Neural Speech Translation at AppTek

Evgeny Matusov, Patrick Wilken, Parnia Bahar, Julian Schamper*, Pavel Golik, Albert Zeyer*,
Joan Albert Silvestre-Cerdà+, Adrià Martínez-Villaronga+, Hendrik Pesch, and Jan-Thorsten Peter*

Applications Technology (AppTek), Aachen, Germany
{ematusov,pwilken,pbahar,jschamper,pgolik,jsilvestre,amartinez,hpesch,jtpeter}@apptek.com
*Also RWTH Aachen University, Germany +Also Universitat Politècnica de València, Spain

Abstract

This work describes AppTek's speech translation pipeline that includes strong state-of-the-art automatic speech recognition (ASR) and neural machine translation (NMT) components. We show how these components can be tightly coupled by encoding ASR confusion networks, as well as ASR-like noise adaptation, vocabulary normalization, and implicit punctuation prediction during translation. In another experimental setup, we propose a direct speech translation approach that can be scaled to translation tasks with large amounts of text-only parallel training data but a limited number of hours of recorded and human-translated speech.

1. Introduction

AppTek participated in the evaluation campaign of the International Workshop on Spoken Language Translation (IWSLT) 2018 with the goal of obtaining best possible speech translation quality by streamlining the interface between ASR and machine translation (MT). We tested a new way of encoding multiple hypotheses of ASR as input to an NMT system. We also experimented with a novel direct neural translation model that translates source language speech into target language text, while at the same time benefiting from text-only parallel training data in a multi-task learning framework. To make these experiments possible, we made sure that our NMT system can handle different types of input, and its source language vocabulary is harmonized w.r.t. the ASR system vocabulary. We also fine-tuned the NMT model on ASR-like noise, making it more robust against recognition errors. Finally, we tested different punctuation prediction approaches and found that the implicit prediction of punctuation marks by the MT component works best in our setting.

Although improving our state-of-the-art NMT model was not our main focus, we benefited from fine-tuning the model on the in-domain data, as well as from ensembles of models which differ in architecture – recurrent neural network (RNN) model with attention [1] or transformer architecture [2] – and/or input modality – ASR confusion network (CN) or first-best ASR output. This paper is organized as follows. We start by reviewing related work in 2, pointing out some differences and novelties in our approach. In Section 3, we describe our methods for data filtering, preprocessing, and punctuation prediction. Section 4 gives an overview of our ASR system. Section 5 describes the details of AppTek's NMT system. Section 6 gives details of how ASR confusion networks can be encoded as an input of the NMT system. In Section 7, we describe our direct speech translation prototype. The results of our speech translation experiments are summarized in Section 8.

2. Related Work

Theoretical background for tighter coupling of statistical ASR and MT systems had been first published in [3]. In practice, it was realized e. g. as statistical phrase-based translation of ASR word lattices with acoustic and language model scores [4] or confusion networks with posterior probabilities [5]. In both cases moderate improvements of translation quality were reported when the ASR scores were included in the log-linear model combination; the improvements were larger when the baseline recognition quality was low.

In the first publication on word lattice translation using a neural model [6], the proposed lattice-to-sequence model had an encoder component with one hidden state for each lattice node, as well as attention over all lattice nodes. This is a different and more computationally expensive model as compared to what we propose in this work. In our encoder, the number of hidden states is the same as the number of slots in the input confusion network, which is usually only slightly higher than the number of words in the utterance.

Adapting the NMT system to ASR-like noise was proposed by [7]. We follow the same strategy, but the noise that we introduce is not random; it is sampled from a distribution of most common ASR errors based on statistics from recognizing the audio of the TED dataset.

Direct translation of foreign speech was proposed by [8], who used a character-level sequence-to-sequence model[1]. They report experimental results on a small (163 hours of speech with transcriptions and translations) Spanish-to-English Fisher and Callhome dataset. The authors use multi-task learning with a single speech encoder and two decoders, one for English (direct translation) and one for Spanish, which allows them to incorporate supervision from Spanish

[1] A later work by [9] extends the approach of [8] to word-level models.

transcripts. In contrast, we follow the opposite approach, in which we have a single target language decoder and two separate encoders, one for source language speech, and one for source language text. This approach allows us to benefit from large quantities of text-only parallel MT training data, in a multi-task learning scenario, and thus, in contrast to previous work, to potentially compete with the standard approach that uses strong, but separate components for ASR and MT.

