German-Arabic Speech-to-Speech Translation for Psychiatric Diagnosis

Juan Hussain, Mohammed Mediani, †Moritz Behr, Sebastian Stüker
KIT - Karlsruhe Institute of Technology
firstname.lastname@kit.edu
†uceul@student.kit.edu

Mohamed Amine Cheragui
LDDI, Ahmed Draia University - Adrar
m_cheragui@univ-adrar.edu.dz

Alexander Waibel
Carnegie Mellon University
alexander.waibel@cmu.edu

Abstract

In this paper we present the Arabic related natural language processing components of our German–Arabic speech-to-speech translation system which is being deployed in the context of interpretation during psychiatric, diagnostic interviews. For this purpose we have built a pipe-lined speech-to-speech translation system consisting of automatic speech recognition, machine translation, text post-processing, and speech synthesis systems. We have implemented two pipelines, from German to Arabic and vice versa, to conduct interpreted two-way dialogues between psychiatrists and potential patients. All systems in our pipeline have been realized as all-neural end-to-end systems, using different architectures suitable for the different components. The speech recognition systems use an encoder/decoder + attention architecture, the machine translation system is based on the Transformer architecture, the post-processing for Arabic employs a sequence-tagger for diacritization, and for the speech synthesis systems we use Tacotron 2 for generating spectrograms and WaveGlow as a vocoder. The speech translation is deployed in a server-based speech translation application that implements a turn-based translation between a German-speaking psychiatrist administrating the Mini-International Neuropsychiatric Interview (M.I.N.I.) and an Arabic speaking person answering the interview. As this is a very specific domain, in addition to the linguistic challenges posed by translating between Arabic and German, we also focus in this paper on the methods we implemented for adapting our speech to speech translation system to the domain of this psychiatric interview.

1 Introduction

In psychiatry the Mini-International Neuropsychiatric Interview (M.I.N.I.) is a short structured diagnostic interview for psychiatric disorders (Sheehan et al., 1998). In Germany it is, among others, used for diagnosing Arabic speaking refugees. Here, the language barrier is an obvious one, that is normally overcome with the help of human interpreters. However, human interpreters are scarce, expensive and very often not readily available when an urgent diagnosis is needed. In the project *Removing language barriers in treating refugees—RELATER* we are therefore building a speech-to-speech translation (S2ST) system for interpreting between a German speaking psychiatrist and an Arabic speaker taking the M.I.N.I. interview.

The natural language processing (NLP) technology for this scenario faces two challenges: a) the general linguistic challenges when translating between German and Arabic and b) the specific domain of the interview for which only very little adaptation data is available.

In this paper, we describe the Arabic related NLP components with which we implemented the speech to speech translation between German and Arabic for the interview. These components are part of a server-based application, where the client application is capable of running on mobile platforms; whereas the components themselves run on remote powerful computation servers. The application realizes a turn-based interpretation system between the psychiatrist and the potential patient.

While end-to-end S2ST translation systems are the latest trend in research, for our system we opted for pipe-lined systems, because a) pipe-lined systems still outperform end-to-end systems (Ansari et al., 2020), and b) to the best of our knowledge no suitable corpus for training German–Arabic end-to-end S2ST systems exists, for any domain.

All systems in our pipeline have been realized as all-neural end-to-end systems, using different architectures suitable for the different components. The speech recognition system uses an encoder/decoder + attention architecture (Nguyen et al., 2020b), the text segmentation component, and the machine translation are based on the Transformer architecture (Vaswani et al., 2017a), and for the speech synthesis we use Tacotron 2[1] (Shen et al., 2018) for generating spectrograms and WaveGlow as vocoder (Prenger et al., 2019).

The rest of the paper is structured as follows: Section 2 describes the Arabic speech recognition system in the pipeline, while section 3 describes the machine translation component and section 4 the Arabic speech synthesis system. As the the topic of diacritization is very prominent for Arabic NLP, we discuss the specific issues we faced in our S2ST system in section 5. In section 6, we study the real-time aspect of our pipeline system which is essential for the deployment.

