corpus exhausts a lot of resources. It is a very lengthy process and time consuming. In [2], the authors extracted parallel speech corpora based on any language pair from dubbed movies in which some corresponding prosodic parameters are extracted. Unlike [3], who explored a method based on machine learning for automatically extracting bilingual audio subtitle pairs from movies. They used raw movie data, and they defined the long term spectral distance, subtitles time distance and subtitle time-stamps to segment the bilingual speech regions. In [4], they used the relative difference of syllable count estimates between source and target material as the similarity constraint. They assumed that more detailed constraints, based on accentuation, stress marks, expected speech durations, articulatory and prosodic features, will be needed to match human dubbing performance, where they used the Heroes corpus[5].

Fedrico et al. [6] presented Prosodic Alignment model which does not require cross-lingual information, but guides the search for the optimal alignment with two types of information: the speaking rate match between corresponding source-target phrases and the linguistic plausibility of the chosen split points. They used TED talks dataset that has both English and Italian. Farrus et al. [7] aimed to not only translate the spoken segments and synthesize them, but also to make the translated and synthesized segments match the original durations and phrasings. Their method explores the attention mechanism output in neural machine translation to find plausible phrasing for the translated dialogue lines and then uses them to condition their synthesis.

[8] created a system involving scansnap SV600 scanner and Google optical character recognition (OCR) for building a parallel corpus, which is a very significant part of the statistical machine translation (SMT). They trained a language model for the SMT system which depends on the amount of parallel corpus. They proposed a precise way of producing parallel corpus between English and Indian languages. They were able to generate 40 parallel sentences in one hour time with this approach. [9] demonstrated a bilingual collaborative annotation method that annotates English discourse units based on Chinese ones, and annotates Chinese discourse structure based on English ones consequently. This process guarantees complete discourse structure alignment between parallel texts and reduces the cost for annotating texts of two languages as well.

[10] presented an unsupervised approach that automatically creates a monolingual parallel corpus for text simplification using sentence similarity based on word embeddings. For any pair of sentences consisting of a complex sentence and its simple counterpart, they used a many-to-one approach of aligning each word in the complex sentence with the most similar word in the simple sentence and calculate sentence similarity by averaging these word similarities. Their experi-

mental results show the good performance of the proposed method in a single-language parallel corpus construction task for the simplicity of the English text. A survey by Khosla et al. [11] was proposed to explore the existing methods of building a parallel corpus. The survey covers some of the techniques of the main parallel corpus built. It conducts only the corpora that are built aligned at the sentence and document level. The first parallel automated sentence alignment was implemented by [12], which is based on the assumption that long sentences will be translated into long sentences and short sentences into short ones. Their strategy works remarkably well on language pairs with a strong correlation with length, such as French and English. A number of various approaches to sentence alignment have been introduced such as sentence length, co-occurrence of word, dictionary usage and parts of speech tagging to deliver a parallel bilingual corpus [13].

In [14], they used a new alignment method based on time overlap to build a parallel corpus for various pairs of languages. The approach was used in exploring a corpus of 23,000 pairs of aligned subtitles covering about 2,700 films in 29 languages. Many adopted the manual approach like [15], who created a parallel bilingual parallel syntactically annotated corpus of Czech-English, which is part of a project at the university of Charles. The target text is manually achieved by translating an established monolingual annotated corpus.

In [16], they have used Amazon's Mechanical Turk to construct parallel corpus for six Indian languages: Bengali, Hindi, Malayalam, Tamil, Telugu and Urdu. The source data is the 100 most viewed documents on Wikipedia for each language and the translations are achieved through human translators. The first step was to create the bilingual dictionaries. These dictionaries were then used to construct glosses of the source sentence and then compared to the manual translations implemented. One of the methods is machine translation [17], where a parallel corpus for multiple languages (English-German, English-Spanish, English-Czech) was built by taking source text and acquiring target text through five MT systems (Joshua, Lucy, Metis, Apertium, MaTrEx). The parallel corpus is annotated with meta information. The size of the corpus built is 2051 sentences translated by five different MT systems in six translation directions and annotated with different metadata information received from the translation model. Table 2.1 benchmarks our study compared to previous studies.

Paper	**Unsupervised**	**Language**	**Result**	**Speech**	**Bilingual**	**Size**	**Dub**
Ours	Y	Any	79% overall accuracy	Y	Y	17 hours	Y
[2]	Y	Any	-	Y	Y	80-49 mins	Y
[3]	N	en-fr	LTSD 41.39% Subtitle 37.88%	Y	Y	1050 segments	Y
[17]	N	en-cs,de,es	BLEU 18.95	N	Y	2051 sentences	N
[16]	N	English and 6 languages from Indian sub-contents	5 votes cast on 65% of sentences	N	Y	100 top doc on wiki	N
[15]	N	cs-en	-	N	Y	1M words	N
[14]	N	29 languages	85% correct alignment	N	Y	23k pairs sentences	N
[9]	N	zh-en	90% annotation agreement	N	Y	140K words	N
[10]	Y	en	BLEU 26.3	N	N	126k pairs of article	N

Table 2.1: Benchmarking our studies with previous work.