Punctuation prediction in MT (and especially neural MT) context was investigated in comparative experiments in [10]. Similarly to that paper, we also confirmed experimentally that implicit prediction of punctuation marks by the NMT system resulted in the best BLEU and TER scores in our setting (see Section 3.3 for details).

3. Data Preparation

3.1. Parallel Data Filtering

In line with the evaluation specifications, we used the TED corpus, the OpenSubtitles2018 corpus [11], as well as the data provided by the WMT 2018 evaluation (Europarl, ParaCrawl, CommonCrawl, News Commentary, and Rapid) as the potential training data for our NMT system, amounting to 65M lines of parallel sentence-aligned text. We then filtered these data based on several heuristics, with the two most important ones described next.

Since especially the crawled corpora are very noisy, they often contain segments in a wrong language, or even things like programming code and XML markup. We used the CLD2 library[2] for sentence-level language identification to keep only those sentence pairs in which the source sentence was labeled as English and target sentence as German with the confidence of at least 90%.

Another heuristic was based on sentence length: we only kept sentences with at least 3 and at most 80 words (after tokenization). We also removed sentence pairs in which source and target sentence lengths differ by a factor of 5 or more.

Overall the filtering yielded a corpus of 37.6M lines and 556M words (on the English side, counted untokenized), which we used in all of the experiments presented in this paper. It included 256K unique lines of TED talks with 4.4M words on the English side.

3.2. Preprocessing

We used two types of preprocessing. The first one was the standard Moses tokenization [12] for text translation and lowercasing on the English side. The German side was truecased using a frequency-based method. The second preprocessing was used only for English with the goal of converting text into speech transcript similar to the one produced by the ASR system. Starting from the Moses tokenization, we removed all punctuation marks and spliced back contractions (e. g. `do n't` → `don't`) to match the corresponding tokens in the ASR lexicon. We also converted numbers written with digits to their spoken form using a tool based on the *num2words*[3] python library.

The final step for both types of preprocessing was segmentation into sub-word units with byte pair encoding (BPE) [13], separately for each language. We used 20K merging operations. During testing, we used the option to revert BPE merge operations resulting in tokens that were observed less than 50 times in the segmented training data.

3.3. Punctuation Prediction

To translate a speech transcript with an NMT system trained with the first, standard preprocessing described above, we need to automatically enrich it with punctuation marks. To this end, we trained a RNN for punctuation restoration similar to the one presented in [14]. Only the words in a sentence are used to predict punctuation marks (period, comma, and question mark only). The acoustic features are not used.

For the setup with the ASR-like preprocessing of English, punctuation prediction is done implicitly during translation, since the target side of the training corpus contains punctuation marks. Thus, the output of the ASR system can be directly used (after BPE) as input to the NMT system.

4. ASR system

The ASR system is based on a hybrid LSTM/HMM acoustic model [15, 16], trained on a total of approx. 390 hours of transcribed speech from the TED-LIUM corpus (excluding the black-listed talks) and the IWSLT Speech-Translation TED corpus[4]. We used the pronunciation lexicon provided with the TED-LIUM corpus. The acoustic model takes 80-dim. MFCC features as input and estimates state posterior probabilities for 5000 tied triphone states. It consists of 4 bidirectional layers with 512 LSTM units for each direction. Frame-level alignment and state tying were obtained from a bootstrap model based on a Gaussian mixtures acoustic model. We trained the neural network for 100 epochs by minimizing the cross-entropy using the Adam update rule [17] with Nesterov momentum and reducing the learning rate following a variant of the Newbob scheme.