2 Automatic Speech Recognition

For Automatic Speech Recognition (ASR), we employ an LSTM-based sequence to sequence (S2S) model with encoder/decoder + attention architecture. This model yields better performance compared to the self-attention S2S model with a comparable parameter size. Before the LSTM layers in the encoder, we place a two-layer Convolutional Neural Network (CNN) with 32 channels and a time stride of two to down-sample the input spectrogram by a factor of four. In the decoder, we adopt two layers of unidirectional LSTMs and the approach of Scaled Dot-Product (SDP) Attention to generate context vectors from the hidden states. More details about the models can be found in (Nguyen et al., 2020b). In section 2.1, we list the data used for the training, testing, and domain adaptation. For the latter, we use primary methods described in section 2.2. We report some implementation details and experimental results in section 2.3.

2.1 Data

We use the following data sets. For each data set we give a short name and the duration in hour (h) or in minutes (m):

- **`Alj.1200h`**: We use this set for the training or bootstrapping of our model. It consists of 1200 hours of broadcast videos recorded during 2005–2015 from the Aljazeera Arabic TV channel as described in (Ali et al., 2016). As reported, 70% of this set is in Modern Standard Arabic (MSA) and the rest is Dialectal Arabic (DA), such as Egyptian (EGY), Gulf (GLF), Levantine (LEV), and North African (NOR). The categories of the speech range from conversation (63%), interview (19%), to report (18%).

- **`Alj.MSA+dialect.10h`**: A test set of 10 hours described in (Ali et al., 2016) as well. It includes non-overlapped speech from Aljazeera, which was prepared according to (Ali et al., 2016) for an Arabic multi-dialect broadcast media recognition challenge. We use the set as it is without normalizing Alif, Hamza or any other characters.

- **`Alj.MSA.2h`**: This is a subset from `Alj.MSA+dialect.10h` where we cut only MSA utterances free from dialects from the beginning of the set until we reached the duration of 2 hours.

- **`mini.que.ans.3.34h`**: This dataset consists of 915 utterances and 3.34 hours of reading M.I.N.I questions (Sheehan et al., 1998) and free answers from two speakers. We transcribed the answers with our ASR system and then corrected them manually with our desktop application DaCToR (Hussain et al., 2020) which we used to correct the automatic transcription. For the recording we employed our online application TEQST[2] which allows the user to read texts and record with their own mobile devices.

- **`mini-ans.42m`**: A test set that has been processed similarly to `mini.que.ans.3.34h`. It consists of 224 free answers on M.I.N.I questions with a duration of 42 minutes by one speaker.

[1] `https://github.com/NVIDIA/tacotron2`
[2] `https://github.com/TEQST/TEQST`

- **mini.ques.50m**: 225 M.I.N.I questions by the same speaker as from mini-ans.42m with a duration of 50 minutes.

2.2 Domain Adaptation for ASR

As we will see in section 2.3, we obtain extremely poor performance on our target domain (psychiatric interview). Therefore, we are investigating many approaches to adapt our speech recognition system to the target domain. In this paper, we report about two experiments: the mixed-fine-tuning and the fine-tuning experiment (Chu et al., 2017). For mixed-fine-tuning, we begin with the pre-trained model on Alj.1200h data as a baseline and mixed-fine-tune it on the data resulting from mixing Alj.1200h with the domain data mini.que.ans.3.34h. For mixed-fine-tune or fine-tuning the decoder, we freeze the encoder, and train only the decoder. For fine-tuning, we don't mix the data. Instead we use only mini.que.ans.3.34h set. For mixed-fine-tuning and fine-tuning, We use the learning rates (0.0009) and (0.00001) respectively. We report the result in section 2.3.

2.3 Experimental Results

We experimented with the LSTM-based encoder-decoder + attention using 40 log-Mel features, two-layer Convolutional Neural Network (CNN) with 32 channels, 6 encoder-layers, 2 decoder-layers. For data augmentation, we use *dynamic time stretching* and *specAugument* as described in (Nguyen et al., 2020b).