Chapter 3

Parallel Corpus Creation

3.1 Parallel Speech Corpus Construction

The process of creating the parallel speech corpus from dubbed series goes through multiple steps that include: data collection, voice activity detection, segmentation, automatic speech recognition, text translation, and segments matching. Figure 3.4 illustrates an overview of this pipeline. In what follows, each step of the pipeline is explained using a running example on Turkish-Arabic dubbed series. The same pipeline could be used for any dubbed series without losing generality.

3.1.1 Data Collection and Video Matching

In this phase, we aim to download and clean a dubbed video in any two languages. In some cases, you might be able to get hold of dubbed videos with equal duration and without any additional noise segments such as commercials; in which case, the videos are ready to be processed. In other cases, one or both of the dubbed versions might include noise segments that should be removed, in order to end up

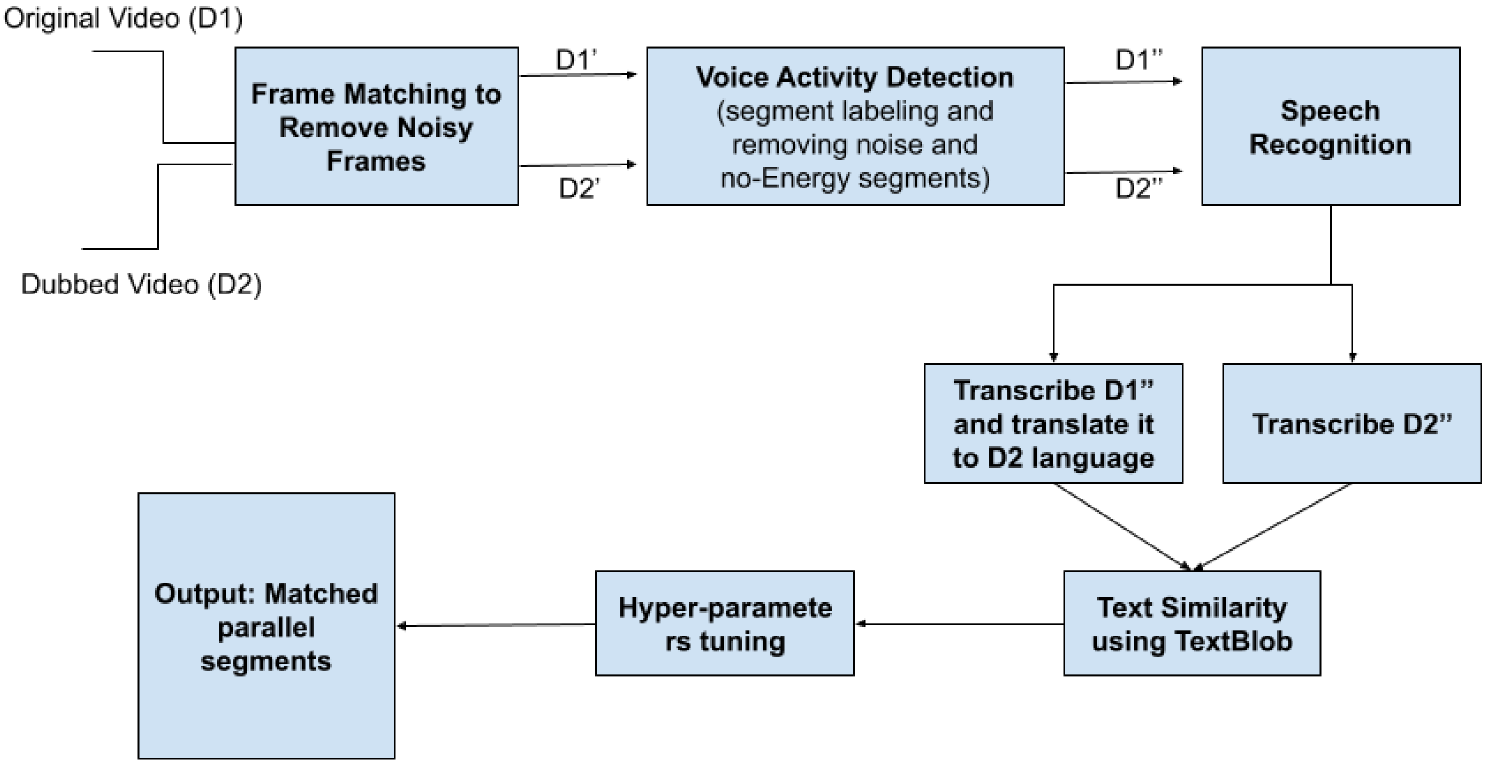
Original Video (D1)
Dubbed Video (D2)
Frame Matching to Remove Noisy Frames
D1'
D2'
Voice Activity Detection (segment labeling and removing noise and no-Energy segments)
D1"
D2"
Speech Recognition
Transcribe D1" and translate it to D2 language
Transcribe D2"
Text Similarity using TextBlob
Hyper-parametete rs tuning
Output: Matched parallel segments