The language model for the single-pass HMM decoding is a simple 4-gram count model trained with Kneser-Ney smoothing on all allowed English text data (approx. 2.8B running words). The vocabulary consists of the same 152k words from the training lexicon and the out-of-vocabulary rate is 0.2% on `TED.dev2010` and 0.5% on `TED.tst2015`. The LM has a perplexity of 133 on

[2] https://github.com/CLD2Owners/cld2

[3] https://github.com/savoirfairelinux/num2words

[4] We realized that the provided audio-to-source-sentence alignments of the TED talks were often not correct. As this could significantly degrade the performance of the audio encoder for the direct speech translation approach described in Section 7, we had to automatically recompute these alignments by force-aligning each TED recording to its corresponding source sentences, and applied heuristics to overcome the problem of transcription gaps (speech segments without a translation in the parallel data).

TED.dev2010 and 122 on TED.tst2015.

Since TED talks are a relatively simple ASR task, we decided not to proceed with sequence training of the acoustic model or LM rescoring with LSTM models in order to have more uncertainty in the lattices. Acoustic training of the baseline model and the HMM decoding were performed with the RWTH ASR toolkit [18]. We trained BLSTM models with RETURNN [19], which integrates into RWTH ASR as an external acoustic model for decoding. Prior to constructing CNs from lattices [20], we decomposed the words into individual arcs according to the BPE scheme described in Section 3.2. The construction algorithm uses arcs from the first-best path as pivot elements to initialize arc clusters [21].

5. Neural Machine Translation System

We used the RETURNN toolkit [22] based on TensorFlow [23] for all NMT experiments. We trained two different architectures of NMT models: an attention-based RNN model similar to [1] with additive attention and a Transformer model [2] with multi-head attention.

In the RNN-based attention model, both the source and the target words are projected into a 620-dimensional embedding space. The models are equipped with either 4 or 6 layers of bidirectional encoder using LSTM cells with 1000 units. A unidirectional decoder with the same number of units was used in all cases. We applied a layer-wise pre-training scheme that lead to both better convergence and faster training speed during the initial pre-train epochs [22]. We also augmented our attention computations using fertility feedback similar to [24, 25].

In the Transformer model, both the self-attentive encoder and the decoder consist of 6 stacked layers. Every layer is composed of two sub-layers: a 8-head self-attention layer followed by a rectified linear unit (ReLU). We applied layer normalization [26] before each sub-layer, whereas dropout [27] and residual connection [28] were applied afterwards. Our model is very similar to "base" Transformer of the original paper [2], such that all projection layers and the multi-head attention layers consist of 512 nodes followed by a feedforward layer equipped with 2048 nodes.

We trained all models using the Adam optimizer [17] with a learning rate of 0.001 for the attention RNN-based model and 0.0003 for the Transformer model. We applied a learning rate scheduling similar to the Newbob scheme based on the perplexity on the validation set for a few consecutive evaluation checkpoints. We also employed label smoothing of 0.1 [29] for all trainings. The dropout rate ranged from 0.1 to 0.3.

6. Translation of ASR Confusion Networks

To encode confusion networks as input to the NMT system, we propose a novel, simple scheme. For a given speech utterance represented by acoustic vectors $\mathbf{o}$, we treat a confusion network C with J slots as the source sentence for the NMT. Instead of the one-hot encoding $x_j \in \{0,1\}^K$ (where K is the source vocabulary size) at position j within the sentence, the input is encoded as a K-dimensional vector $\bar{x}_j \in \mathbb{R}^K$ with $\bar{x}_j^k := p_j(w_k|\mathbf{o}), k = 1,\ldots,K$. Here, w_k is the k-th word in the vocabulary, and $p_j(w_k|\mathbf{o})$ is the posterior probability of the word w_k to appear at position j in C. In practice, $p_j(w_k)$ is different from 0 only for a small number of words.

In the end, following the notation of [1], we represent the input to the RNN encoder as the vector $E\bar{x}_j$ where $E \in \mathbb{R}^{N\times K}$ is the word embedding matrix and N is the dimension of the word embedding (e. g. 620). Thus, $E\bar{x}_j$ is a weighted combination of word embeddings for all the words in the CN slot j, with the highest weight given to the word with the highest posterior probability. In the corner case of only one arc per slot with the posterior probability of 1.0, we obtain a single word sequence. Thus, we can still use normal sentence pairs (e.g. from text-only parallel data) for training, along with the pairs of source CNs and their target language translations. The new input representation $E\bar{x}_j$ has the same dimensions as Ex_j and thus can be directly used to train a standard RNN NMT model or any other model that uses word embeddings. We kept the posterior weights fixed during back-propagation.