For the output, we employed a sub-word tokenization method, byte-pair encoding (BPE) (Sennrich et al., 2015), (Gage, 1994). Empirical experiments indicated that 4k tokens yielded the best performance. The tokenizer is trained on MSA texts since we aim to have a well-performing system on MSA. For this reason, we obtain on the first test set MSA+dialect.10h, which contains dialect besides MSA, Word Error Rate (WER) of 18.8% (see Table 1). While, On the second test set MSA.2h with only MSA data, we reach a WER of 12.6%. We employ the beam-search algorithm for the output sequence prediction, where the beam-size of 4 yields the best WER. This low beam-size is considered very efficient for the real-time capability of the system.

For **Domain adaption**, the results of the main model on both test sets mini-ans.42m and mini.ques.50m of the target domain have a very high WER: 40% and 30.4% respectively. Our speech recognition model is S2S without an additional language model scoring or a dictionary. The decoder learns the language model from the training data directly. The out-of-vocabulary (OOV) rate between the test sets and the training data Alj.1200h is 3%. Hence, The main reason of the high WER is not the OOV but the domain mismatch since the training data is from broadcast videos and the domain of both test sets is psychiatric interviews. The results in the next section gives evidence for this hypothesis, since tuning only the decoder yields a considerable improvement.

By mixed-fine-tuning the whole model with the same architecture, we reach an improvement on both domains, where the WER is reduced by 0.7% on both Alj.MSA+dialect.10h and Alj.MSA.2h. Besides, on mini-ans.42m and mini.ques.50m we obtain a WER reduction of about 22%. By mixed-fine-tuning only the encoder we obtain comparable results. On the other hand, although fine-tuning the decoder only causes the forgetting of the out-of-domain (i.e. Alj.MSA+dialect.10h and Alj.MSA+dialect.10h) by increasing their WER by about 8%, it improves the in-domain (i.e. mini.ques.50m) to 6.2%. This is likely due to the questions being identical in the training and test set, however by different speakers. The reason of forgetting is that the fine-tuning does not use the whole training but only the in-domain data.

3 Machine Translation

It is a known fact that translating between structurally or morphologically different languages is a very difficult task. Two well-known examples of these hard language pairs are English–German and English–Arabic. In this work, we are associating the worst of these two worlds: Arabic and German. An example of the unmistakable differences between these two languages is that both of them are morphologically rich: Arabic is highly inflectional ((Farghaly and Shaalan, 2009)) and German possesses word com-

test set	baseline	mixed-FT	Dec mixed-FT	Dec FT
Alj.MSA+dialect.10h	18.8	18.1	18.6	25.7
Alj.MSA.2h	12.6	11.8	12.3	20.5
mini-ans.42m	40.0	16.4	18.4	14.8
mini.ques.50m	30.4	8.6	10.3	6.2

Table 1: ASR results, where FT stands for fine-tuning and dec for Decoder. The values are the Word Error Rate (WER) ↓ in percent

pounding. At the syntactic level, word order is a substantial difference between the two languages. Arabic is much more flexible in this respect.

In this work, we face two major challenges: data scarcity caused by the understudied language pair and the specificity of the domain of application. Indeed, the language pair under consideration here has been out of the focus of the international machine translation campaigns. Such campaigns (such as WMT[3] and IWSLT[4]) are organized on a yearly basis, and have boosted considerably both the performance and the available resources for the language pairs they consider. The data scarcity problem becomes even more severe when we know that the resulting system is to be involved in the communication between psychiatrists and their patients. The genre of the language used therein is quite different from that used in the news. This latter makes most of the available training data.

In the following, we explain the steps we followed to overcome these limitations and to produce a reasonable translation system. Then, we show the developed systems in action through empirical evaluations.

3.1 Data

Although the language pair under consideration is not commonly studied, a reasonable training set could be gathered, thanks to the different data sources publicly exposed on the Internet. In Part A of Table 2, we give a summary about the exploited data sets and some of their important attributes.