Figure 3.1: Pipeline for the Parallel Speech Corpus Construction: data collection, voice activity detection (VAD), automatic speech recognition (ASR), machine translation (MT).

with a clean version of the dubbed videos. Algorithm 1 takes as input the dubbed versions and cleans them up. The first step in the algorithm is to convert every dubbed video into frames, 30 frames per second. At time t, we pick the frame in D1 and compare it with 500 consecutive frames in D2 (also starting at t) i.e. we are searching whether the frame in D1 at time t exists in a 16 seconds video segment in D2 starting at time t. Every two frames are compared using skimage [18] to compute the mean structural similarity index between two images. The values returned by skimage are in the range of [0,1], where a value close to 1 indicates high similarity between the two images. In case two frames returned a similarity greater than 0.75, we assume that these two images are similar, and thus the frame we are investigating is a valid frame, and not a noise frame. This process is repeated for every frame in D1. A frame in D1 with low similarity in the corresponding consecutive frames in D2 is removed. Once done, we run the same algorithm starting with the other dubbed version. For instance, all frames in D1 that correspond to images from an Ad segment will be removed, since no matching frames exist for them in D2.

Algorithm 1 Proposed procedure to remove noise frames from the dubbed videos

```
D1: Dubbed version 1
        D2: Dubbed version 2
        Result: Clean and matched dubbed videos
Convert D1 and D2 into frames, 30 frames per second; while there are more
 frames f in D1 do
    Extract frame f at time t from D1
    while $i$ <500 do
        Extract frame f' at time t from D2
        t' = t
        r = skimage(f, f')
        if $r \geq 0.75$ then
            break;
        else
            t'=t'+1
            i=i+1
        end
    end
    if i==499 then
        remove frame f at time t from D1
    end
    t = t + 1 (move to analyze the next frame in D1)
end
```

To apply the algorithm on real data, we downloaded three Turkish (TR) - Arabic (AR) dubbed series from YouTube. The three Turkish series had durations of 12, 9, and 30 hours, respectively. The corresponding Arabic series had durations of 13, 10, and 31 hours, respectively. When checking the series manually, we noticed that the Arabic version contained a lot of commercials, the fact that explains the extra duration of the Arabic series when compared to the Turkish series. After running Algorithm 1, we ended up with a matching duration of both series equivalent to 12, 9, and 30 hours respectively. Moving forward, we operate on all series as one big video; i.e., the new input is a TR-AR dubbed video pair composed of 51 hours each.

3.1.2 Voice Analysis Detection

Algorithm 1 produced matching dubbed videos without noise frames, such as commercials. We now extract the audio from the dubbed video samples at 16Khz. Then, we process the audio files of each dubbed videos. We investigate the impact of segmentation using voice activity detection (VAD). We use the pre-trained model suggested in [19] that extract meta-data from each audio file; gender, noise, music and noEnergy.

We applied this phase to the TR-AR dubbed series. Figure 3.2 shows the distribution of the five labels in the Turkish Dubbed version and the Arabic Dubbed version. The frequency of most labels is similar with the exception of noEnergy. We attribute this to the Arabic dubbing, where the actors ignore the amount of silence in the original video while abiding by the scene time.

3.1.3 Speech Segments Matching

As of now, phase 2 resulted in speech segments that are labeled as female, male, and music. Every segment has a start time, end time, and duration. In this phase, we start by transcribing every dubbed version. To perform this step, Google Speech Recognition API can be used regardless of the dubbed version language. For every dubbed version, every segment now corresponds to its transcription.