Word sequences of different length can be obtained from a CN because epsilon arcs can be inserted as alternatives in some of the slots. The best solution when training an NMT system on CNs would be to add an artificial source language token EPS that would not appear in the original text-only training data. However, because we decided against re-training the system on CN input from scratch, we mapped all epsilon arcs to the English word "eh", which denotes hesitation. It appears often enough in the English side of the parallel text-only corpus, but is almost always omitted in the human translation into German.

We also used CNs to simulate ASR word errors in text data. Following the work of [7], we used such noisy data in the training of the NMT system to make it more robust against similar real ASR word errors. To this end, for each word w in the first-best ASR output for the TED training corpus, we collected all the slot alternatives w'_n, $n = 1,\ldots,N_w$ to this word in the corresponding ASR CNs with their averaged posterior probabilities. After re-normalization of these probabilities, for each word w we obtained a confusion probability distribution p_w. Then, in a given sentence, we replaced every occurrence of the word w by one of its alternatives w'_n with probability $p_w(w'_n)$ from this distribution. One of the alternatives can also be an epsilon arc, we keep them (converted to "eh" as described above) to adapt the NMT system to epsilon arcs in CN input, inserting up to 2 consecutive arcs after each word with a probability e.

Finally, we used two control parameters to limit the noise level: probability to change a word p and probability to change anything at all in a given sentence s. Experimentally, we determined the settings $e = 0.02, p = 0.25, s = 0.6$ which resulted in WER of the noisy text as compared to its original text that was similar to the WER of the baseline ASR system.

Table 1: Results measured in BLEU [%] and TER [%] for the individual systems for the English→German speech translation task, translation of correct transcript vs. first-best ASR output of the TED.tst2015 set.

		correct transcript		**ASR output (WER of 10.9%)**	
#	**System**	BLEU	TER	BLEU	TER
0	RWTH IWSLT 2017 best non-ensemble system	30.5	52.3	–	–
1	text translation baseline (RNN)	32.4	50.5	25.2	60.2
2	text translation baseline (Transformer)	33.0	50.5	26.3	58.7
3	speech translation baseline (RNN)	31.4	51.9	26.6	60.0
4	speech translation baseline (Transformer)	30.7	52.8	25.8	59.5

7. Direct Speech Translation

In the direct approach to speech translation, a single neural network is used to predict the target translation given the audio features of the source sentence. The amount of training data for this setting, i.e. audio with the corresponding reference translations pairs from the TED corpus, is comparatively low. To exploit the much larger parallel text corpora, we choose a multi-task setup in which the network simultaneously learns to translate either from source audio or from source text. For this, we extend the RNN-based attention model described in Section 5 with an additional audio encoder that takes MFCC features as input. It consists of 5 bidirectional LSTM layers with 512 units each. Max-pooling layers with a pool size of 2 are inserted after each of the first 3 LSTM layers, reducing the sequence length by a factor of 8. Also, a separate attention mechanism is added for the audio encoder. The decoder switches between the context vector from the text encoder $c_{i,\text{text}}$ and the one from the audio encoder $c_{i,\text{audio}}$ depending on which input is given (using notation from [1]). The remaining part of the decoder is shared between both tasks.

To ensure that both types of input are seen frequently enough during training, we duplicate the speech translation corpus so that it grows to 30% the size of the parallel text corpus (66 duplicates). The concatenation of text and audio examples is then traversed in random order. For the direct system, the same optimization and regularization techniques are applied as in the NMT system described in Section 5.

8. Experimental Evaluation

We participated in the speech translation task of the IWSLT 2018 evaluation, the translation direction was English→German. All NMT models are trained on the filtered bilingual data as described in Section 3.1, no monolingual data was used. For the fine-tuning experiments, we used the TED talk part of the bilingual data together with the test sets TED.tst2010,2013,2014 (which were not used for tuning or evaluation). The TED talk part was also included in the baseline system. For the experiments with the confusion networks, we ran the ASR system to recognize the speech of the 170K TED training set and the test sets TED.tst2010,2013,2014 and used the resulting CNs with the corresponding German translations as (additional) training data.