Corpus	Sent. Pairs ($\times 10^3$)	Words ($\times 10^6$)		Vocab ($\times 10^3$)	
		Ar	De	Ar	De
A: Training Data					
TED	199	2.800	3.200	278.0	214.0
Multi UN	165	4.700	4.800	165.0	121.0
News Commentary	208.783	7.565	6.396	226.779	228.481
Wikipedia	9.833	0.146	0.142	33.308	32.448
QED	18.537	0.124	0.146	24.796	20.184
JW300	358.501	5.805	5.479	215.926	255.981
GlobalVoices	9	0.150	0.170	43.0	35.0
Tatoeba	1	0.004	0.003	2.2	1.9
Q+A-TRAIN[5]	0.257	0.004	0.005	1.899	1.657
Total	**954.408**	**21.383**	**20.348**	-	-
B: Test Data					
TED-TEST	1.717	0.021	0.025	8.572	6.897
JW-TEST	1.974	0.031	0.029	10.455	9.245
Q+A-TEST[5]	0.129	0.002	0.002	1.209	1.060

Table 2: Summary of the translation data sets

The TED corpus[6] is our first source of data. This corpus is collected by (Cettolo et al., 2012) from the translations generated by volunteers for the TED talks, over the course of several years. By looking at the data, we think that TED is cleaner and more suitable for speech translation. Therefore, we always start by a baseline trained on TED data only, and then we gradually introduce more data from other sources.

[3] http://www.statmt.org/wmt20/

[4] http://iwslt.org/doku.php

[5] In-domain data

[6] https://wit3.fbk.eu/

Another extremely important source is the OPUS[7] repository. OPUS is a large depot where parallel data is collected from different sources and sometimes also preprocessed for a very large number of language pairs. It turned out that not all of the data available in this repository is in a good shape. Therefore, we manually examined those corresponding to our language pair, and discarded those of which we were convinced that they include very large amounts of noise. A good example of this noisy data is the `OpenSubtitle` corpus. After this manual filtering, the corpora used from OPUS are: `Multi UN`, `News Commentary`, `Global Voices`, and `Tatoeba`.

The `Wikipedia` corpus is generated from German–English and Arabic–English corpora by pivoting over their English side. Although OPUS repository offers these two corpora for download, it does not offer the direct version Arabic–German. We use strict string matching of the English sides to find the Arabic–German translations (i.e. Finding sentence pairs from the two corpora where the English sentences are exactly equal in both pairs).

The `QED` corpus is a multilingual corpus for the educational domain produced by QCRI in Qatar (Abdelali et al., 2014). Similar to TED talks, this corpus consists of the transcription and translation of educational lectures. In this work, we use the version 1.4 of this corpus.[8]

Finally, the `JW300` corpus was crawled from the Jehovah's Witnesses website[9] by (Agić and Vulić, 2019). The contents are of religious nature and are available in a large number of language pairs. This corpus is also made available on the OPUS repository. However, unlike the other corpora, this one comes unaligned on the sentence level. Consequently, we perform sentence alignment on the raw downloaded data. The process was held by the `hunalign`[10] tool. However, this tool requires a dictionary for the language pair; we automatically created one from the other aligned corpora. We used the `fast_align`[11] tool to align the corpora at the word level in two directions. Afterwards, We restricted our dictionary to word pairs appearing 5 times or more in the two directions.

The last row in Part A of Table 2 shows a part of our in-domain data. The other small part is used for testing (Last row, `Q+A-TEST` in Part B of the same Table). This consists of the translations of the M.I.N.I questions. Added to them are manual translations of the transcribed answers by some patients. These latter correspond to the `mini.que.ans.3.34h` data set (mentioned in speech data stes in Section 2.1).

In addition to the training data, we use three test sets to evaluate the performance of our systems on different kinds of data. Some details about these test sets are given in Part B of Table 2 (`JW-TEST`, `TED-TEST`, `Q+A-TEST`). The first two of these sets are considered as general domain and used to measure the model's performance on out-of-domain tasks. While these two sets are both out-of-domain for our purpose, they still represent different domains. The `JW-TEST` is a random subset drawn from the `JW300`, while ensuring that the test and the training sets remain disjoint. The `TED-TEST` set is the test set used in IWSLT evaluation in 2012 (IWSLT2012). The last test set (`Q+A-TEST`), as mentioned in the previous paragraph, consists of what was held out from the in-domain data to evaluate the system's performance on in-domain tasks. This latter test set, as well as its training counter-part (`Q+A-TRAIN`), is work in progress and will be extended in the future.