We then translate the segments' transcription of one dubbed version to the

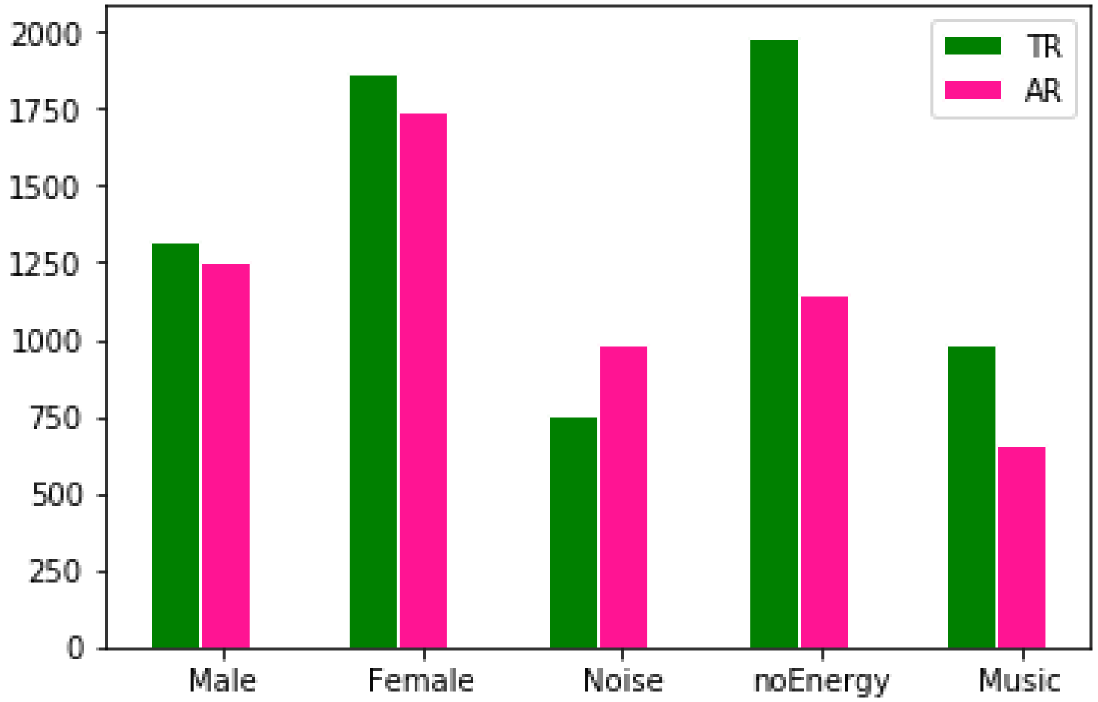

Figure 3.2: Frequency of TR-AR labeled segments, *y-axis* is the number of segments.

language of the other dubbed version. For example, we translate the Turkish transcribed segments into Modern Standard Arabic (MSA) using Google Translate API, which would allow us to compare the translated Turkish segments with the transcribed Arabic Levantine segments. Any segment that was not recognized by the transcription API is removed. At this point, and when applying this step on the TR-AR dubbed versions, we ended up with 36 hours videos. Next we calculate the similarity between every translated segment in one dubbed version (e.g. the Arabic text translated from the Turkish transcribed segments), and all the transcribed segments in the other version (e.g. the Arabic text transcribed from the Arabic segments). To find the similarity between two text segments, we use TextBlob library which is based on gensim and Fasttext pretrained word2vec model; the procedure used in calculating similarity is calculating the mean feature vector for each sentence, then calculating the cosine distance between those two vectors. Figure 3.3 captures the data structure that is used to record the similarity between every translated segment in the first dubbed version and its corresponding transcribed version.

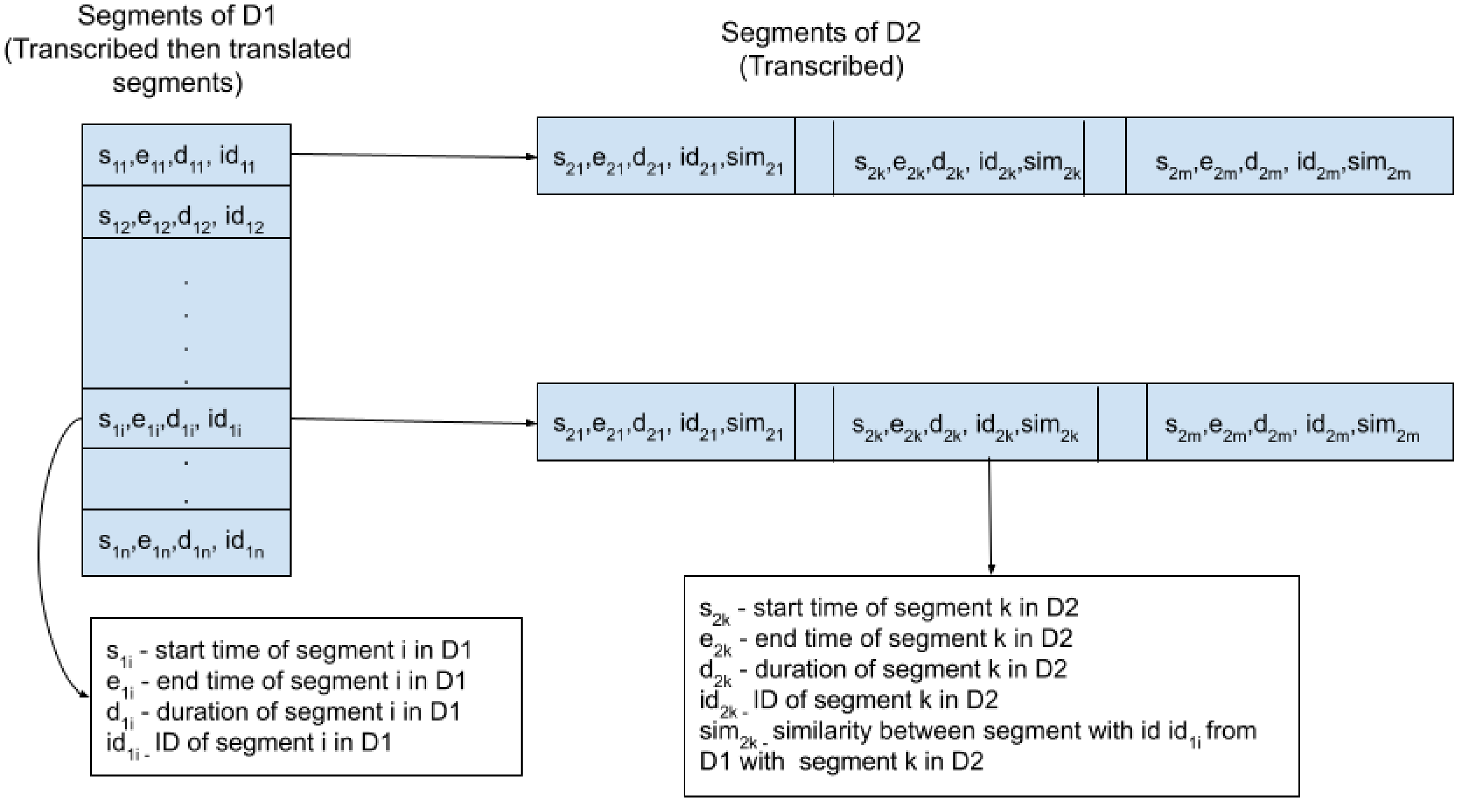
Segments of D1 (Transcribed then translated segments)
Segments of D2 (Transcribed)
$s_{11}, e_{11}, d_{11}, id_{11}$
$s_{12}, e_{12}, d_{12}, id_{12}$
$s_{1i}, e_{1i}, d_{1i}, id_{1i}$
$s_{1n}, e_{1n}, d_{1n}, id_{1n}$
$s_{21}, e_{21}, d_{21}, id_{21}, sim_{21}$
$s_{2k}, e_{2k}, d_{2k}, id_{2k}, sim_{2k}$
$s_{2m}, e_{2m}, d_{2m}, id_{2m}, sim_{2m}$
s_{1i} - start time of segment i in D1
e_{1i} - end time of segment i in D1
d_{1i} - duration of segment i in D1
id_{1i} - ID of segment i in D1
s_{2k} - start time of segment k in D2
e_{2k} - end time of segment k in D2
d_{2k} - duration of segment k in D2
id_{2k} - ID of segment k in D2
sim_{2k} - similarity between segment with id id_{1i} from D1 with segment k in D2

