Human Language Technologies – The Baltic Perspective
A. Utka et al. (Eds.)

doi:10.3233/FAIA200605

Data Augmentation for Pipeline-Based Speech Translation

Diego ALVES [a], Askars SALIMBAJEVS [b,c]
and Mārcis PINNIS [b,c,1]

[a] *Faculty of Humanities and Social Sciences, University of Zagreb,*
Ul. Ivana Lučića 3, 10000, Zagreb, Croatia
[b] *Tilde, Vienības gatve 75A, Riga, Latvia, LV-1004*
[c] *University of Latvia, Raiņa bulv. 19-125, Riga, Latvia, LV-1586*

Abstract. Pipeline-based speech translation methods may suffer from errors found in speech recognition system output. Therefore, it is crucial that machine translation systems are trained to be robust against such noise. In this paper, we propose two methods for parallel data augmentation for pipeline-based speech translation system development. The first method utilises a speech processing workflow to introduce errors and the second method generates commonly found suffix errors using a rule-based method. We show that the methods in combination allow significantly improving speech translation quality by 1.87 BLEU points over a baseline system.

Keywords. Neural machine translation, speech translation, robustness

1. Introduction

Speech translation systems are either end-to-end [1,2,3] or pipeline-based where automatic speech recognition (ASR) and machine translation (MT) systems work sequentially [4,5,6]. End-to-end speech translation systems require parallel data that has a speech audio signal on the source side and translated transcriptions on the target side. Such parallel data may be hard to obtain even for languages where the necessary components for the pipeline-based systems are readily available. Thus, the pipeline approach is often more realistic and practical. However, since ASR systems can introduce errors that a standard MT system has not seen during training and thus cannot handle, the translation quality can suffer, even to an extent where the translations are incomprehensible [7]. Therefore, in this paper, we investigate how to train neural MT (NMT) systems that are suitable for work in tandem with ASR for pipeline-based speech translation. More specifically, we propose data augmentation methods to produce data that features mistakes common to speech recognition system output, thereby improving the NMT system robustness to noise propagated within the pipeline.

[1] Corresponding Author: Mārcis Pinnis; E-mail: marcis.pinnis@tilde.lv.

2. Related Work

The adverse effects of error propagation in pipeline-based speech translation systems have been studied [7] and addressed before [4,8,9,10]. Some of the work has proposed to mitigate ASR errors by translating either N-best lists [4] or lattices [9,10]. In this work, we focus on methods that do not require drastic changes in already existing MT workflows. Therefore, closest to our work is the work on ASR noise modelling [8,9]. Different from previous work, our noise generation method tries to introduce noise that is either actually generated by the ASR system or highly probable based on ASR error analysis. Our method is also suited for morphologically rich languages, for which a large proportion of errors are inflection mistakes.

3. Synthetic Data Generation

We investigate two methods for the generation of noise typical to ASR system output. The first method (see Section 3.1) generates noise using a speech processing workflow. Since the workflow may be limited in the lexical variety of errors it can introduce (due to a limited number of speakers), we performed an analysis of the types of errors introduced by the first method and devised also a rule-based method (see Section 3.2).

3.1. Synthetic Data Generation Using Speech Synthesis and Recognition

We propose to generate data with synthetic ASR noise by using a pipeline of text-to-speech (TTS) and ASR systems. Unlike the previous work [9], our approach generates not only substitution errors, but insertion and deletion errors as well. The main limitation of generating noise using a pipeline of TTS and ASR systems is the availability of such systems and the selection of different TTS voices. Other than that, no extra resources besides those used for MT system training are required.

We generate synthetic data as follows (see Figure 1):

- First, we synthesise source language sentences from the MT training data using TTS (the gray boxes in Figure 1).
- Then, we use ASR to acquire transcriptions of the synthesised sentences (the orange boxes in Figure 1).
- Finally, we use the ASR transcriptions together with the original target sentences as the synthetic MT training data.