We shuffled the training samples before each epoch and removed sentences longer than 75 and 100 sub-words in the attention RNN-based and the Transformer setup, respectively. We evaluate our models almost every 10K iterations and select the best checkpoint based on perplexity on the validation set. NMT decoding is performed using beam search with a beam size of 12 and the scores are normalized w.r.t the length of the hypotheses. We used TED.dev2010 consisting of 888 sentences as our validation set and evaluated our models on TED.tst2015 test set with 1080 segments. The systems were evaluated using case-sensitive BLEU [30] and normalized case-sensitive TER [31].

8.1. Baselines

First we trained a model with standard preprocessing for written text described in Section 3.2 and evaluated its quality on the correct transcript with punctuation marks of the TED.tst2015 set, as shown in Table 1. We observed a slightly better BLEU score for the Transformer architecture (line 2) as compared to the recurrent architecture (line 1). We also made a comparison to the best single system of RWTH Aachen University on this set from the IWSLT 2017 evaluation. With our baseline system we improved upon that result by 1.9% to 2.5% absolute.

We then trained a model with speech-like preprocessing of the English side of the parallel corpus as described in Section 3.2. This model not only translates English words to German, but also predicts punctuation marks. To match this condition, we applied the same preprocessing to the correct English transcript of TED.tst2015, removing the punctuation marks. The evaluation included punctuation marks. Because of the dual task (translation and punctuation prediction), the MT quality is lower, but only by 1% BLEU (line 3 of Table 1). Here, the recurrent architecture outperforms the transformer architecture (line 4) by a significant margin. Because of this, most of our subsequent experiments were based on the recurrent model.

8.2. Effects of ASR errors and Punctuation Prediction

When we translate the first-best ASR output for TED.tst2015, which has an ASR word error rate of

Table 2: Results measured in BLEU [%] and TER [%] for the individual systems for the English→German speech translation task, translation of ASR output.

#	System	TED.dev2010		TED.tst2015		tst2018	
		BLEU	TER	BLEU	TER	BLEU	TER
1	speech translation baseline (RNN)	26.5	55.2	26.6	60.0	–	–
2	+ fine-tuning on TED corpus	27.1	54.2	27.5	57.5	–	–
3	+ 2 additional encoder layers	27.3	54.7	27.6	57.5	–	–
4	+ fine-tuning on TED with noise	27.1	54.1	28.0	56.5	21.1	64.1
5	fine-tuning of 1) on TED CNs only	26.6	55.7	26.9	58.3	–	–
6	fine-tuning of 1) on TED correct + CNs	26.6	55.5	27.0	58.5	20.3	66.5
7	fine-tuning of 1) on TED correct+noise + CNs	26.2	55.9	27.0	57.9	20.2	66.7
8	speech translation baseline (Transformer)	26.1	55.6	25.8	59.5	–	–
9	+ fine-tuning on TED corpus	27.0	54.4	27.0	57.7	–	–
10	Ensemble of 2, 3, 4, 9	27.9	53.7	28.3	56.7	21.4	64.2
11	Ensemble of 2, 3, 4, 6	27.3	55.6	28.0	58.1	21.2	64.4
12	Ensemble of 2, 3, 4, 5, 6, 7	27.5	54.2	28.3	56.7	21.5	64.1

10.9%, we observe a significant degradation of MT quality. For example, the BLEU score goes down from 31.4% to 26.6%, cf. line 3 of Table 1. This means that the NMT system is sensitive to ASR errors. Otherwise, the differences between architectures are similar when compared on the ASR first-best output as opposed to correct transcript.

8.3. Confusion Network Translation

For the subsequent experiments, we start with the RNN speech translation baseline. Table 2 presents the results on the ASR output for the `TED.dev2010` validation set and `TED.tst2015` test set. For the lines where confusion networks are mentioned, they were used as input to the NMT system as described in Section 6. The CNs were pruned based on the threshold of 0.0001 for the posterior probability; a maximum of 20 arcs per slot with highest probability were kept. The average density of the final CNs on the training and validation sets was 1.8 and 2.2, respectively.