Figure 3.3: Similarity matching between a segment in D1 (first dubbed version) and all segments in D2 (second dubbed version)

The next step is to match one or more segments in the first dubbed version (e.g. TR) to one or more segments in the second dubbed version (e.g. AR), with the following three possibilities:

1. Matching one segment in the first dubbed version to one segment in the second dubbed version.

2. Matching one long segment in the first dubbed version to many segments

in the second dubbed version.

3. Matching many segments in the first dubbed version to one long segment in the second dubbed version.

To do the matching according to one of three mentioned possibilities, we devise the following set of matching rules based on the start time, end time, duration, label, and similarity score of each segment:

- Rule 1: Difference between the segment start time in the first version and the segment start time in the second version, is below a certain threshold.
- Rule 2: Difference between the segment duration in the first version and the segment duration in the second version, is below a certain threshold.
- Rule 3: The segments in both versions should have the same label.
- Rule 4: The similarity score between two segments is above 0.5.
- Rule 5: To combine multiple short segments in one version and match them with one long segment in another version, we adopt a sliding window approach that starts with one short segment abiding by Rule 1, then continues adding more consecutive short segments in the same version until the cumulative duration of the short segments abides by Rule 2, and the similarity score abides by Rule 4.

Since these thresholds are not fixed, we can run multiple experiments while varying the thresholds, with the aim of picking the thresholds that maximize the duration of the matched segments while maintaining a similarity score above 0.5. We applied the algorithms and rules mentioned in this section to the TR-AR dubbed video. Table 3.1 highlights the application of the proposed methodology

Table 3.1: Set of rules for increasing and decreasing the number of segments.

Dur In	Seg Tr,Ar In	Dif Start Time	Dif Dur	Seg Tr-Ar Out	Dur Tr-Ar Out	Avg Sim	Percent
36 hrs	28,800;27,678	<=3	<=2	9,998;9,887	14.3hrs	0.55	39%
36 hrs	28,800;27,678	<=2	<=1	9,158;8,016	13.1hrs	0.53	36%
36 hrs	28,800;27,678	<=4	<=3	10,116;8,848	15hrs	0.55	41%
36 hrs	28,800;27,678	<=5	<=4	8,018;6,915	16hrs	0.54	44%
36 hrs	**28,800;27,678**	**<=9**	**<=8**	**11,581;10,060**	**17.6hrs**	**0.54**	**48%**
36 hrs	28,800;27,678	<=20	<=5	10,330;9,610	15.8hrs	0.53	43%

on dubbed videos. Every row in the table presents the original duration (Dur In) of the video and the corresponding number of extracted segments (Seg Tr-Ar In). Every row also contains values that represent the hyper-parameters, namely, *Dif Start Time* and *Dif Dur* as presented in Rules 1 and 2. The last four entries in every row indicate the resultant number of parallel segments (Seg Tr-Ar Out), their duration (Dur Tr-Ar Out), their average similarity score (Avg Sim), and the percentage duration of parallel corpus produced.