3.2 Domain Adaptation for Machine Translation

The similarity between training and test data is a common assumption in machine learning (See e.g. (Daumé and Marcu, 2006) for an elaborate description of this issue). In most cases however, this assumption is violated in the field. This issue is usually resolved by performing in-domain adaptation. That is tweaking a model trained on large quantities of out-of-domain data using only the very few available examples from the domain under consideration. As a result, the similarity between probability distributions of the model and the domain is augmented.

The tasks we perform in this work are no exception to the aforementioned problem. In order to adapt

[7] `http://opus.nlpl.eu/`
[8] `http://alt.qcri.org/resources/qedcorpus/`
[9] `https://www.jw.org/en/`
[10] `http://mokk.bme.hu/resources/hunalign/`
[11] `https://github.com/clab/fast_align`

the general system to the domain under consideration, some few in-domain examples are mandatory. In our work, these examples are the Q+A sets shown in the last rows of Parts A and B of Table 2. It is noteworthy however, that these sets are being actively extended, since the manual data creation is time-consuming.

So far, we explored two approaches to the adaptation: fine-tuning and data selection. Fine-tuning is accomplished by resuming the training for very few additional epochs using only the in-domain data. Data selection provides us with more in-domain data. This is achieved by choosing a special subset of training examples from the general domain training set. This subset consists of general domain examples, which are the most similar to the in-domain examples. The selection process is carried out using (Moore and Lewis, 2010) with more careful out-of-domain language model selection as proposed by (Mediani et al., 2014). The language models used in this procedure are 4-gram language models with Witten-Bell smoothing. We perform the selection for both languages (i.e. once for Arabic and once for German). Then, we take the intersection of the two selected subsets.[12]

3.3 Experimental Results

The input data is preprocessed using the `sentence_piece`[13] algorithm. For Arabic the vocabulary size is set to 4000 and for German to 16000. Lines longer than 80 words are discarded. The model's architecture is transformer encoder-decoder model (Vaswani et al., 2017b). Both the encoder and decoder are 8-layers. Each layer is of size 512. The inner size of the feed-forward network inside each layer is 2048. The attention block consists of 8 heads. The dropout is set to 0.2. All trainings are run for 100 epochs. While the number of epochs used for fine-tuning is taken to be 10. We use the Adam scheduling with a learning rate initially set to 1 and with 2048 warming steps. The results are summarized in Table 3.

System	JW-TEST	TED-TEST	Q+A-TEST
A: German → Arabic			
TED only	5.06	12.44	7.99
TED+Extra data	23.90	14.62	8.19
Fine-tuned	5.34	12.94	12.18
Fine-tune-select	23.18	14.35	19.16
B: Arabic → German			
TED+Extra data	17.17	9.67	6.18
Fine-tune-select	16.60	9.06	15.96

Table 3: Summary of the translation experiments (results are expressed as BLEU (↑) scores)

The scores in Table 3 are BLEU scores (Papineni et al., 2002). The table is subdivided into two panels, one for each translation direction. We report all tested combinations for the German → Arabic direction. For the reverse direction we report results only for the most promising configurations. The configurations shown here are as follows:

- TED only The training is performed on TED corpus only.
- TED+Extra data the system is trained on all data.
- Fine-tune The fine-tuning is done with the very small in-domain `Q+A-train` only.
- Fine-tune-select where the fine-tuning is accomplished using the set consisting of the merge between `Q+A-train` and the 20K more sentences selected from the training corpora.

[12] Other ways of combining selections from the two sides of the parallel corpus were not explored in this work. Our choice (i.e. the intersection) is motivated by our aim for higher precision.

[13] https://github.com/google/sentencepiece

As the table shows, using all the available data is helpful for all test sets. However, the JW-TEST is the one which benefits the most. This is to be expected as an important part from the training data comes from the JW300 corpus. That corpus was originally in the same set with JW-TEST. The fine tuning on the tiny Q+A-train data set was not a good help to all test sets. While it introduces small improvement on the in-domain test, it was very harmful to the other sets. It causes the system to quickly overfit; most likely due to its small size and very restricted type of sentences. This is where the data selection comes in handy to fix these problems of the in-domain data set. This is demonstrated on the last row of the two panels of the table. The selection was able to bring a large improvement for the in-domain test set, without harming the other test sets from different domains.