Fine-tuning on the TED corpus (using the correct transcript as the English side of the parallel corpus) improves the result on the test set by 0.9% BLEU absolute, as shown in line 2 of Table 2. We fine-tuned our models with a small learning rate of 0.00001 for the Transformer model and the models using CNs, which is additionally decayed by a factor ranging from 0.8 to 0.9 after each half an epoch. For the attention RNN-based models which do not use CNs as input, the learning rate was set to 0.0001 with decay rate of 0.9. We also tried fine-tuning using a model with 6 encoder layers instead of 4, but have not obtained any further improvements as compared to the fine-tuned model with 4 encoder layers.

Next, we duplicated the TED parallel corpus and introduced noise into the duplicate. The level of noise was selected to be similar to the ASR word error rate on the development set, and the noise itself was created as described in Section 6. Line 4 of Table 2 shows that after fine-tuning on both correct and noisy TED corpus, we obtain an improvement of 0.5% BLEU and 1.0% TER when translating the first-best ASR output for the `TED.tst2015` set, as compared to fine-tuning without the noise (line 2). Thus, to some extent the NMT system was able to learn how to cope with noise that is based on common ASR errors.

Because we could only run ASR on the 170K sentences from the TED corpus, for which the speech was well-aligned with reference translations, we decided to use confusion networks for fine-tuning of the NMT system only. To make it possible, we replaced the original English word embeddings of the model with the linear combination of the embeddings of CN slot alternatives, as described in Section 6. The fine-tuning was done on a random mix of correct TED talk transcripts and ASR CNs for these transcripts at the same time.

Lines 5-7 of Table 2 list the results of three fine-tuning experiments which include CNs in the source-side training data. We either continued training of the model on 170K CNs (and the reference translations of the corresponding transcripts), or on CNs plus correct transcripts of the same set, or on CNs plus correct transcripts and transcripts with noise that was inserted as in the experiment in line 4 of Table 2. In all three cases we used a lower learning rate, a smaller batch size, and continued fine-tuning for 5-6 epochs. Unfortunately, the BLEU/TER scores go down as compared to the best result in line 4 when translating first-best ASR output.

Detailed analysis of the NMT output from line 6 (best result on the validation set) showed that when translating confusion networks, the model is able to recover from some recognition errors. For example, the `TED.tst2015` utterance `you throw the ball but you’re hit right as you throw` is translated by the system in line 4 of Table 2 as `Sie werfen den Ball, aber das ist Ihr Thron` because of the ASR error `as you throw` → `is your throne`. The system that translates the corresponding ASR confusion network, however, is able to produce a correct translation: `Sie werfen den Ball, aber Sie werden getroffen`. In another anecdotal example there is no error in the first-best ASR output, but the NMT system is not able to disambiguate the meaning of the word picking. The English source is: see over there somebody is kind of picking their nose. The translation of the first-best ASR output is: Da drüben, da ist jemand in der Nase (Over there, someone is in the nose). When the corresponding CN is translated, the translation preserves the original meaning: Sehen Sie, da ist jemand, der sich in die Nase bohrt. We looked at the CN alternatives for the word picking. It had a posterior probability of 0.77, and the top competing hypotheses were taking (0.14), making (0.06), shaking (0.02), sticking (0.004). The embeddings of these words seem to have helped the NMT system to correctly infer the meaning of picking in the given context.

In many cases, however, multiple alternatives, especially to an empty slot, sometimes confused the system to a point where translation quality was adversely affected. In a final contrastive experiment, we used the system from line 7 in Table 2 to translate not the CNs, but the first-best ASR output for the same test set. The result – BLEU of 28.0% and TER of 56.5% on TED.tst2015 – was nearly identical to line 4 in the table and showed that the system was fine-tuned well to the TED domain, and the noise in the CNs made the NMT system more robust against the errors which the ASR system could not avoid to make. However, the system does not generalize well to unseen CNs and may make errors by encoding meaning from alternatives to correctly recognized words, even if their posterior probability is low. Nevertheless, we think that the approach is promising and can benefit from a better training strategy, more CNs in training, and a better, adaptive weighting of CN alternatives. We plan to implement these improvements in our future work.