As shown in table 3.1, the difference-start-time threshold of 9 seconds and the difference-duration threshold of 8 seconds, produce a parallel corpus of duration 17.6 hours; i.e. 48% of the cleaned dubbed videos can be transformed into a parallel corpus. When compared to the original dubbed TR-AR videos before cleaning (51 hours), we are able to produce a parallel speech corpus of duration 36% of the original videos. It is also worth noting that the rules above produce an appropriate average similarity score when varying the thresholds, while being able to extract a good percentage of parallel corpus duration. We next evaluate the quality of the produced parallel speech corpus.

Table 3.2: Label distribution for each score.

Score	total	female	male	music
1	512	265	154	93
0.5	163	88	53	22
0	325	173	102	50

3.2 Evaluation

To evaluate the quality of the TR-AR parallel speech corpus, two bilingual speakers were provided with a random sample of 1,000 speech segments. The duration of the segments ranged from 2 seconds to 10 seconds. The annotators were asked to listen to every parallel segment pairs and give the pair a score of 1 if the pair is matching, 0.5 if the pair has minor difference - one version has one word more or different than the other - and 0 if the pair is not matching. Table 3.2 shows the agreement results, where 512 (52%) segment pairs were identical, 163 (17%) segment pairs had minor difference, and 352 (31%) segments pairs were not similar. Since the minor-difference pairs are still almost identical (one word error), the overall resultant similarity is around 70%. A sample pair that is labeled minor-difference is: the Turkish segment transcribed as *Olur olur ya* (English: Okay), and the corresponding Arabic segment transcribed as Buckwalter: ElY Eyny (English: On my eyes), which is understood as "Okay" in the Arabic culture.

In order to check how useful can the parallel speech corpus be in applications, we also asked the same annotators to annotate each segment in every language for emotion (the same 1,000 segment pairs were used). We asked the annotators to label every segment by one of the standard seven classes for emotions: neutral, calm, happy, sad, angry, surprise, and disgust. 791 segments were found to have the same emotion (around 80%). The distribution of the matching emotions are given in table 3.3. For the segment pairs that were found to be identical and with minor-difference, the emotions were almost completely matching. We were interested to know why some of the segment pairs that were not identical would have a similar emotion annotation. For example, the Turkish segment transcribed as *Hoşgeldin Merhaba* (English: Welcome Hello), and the corresponding Arabic segment transcribed as Buckwalter: Ahlyn fyk kyfk yZhr jAyp mn $An mAmA (English: Welcome, you seem to be coming for my mom), both received an emotion annotation of *Happy*. Although the segments are not matching, there are some similarities between them, but one has more information than the other. Hence, the meaning of both segments is preserved when it comes to emotion annotation.

Table 3.3: Emotion distribution of the parallel segments.

Emotion	**Frequency**
Anger	77
Neutral	442
Sad	65
Disgust	15
Calm	26
Happy	39
Surprise	127

To measure the Inter-Annotator Agreement (IAA), we used Cohen's kappa coefficient, a pairwise reliability measure between two annotators, which calculates the accuracy of qualitative items between annotators. The Kappa score was 0.79 which indicates substantial agreement. Based on the high Inter-Annotator Agreement, the high matching between the parallel segments (70%), and the high emotion matching between the parallel segments (80%), we argue that the proposed unsupervised pipeline for generating parallel speech corpus from dubbed series is highly effective.

3.3 Architecture for Speech to Speech Application

In this section, we will present the architecture that we used for the speech2speech implementation.

Our goal is end-to-end spoken language translation. Given an input spectrogram of a sentence spoken in one language, our model outputs a spectrogram of the same sentence spoken in a different language. Seq2Seq is a type of Encoder-Decoder model using RNN. It can be used as a model for machine interaction and machine translation. We aim to map the Turkish speech with the Arabic speech. We convert the speech to spectrogram. A spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time. When applied to an audio signal, spectrograms are sometimes called sonography, voiceprints, or voicegrams. When the data is represented in a 3D plot they may be called waterfalls. Spectrograms are used extensively in the fields of music, sonar, radar, and speech processing, seismology, and others. Spectrograms of audio can be used to identify spoken words phonetically. A spectrogram can be generated by an optical spectrometer, a bank of band-pass filters, by Fourier transform or by a wavelet transform (in which case it is also known as a scaleogram).
We used the TTS Tacotron2 where we removed the embedding layer and did modifications inspired by Translatotron [20] a direct speech to-speech translation model which is trained end-to-end. The Translatotron is a sequence-to-sequence encoder stack maps 80-channel log-mel spectrogram input features into hidden states which are passed through an attention-based alignment mechanism to condition an autoregressive decoder, which predicts 1025-dim log spectrogram frames corresponding to the translated speech. Two optional auxiliary decoders, each with their own attention components, predict source and target phoneme sequences.