4 Speech Synthesis

For speech synthesis or Text-To-Speech (TTS), we use the state-of-the-art model, namely Tacotron 2[14] (Shen et al., 2018) which is a recurrent S2S network with attention that predicts a sequence of Mel spectrogram frames for a given text sequence. For generating time-domain waveform samples conditioned on the predicted Mel spectrogram frames, we chose WaveGlow (Prenger et al., 2019) where the authors claim that it has a faster inference than WaveNet (Oord et al., 2016) used with Tacotron 2 in (Shen et al., 2018).

4.1 Experiments

We trained Tacotron 2 and used a pre-trained WaveGlow model found on the Github page of WaveGlow. We also kept most of the default parameters from the implementation except for the sampling rate where we use 16kHz for the audio. As input to the Tacotron 2, we used Arabic character sequences with diacritics in Buckwalter format[15].

The corpus used for the training (Halabi, 2016) contains 1813 utterances with a duration of 3.8 hours. We first experimented with Arabic characters without diacritics to synthesize the audio directly from the MT output. Unfortunately, the model did not converge since even the same word in a different context is diacritized differently. Employing BPE used for ASR (section 2) without diacritics did not work either even with pretraining with over 400 hours of the data set Alj.1200h. For this reason we developed the diacritization component (section 5) to our pipeline.

The resulting speech sounds natural (see section 4.2), however, when synthesizing for only one word, the model does not terminate and reaches the maximum decoding steps. As a solution, we implemented data augmentation by both splitting and merging utterances. We apply splitting on 708 out of the total 1813 utterances which are in form of spoken separate words separated by silence, for instance:

"ta>aa˜wSawa˜ra - wata>aa˜wSara - watu&a˜wSa - taSa>aa˜w"

with an automatic approach. The algorithm simply finds positions with values under a threshold. These are simple to find since the corpus is recorded using a professional studio. Besides, we merge randomly 2 and 3 utterances with the probabilities 0.2 and 0.1 respectively if the duration stays under 30 seconds. This approach solved the one words synthesizing. However, another issue appeared, the synthesis contained short silence where it is not supposed to. This is due to the silence between the concatenated utterances which we solved by adding the symbol for silence "-" used already in the corpus between to concatenated utterances. It is worth to mention that this method solved an issue of the German synthesis for long sentences.

4.2 Results

For the evaluation of the naturalness of the resulting speech, we constructed an online form including 10 synthesized speech from the first 10 utterances of the test set Alj.MSA.2h (section 2.1) after diacritizing them manually. Besides, we added 3 real speech samples from the training data to judge the seriousness of the evaluators and whether the task has been properly understood by the participants. The scale we use is 1 if the speaker is a robot, 2 near to a robot sound, 3 no difference, 4 near to human sound,

[14]The source code is available at https://github.com/NVIDIA/tacotron2

[15]Buckwalter transliteration can be found here: http://www.qamus.org/transliteration.htm

and 5 the sound is from a human. From the 18 participations we received, we deleted the whole evaluation of participants who evaluated one or more of the 3 real audio samples with 2 or below. It remains 11 valid votes with an average of 4.05 as a Mean Of Opinion (MOP ↑). For the whole 18 participations without deleting any invalid votes we obtain a MOP of 3.82. This means that the synthesized speech is of very good quality, in most cases judged as very close to human speech.

5 Diacritization

As mentioned earlier (Section 4), the dicritization was introduced to make the TTS understandable. We were, indeed, unable to synthesize good Arabic speech without these short vowels. The synthesis, then, consists of incomprehensible mumbles as the system tends to insert the vowels arbitrarily.

Diactritics have a crucial role in giving the Arabic text a phonetics (Abbad and Xiong, 2020). Moreover, they allow for better comprehension by reducing the level of ambiguity inherent in the Arabic transcription system. Having that said, most modern written Arabic resources omit these diacritics and rely on the ability of the native speakers to guess them from the context. In particular, all the parallel data used to train our translation systems has no diacritics.