8.4. Direct Speech Translation

The direct translation system trained on the 170K segments of the TED corpus where both the audio files and their translations are available and well aligned yields a BLEU score of 15.6% when translating the speech of the TED.tst2015 set. For comparison, a standard text-only attention RNN model trained on the same corpus using the correct English transcript reaches the BLEU/TER scores of 18.5% on the first-ASR output for the same set. Although the results are much worse than for the NMT systems in Table 2 trained on large amounts of data, we see that the direct system can still produce results only moderately worse than a system trained on the same data, but on English text instead of speech. When we try to improve the direct system by multi-task learning using all of the text parallel data as described in Section 7, we obtain a BLEU score of 17.1% after many days of training that has not converged, neither until the evaluation nor paper submission deadline. Thus, although in preliminary tests multi-task learning seems to work correctly and bring improvements, the approach requires a faster implementation and a better training/optimization strategy.

8.5. Final Results

The Transformer model, even when fine-tuned on the TED corpus, did not result in additional improvements over the attention based RNN model (lines 8 and 9 of Table 2). However, it contributed to the ensemble of several systems (line 10). The RNN model that uses CN input can potentially be ensembled with the Transformer NMT model (using either first-best or CN ASR output as input). However, we did not have time to implement the necessary changes for such model combination. Therefore, for our primary evaluation submission we combined only the RNN models which translate either first-best ASR input (models from lines 2, 3, 4 of Table 2) or confusion networks (models from lines 5-7 of Table 2). The BLEU of this ensemble system in line 12 is only marginally better than of the best single system from line 4.

For the 2018 evaluation data, we first used the acoustic sentence segmentation of the ASR system. Because it was too fine-grained and unreliable due to many pauses of the speakers and could lead to context loss for the NMT system, we ran the punctuation prediction algorithm described in Section 3.3 on the first-best ASR output, but used its results only to define new segment boundaries at time points when a period or question mark was predicted after a word. We then re-ran the recognition to generate first-best and CN ASR output using the segmentation obtained in this way. The performance on the 2018 evaluation data is reported in the last column of Table 2. The BLEU and TER scores were provided by the organizers for our primary and contrastive submissions. We observed similar tendencies here as on the TED.tst2015 set: unfortunately, using ASR confusion networks as input to NMT results in worse scores (e.g. by 0.8% absolute in BLEU) as compared to the best single system translating ASR first-best output. Our primary submission from line 12 obtains the best results also on the 2018 evaluation set, but the improvement due to ensembling of multiple systems is not significant.

9. Conclusion

AppTek participated in the speech translation task of the IWSLT 2018 evaluation, achieving the BLEU score of 21.5% on the 2018 English to German evaluation data with the primary submission. Our best setup used an ensemble of attention RNN MT models which translate either first-best ASR output or ASR confusion networks, generating target language text with punctuation marks. We proposed a novel scheme for encoding CNs in NMT and showed that the negative effect of some ASR errors can be reduced when CNs are translated, although further improvements in training strategy are necessary to obtain significant improvements in speech translation quality. Preliminary experiments with direct speech translation with a single sequence-to-sequence model showed promising improvements due to a novel multi-task learning scenario that allows for exploitation of text-only parallel MT training data.

10. References

[1] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," May 2015. [Online]. Available: http://arxiv.org/abs/1409.0473

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[3] H. Ney, "Speech translation: Coupling of recognition and translation," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Phoenix, Arizona, USA, Mar. 1999, pp. 517–520.

[4] E. Matusov, B. Hoffmeister, and H. Ney, "ASR word lattice translation with exhaustive reordering is possible," in *Interspeech*, Brisbane, Australia, Sept. 2008, pp. 2342–2345.

[5] N. Bertoldi, R. Zens, and M. Federico, "Speech translation by confusion network decoding," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Honululu, HI, USA, Apr. 2007, pp. 1297–1300.

[6] M. Sperber, G. Neubig, J. Niehues, and A. Waibel, "Neural lattice-to-sequence models for uncertain inputs," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 1380–1389.