To produce a waveform, we need both the magnitude and the phase components. Since our model does not predict phase, we use our predicted magnitude and apply a Griffin-Lim phase recovery [21] to generate the final waveform of the

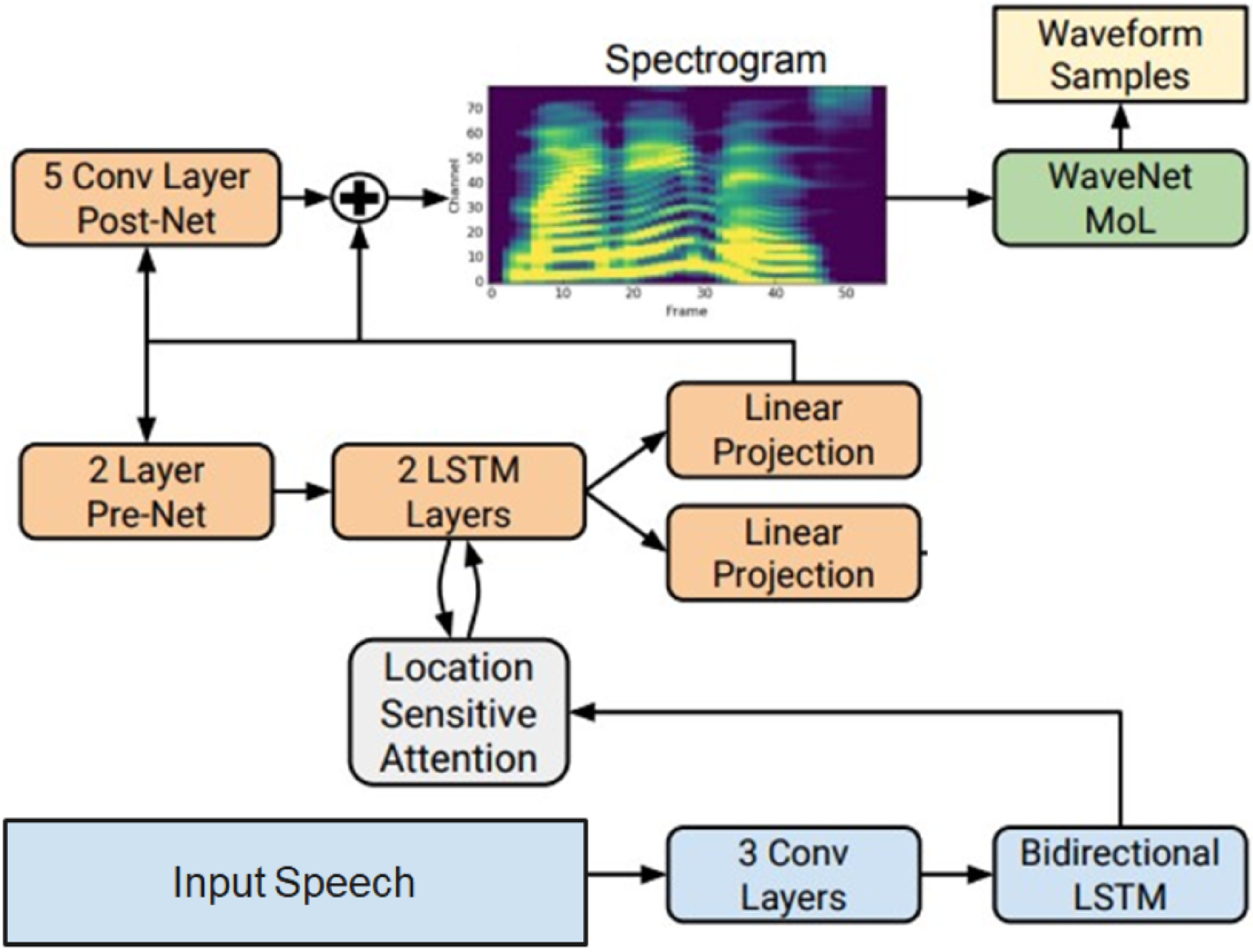

Figure 3.4: Block diagram of the Tacotron 2 system architecture.

translated sentence.

We ran our experiment on QCRI HPC node equipped with 4 NVIDIA Tesla V100 GPUs with 16 GB memory, and 20 cores of Xeon(R) E5-2690 CPUs.

Table 3.4: Experiment & Evaluation.

network	**loss**	**mse**	**mae**	**val loss**	**val mse**	**val mae**	**batch size**
Tacotron	**0.0722**	**0.0102**	**0.0722**	**0.5211**	**0.3315**	**0.5211**	**64**
Tacotron	0.1958	0.1727	0.1958	4.1904	3.2147	4.1904	32

3.4 Conclusion

In this work, we introduced an unsupervised approach to creating parallel speech corpora from dubbed videos. Unlike existing approaches that are either supervised, or unsupervised but inefficient, the proposed approach can be tuned to produce large corpora with high quality. For future work, we plan to apply the approach on many dubbed videos in different languages and demonstrate the effectiveness of the produced corpora when used in speech technology applications.