We employ a sequence-tagger trained using the Flair-framework (Akbik et al., 2018), which is a BiLSTM-CRF proposed by (Huang et al., 2015). besides (Akbik et al., 2018) introduces Contextual String Embeddings. Thereby, sentences are passed as sequences of characters into a character-level Language Model (LM) to form word-level embeddings. This is done by concatenating the output hidden state after the last character in the word from the forward LM and the output hidden state before the word's first character from the backward LM. These two language models result from utilizing the hidden states of the forward-backward recurrent neural network (see (Akbik et al., 2018) for more details about the approach).

5.1 Data

Fortunately, some available Arabic resources are diacritized. Al-Shamela corpus (Belinkov et al., 2016) is a large-scale, historical corpus of Arabic of about 1 billion words from diverse periods of time. It is important to specify that the corpus in its initial state will not be exploitable in our resolution process, for this reason, a series of operations will be carried out, as follows:

- Delete empty lines and lines with a single character and keep only lines with a high amount of vowels ($> 40\%$ of characters are short vowels)
- Split words to obtain pairs consisting of (letter, Diacritic Mark).

Applying these steps to the Al-Shamela corpus gives an amount of around 5 million lines of training data. We used only a part of this large corpus (≈ 1.5 million lines). Some other parts of this corpus were held out for validation and testing (1300 validation set and 8000 lines for testing).

5.2 Results

We use the sequence to sequence biLSTM provided by the Flair framework. We kept the configuration as simple as possible. The embeddings are obtained by stacking forward and backward embeddings of size 512 each. They were computed while training a one layer language model. The network consists of an RNN encoder of 1 layer and 128 hidden states and a CRF decoder. With such a configuration, an accuracy of 95.34% was achieved on the held out test set.

6 Real-Time Aspects

All components of the pipeline reside on only one GPU of type `TITAN RTX` with 24 GB memory. The number of components is 7 where there are 3 for each S2ST direction plus the diacritization component for Arabic side.

We measured the latency of the system components by using 5 Arabic sentences from the test set `Alj.MSA.2h` consisting of 16, 9, 8, 12 and 9 words, hence 54 words in total. For ASR, we use

the synthesis of this sentences which has the total duration of 47.85 seconds. The ratio of the total duration of processing the 5 sentences by the whole pipeline system to the total input speech duration is $\frac{7.95}{47.85} = 0.17$. This means that the processing time is about 6 times faster than the input speech duration. Hence, the system is real-time capable and Long sentences can be chunked into smaller parts in a stream-like output (see (Nguyen et al., 2020a) as an example of using The LSTM-based model for ASR to process a stream). Table 4 shows the time duration in total for the 5 sentences, the average time per sentence, per word, and the model size on the GPU.

component	total time	avg. per sentence	per word	GPU model size (GB)
ASR	1.06	0.212	0.020	1.4
MT	1.90	0.381	0.035	0.9
Diacritization	0.68	0.136	0.014	0.8
TTS	4.31	0, 861	0.080	1.7

Table 4: The measurement of the latency for each component of the pipeline. Times are in seconds(↓). The measurements are for 5 sentences with (16,9,8,12,9) words, i.e 54 words in total

It can also be noted from the Table that the TTS is the slowest component in the pipeline.

7 Conclusion

In this work, we presented the different components of our German–Arabic speech-to-speech translation system which is being deployed in the context of interpretation during psychiatric, diagnostic interviews. For this purpose, we have built a pipe-lined speech-to-speech translation system consisting of automatic speech recognition, machine translation, text post-processing, and speech synthesis systems. We also described the problems we faced while building these components and how we proceeded to overcome them.

The speech recognition system uses an LSTM-based encoder/decoder + attention architecture, the machine translation component is based on the Transformer architecture, the post-processing for Arabic employs a sequence-tagger for diacritization, and for the speech synthesis systems we use Tacotron 2 for generating spectrograms and WaveGlow as a vocoder.

For domain adaptation, we used fine-tuning and mixed-fine-tuning on hand-created in-domain data. We achieved thereby considerable improvements despite the scarce collected domain data. This result confirms the already well established fact about the extreme importance of in-domain data. Therefore, collecting more of this domain-related data is on the top of our priorities in the near future. Certainly, this data collection process will not prevent us from trying more ways to exploit the data we already have. For instance, more efforts will be put into exploring other adaptation techniques. Additionally, we are investigating the exploitation of the large quantities of monolingual data in the translation components.