[7] M. Sperber, J. Niehues, and A. Waibel, "Toward robust neural machine translation for noisy input sequences," in *International Workshop on Spoken Language Translation (IWSLT)*, 2017.

[8] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-sequence models can directly translate foreign speech," *arXiv preprint arXiv:1703.08581*, 2017.

[9] S. Bansal, H. Kamper, K. Livescu, A. Lopez, and S. Goldwater, "Low-resource speech-to-text translation," *arXiv preprint arXiv:1803.09164*, 2018.

[10] V. Vandeghinste, C.-K. Leuven, J. Pelemans, L. Verwimp, and P. Wambacq, "A comparison of different punctuation prediction approaches in a translation context," in *21st Annual Conference of the European Association for Machine Translation*, p. 269.

[11] P. Lison and J. Tiedemann, "Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles," 2016.

[12] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open source toolkit for statistical machine translation," in *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*, 2007. [Online]. Available: http://aclweb.org/anthology/P07-2045

[13] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016. [Online]. Available: http://aclweb.org/anthology/P/P16/P16-1162.pdf

[14] O. Tilk and T. Alumäe, "Bidirectional recurrent neural network with attention mechanism for punctuation restoration." in *Interspeech*, 2016, pp. 3047–3051.

[15] H. Bourlard and C. J. Wellekens, "Links between Markov models and multilayer perceptrons," in *Advances in Neural Information Processing Systems I*, D. Touretzky, Ed. San Mateo, CA, USA: Morgan Kaufmann, 1989, pp. 502–510.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," San Diego, CA, USA, May 2015.

[18] S. Wiesler, A. Richard, P. Golik, R. Schlüter, and H. Ney, "RASR/NN: The RWTH neural network toolkit for speech recognition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Florence, Italy, May 2014, pp. 3313–3317.

[19] P. Doetsch, A. Zeyer, P. Voigtlaender, I. Kulikov, R. Schlüter, and H. Ney, "RETURNN: the RWTH extensible training framework for universal recurrent neural networks," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, New Orleans, LA, USA, Mar. 2017, pp. 5345–5349.

[20] L. Mangu, E. Brill, and A. Stolcke, "Finding Consensus in Speech Recognition: Word Error Minimization and Other Applications of Confusion Networks," *Computer, Speech and Language*, vol. 14, no. 4, pp. 373–400, Oct. 2000.

[21] D. Hakkani-Tür and G. Riccardi, "A general algorithm for word graph matrix decomposition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Hong Kong, Apr. 2003, pp. 596–599.

[22] A. Zeyer, T. Alkhouli, and H. Ney, “RETURNN as a generic flexible neural toolkit with application to translation and speech recognition,” in *Proceedings of ACL 2018, Melbourne, Australia, July 15-20, 2018, System Demonstrations*, 2018, pp. 128–133. [Online]. Available: https://aclanthology.info/papers/P18-4022/p18-4022

[23] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[24] Z. Tu, Z. Lu, Y. Liu, X. Liu, and H. Li, “Coverage-based neural machine translation,” *CoRR*, vol. abs/1601.04811, 2016. [Online]. Available: http://arxiv.org/abs/1601.04811

[25] P. Bahar, J. Rosendahl, N. Rossenbach, and H. Ney, “The RWTH Aachen machine translation systems for IWSLT 2017,” in *14th International Workshop on Spoken Language Translation*, 2017, pp. 29–34.

[26] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *arXiv preprint arXiv:1607.06450*, 2016, version 1.

[27] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[28] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016, pp. 770–778. [Online]. Available: https://doi.org/10.1109/CVPR.2016.90

[29] G. Pereyra, G. Tucker, J. Chorowski, L. Kaiser, and G. E. Hinton, “Regularizing neural networks by penalizing confident output distributions,” *CoRR*, vol. abs/1701.06548, 2017. [Online]. Available: http://arxiv.org/abs/1701.06548

[30] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a Method for Automatic Evaluation of Machine Translation,” in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Philadelphia, Pennsylvania, USA, 2002, pp. 311–318.

[31] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, “A Study of Translation Edit Rate with Targeted Human Annotation,” in *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas*, Cambridge, Massachusetts, USA, 2006, pp. 223–231.