3.5 Future Work

We plan to apply the approach on many dubbed videos in different languages and demonstrate the effectiveness of the produced corpora when used in speech technology applications. We also plan to increase the number of hours to feed them to the architecture we applied in this section 3.3 We would also improve the architecture we used in section 3.3 there are a lot of experiments and improvements that could be applied to the whole pipeline.

Bibliography

[1] A. Buccianti, “Dubbed turkish soap operas conquering the arab world: social liberation or cultural alienation?,” *Arab Media and Society*, vol. 10, pp. 4–28, 2010.

[2] A. Öktem, M. Farrús, and L. Wanner, “Automatic extraction of parallel speech corpora from dubbed movies,” in *Proceedings of the 10th Workshop on Building and Using Comparable Corpora (BUCC); 2017 30 July-4 Aug; Vancouver, Canada.[Unknown place]: ACL, 2017. p. 31-5.*, ACL (Association for Computational Linguistics), 2017.

[3] A. Tsiartas, P. Ghosh, P. G. Georgiou, and S. Narayanan, “Bilingual audio-subtitle extraction using automatic segmentation of movie audio,” in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5624–5627, IEEE, 2011.

[4] A. Saboo and T. Baumann, “Integration of dubbing constraints into machine translation,” in *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pp. 94–101, 2019.

[5] A. Öktem, M. Farrús, and A. Bonafonte, “Bilingual prosodic dataset compilation for spoken language translation,” *IberSpeech 2018; 2018 Nov 21-23; Barcelona, Spain. Baixas, France: ISCA; 2018. p. 20-4.*, 2018.

[6] M. Federico, Y. Virkar, R. Enyedi, and R. Barra-Chicote, “Evaluating and optimizing prosodic alignment for automatic dubbing,” in *Proceedings of Interspeech*, p. 5, 2020.

[7] A. Öktem, M. Farrús, and A. Bonafonte, “Prosodic phrase alignment for machine dubbing,” *arXiv preprint arXiv:1908.07226*, 2019.

[8] B. Premjith, S. S. Kumar, R. Shyam, M. A. Kumar, and K. Soman, “A fast and efficient framework for creating parallel corpus,” *Indian J. Sci. Technol*, vol. 9, pp. 1–7, 2016.

[9] W. Feng, H. Ren, X. Li, and H. Guo, “Building a parallel corpus with bilingual discourse alignment,” in *Workshop on Chinese Lexical Semantics*, pp. 374–382, Springer, 2017.

[10] T. Kajiwara and M. Komachi, “Building a monolingual parallel corpus for text simplification using sentence similarity based on alignment between word embeddings,” in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 1147–1158, 2016.

[11] S. Khosla and H. Acharya, “A survey report on the existing methods of building a parallel corpus.,” *International Journal of Advanced Research in Computer Science*, vol. 9, no. 4, 2018.

[12] W. A. Gale and K. Church, “A program for aligning sentences in bilingual corpora,” *Computational linguistics*, vol. 19, no. 1, pp. 75–102, 1993.

[13] W. Liu, Z. Chang, and W. J. Teahan, “Experiments with a ppm compression-based method for english-chinese bilingual sentence alignment,” in *Interna-*

tional Conference on Statistical Language and Speech Processing, pp. 70–81, Springer, 2014.

[14] J. Tiedemann, "Improved sentence alignment for building a parallel subtitle corpus: Building a multilingual parallel subtitle corpus," *LOT Occasional Series*, vol. 7, pp. 147–162, 2007.

[15] J. Cuřín, M. Čmejrek, J. Havelka, and V. Kuboň, "Building a parallel bilingual syntactically annotated corpus," in *International Conference on Natural Language Processing*, pp. 168–176, Springer, 2004.

[16] M. Post, C. Callison-Burch, and M. Osborne, "Constructing parallel corpora for six indian languages via crowdsourcing," in *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pp. 401–409, 2012.

[17] E. Avramidis, M. Ruiz Costa-Jussà, C. Federmann, M. Melero, P. Pecina, and J. Van Genabith, "A richly annotated, multilingual parallel corpus for hybrid machine translation," in *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pp. 2189–2193, European Language Resources Association (ELRA), 2012.

[18] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, "scikit-image: image processing in python," *PeerJ*, vol. 2, p. e453, 2014.

[19] D. Doukhan, J. Carrive, F. Vallet, A. Larcher, and S. Meignier, "An open-source speaker gender detection framework for monitoring gender equality," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5214–5218, IEEE, 2018.

[20] Y. Jia, R. J. Weiss, F. Biadsy, W. Macherey, M. Johnson, Z. Chen, and Y. Wu, “Direct speech-to-speech translation with a sequence-to-sequence model,” *arXiv preprint arXiv:1904.06037*, 2019.

[21] S. Nawab, T. Quatieri, and J. Lim, “Algorithms for signal reconstruction from short-time fourier transform magnitude,” in *ICASSP’83. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 8, pp. 800–803, IEEE, 1983.