Acknowledgements

The work in this paper was funded by the German Ministry of Education and Science within the project *Removing language barriers in treating refugees—RELATER*, no. 01EF1803B.

References

Hamza Abbad and Shengwu Xiong. 2020. Multi-components system for automatic arabic diacritization. In *European Conference on Information Retrieval*, pages 341–355. Springer.

Ahmed Abdelali, Francisco Guzman, Hassan Sajjad, and Stephan Vogel. 2014. The amara corpus: Building parallel language resources for the educational domain. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may. European Language Resources Association (ELRA).

Željko Agić and Ivan Vulić. 2019. JW300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3204–3210, Florence, Italy, July. Association for Computational Linguistics.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649.

Ahmed Ali, Peter Bell, James Glass, Yacine Messaoui, Hamdy Mubarak, Steve Renals, and Yifan Zhang. 2016. The mgb-2 challenge: Arabic multi-dialect broadcast media recognition. In *2016 IEEE Spoken Language Technology Workshop (SLT)*, pages 279–284. IEEE.

Ebrahim Ansari, Amittai Axelrod, Nguyen Bach, Ondřej Bojar, Roldano Cattoni, Fahim Dalvi, Nadir Durrani, Marcello Federico, Christian Federmann, Jiatao Gu, Fei Huang, Kevin Knight, Xutai Ma, Ajay Nagesh, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Xing Shi, Sebastian Stüker, Marco Turchi, Alexander Waibel, and Changhan Wang. 2020. FINDINGS OF THE IWSLT 2020 EVALUATION CAMPAIGN. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 1–34, Online, July. Association for Computational Linguistics.

Yonatan Belinkov, Alexander Magidow, Maxim Romanov, Avi Shmidman, and Moshe Koppel. 2016. Shamela: A large-scale historical arabic corpus. *arXiv preprint arXiv:1612.08989*.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy, May.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of domain adaptation methods for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 385–391.

III Hal Daumé and Daniel Marcu. 2006. Domain Adaptation for Statistical Classifiers. *J. Artif. Int. Res.*, 26(1):101–126, May.

Ali Farghaly and Khaled Shaalan. 2009. Arabic natural language processing: Challenges and solutions. *ACM Transactions on Asian Language Information Processing*, 8(4), December.

Philip Gage. 1994. A new algorithm for data compression. *C Users Journal*, 12(2):23–38.

Nawar Halabi. 2016. *Modern standard arabic phonetics for speech synthesis*. Ph.D. thesis, University of Southampton.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Juan Hussain, Oussama Zenkri, Sebastian Stüker, and Alex Waibel. 2020. Dactor: A data collection tool for the relater project. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6627–6632.

Mohammed Mediani, Joshua Winebarger, and Alexander Waibel. 2014. Improving In-Domain Data Selection for Small In-Domain Sets.

Robert C. Moore and William D. Lewis. 2010. Intelligent Selection of Language Model Training Data. In *ACL (Short Papers)*, pages 220–224.

Thai-Son Nguyen, Ngoc-Quan Pham, Sebastian Stueker, and Alex Waibel. 2020a. High performance sequence-to-sequence model for streaming speech recognition. *arXiv preprint arXiv:2003.10022*.

Thai-Son Nguyen, Sebastian Stüker, Jan Niehues, and Alex Waibel. 2020b. Improving sequence-to-sequence speech recognition training with on-the-fly data augmentation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7689–7693. IEEE.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July. Association for Computational Linguistics.

Ryan Prenger, Rafael Valle, and Bryan Catanzaro. 2019. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621. IEEE.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

David V. Sheehan, Yves Lecrubier, K. Harnett Sheehan, Patricia Amorim, Juris Janavs, Emmanuelle Weiller, Thierry Hergueta, Roxy Baker, and Geoffrey C. Dunbar. 1998. The mini-international neuropsychiatric interview (m.i.n.i.): The development and validation of a structured diagnostic psychiatric interview for dsm-iv and icd-10. *The Journal of Clinical Psychiatry*, 59(20):22–33.

Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. 2018. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017a. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017b. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.