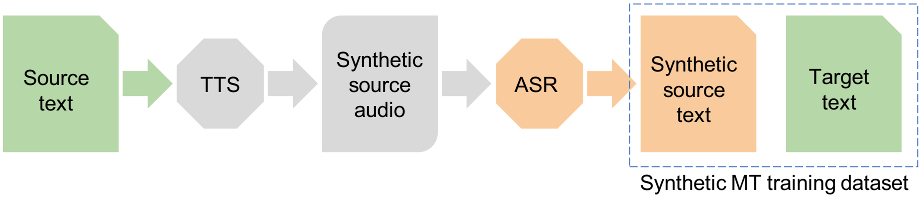

Figure 1. The synthetic data generation pipeline

3.1.1. Synthetic Data Filtering

Although the speech processing workflow introduces real noise, it is also evident that it introduces errors due to the limitations of the workflow itself. For instance, the speech synthesiser is unable to pronounce foreign named entities and complex identifiers correctly. This results in misrecognition by the ASR system (i.e., in such cases, it either deletes words or recognises the mispronounced names as some other common phrases, which are not even necessarily phonetically similar). The synthesiser also drops most Unicode characters that it cannot pronounce, which also introduces noise in the parallel corpus. Examples of errors introduced by the workflow itself are given in Table 1.

Table 1. Examples of noise introduced by limitations of speech processing tools

Source / Target in the parallel corpus	Synthetic segment (Latvian) / Translation (English)
Pierre Schapira, Attīstības komiteja	**ieviests papīra** attīstības komiteja
Pierre Schapira, Committee on Development	**introduced paper** committee on development
Mieczysław Edmund Janowski	edmunda jānoski
Mieczysław Edmund Janowski	edmund jnoski
Skatīt arī **MEMO /** 14 **/** 597.	skatīt arī **melo** 14 597
See also **MEMO /** 14 **/** 597.	see also **lie** 14 597

To address these issues, we filter the generated synthetic data by discarding sentences:

- that contain website addresses or Roman or Arabic digits,
- that contain one character,
- for which the Levenshtein [11] distance-based similarity between the original and the synthetic sentence is lower than 0.9.

We applied the Levenshtein distance-based similarity threshold to identify segments that have been too much corrupted by the synthetic data generation workflow. For instance, this allows addressing issues introduced by mis-recognising foreign named entities and complex identifiers.

3.2. Rule-Based Synthetic Noise Generation

After filtering, we performed error analysis to investigate what types of errors common to ASR systems were present in the filtered data. The results showed that 35.6 % of all errors were suffix-related. This was to be expected as Latvian is a morphologically rich language with fusional morphology and incorrect inflections are common mistakes for ASR systems. Other errors were deletions (32.9 %), insertions (25.2 %), and the remaining 6.3 % were related to other types of lexical misrecognition.

Next, we analysed what types of suffix (or inflection) errors are present in the data. For this, we used the Tilde's Latvian part-of-speech (POS) tagger[2] and extracted a list of most common inflection changes. The analysis revealed that the 26 most common inflection changes amount to 90 % of all suffix errors. We found that 26 most common inflection changes amounted to 90 % of all suffix errors. Therefore, we devised a method that in a random fashion iterates through the original dataset and randomly in each sentence

[2] The source code can be found online at: https://github.com/pdonald/latvian

generates one of the error types for a random number of words for which that particular error type can be introduced. As an additional step, we validated the generated errors using a vocabulary to make sure that the generation produced real Latvian words. After the generation, we selected only those sentences that had at least one error.

4. Experiments

We use the Latvian-English parallel data from the WMT 2017[3] shared task on news translation to train MT systems. The raw parallel corpus consists of 4,486,467 sentences. The corpus was pre-processed using a standard parallel data pre-processing workflow from the Moses statistical MT toolkit [12][4] and split into sub-word units. First, punctuation was normalised[5]. Then, the corpus was cleaned[6] by removing too long segments and segments where the source and target length ratio exceeds three times. After cleaning, 4,407,375 sentences remained in the training data. The data were further processed using the Moses tokeniser[7] and truecaser[8]. Finally, words were split into sub-word units using byte-pair encoding[9] [13,14] with 24.5K merge operations.

In order to generate the synthetic data, we use a selection of three Latvian TTS voices. For each sentence, we choose one of the voices at random. Characters that are not pronounced by TTS, are filtered from the source text (e.g., parentheses, quotation marks, various other punctuation marks).

For ASR, we use an ASR system, which is based on a hybrid Hidden Markov Model and a Time-delay Neural Network acoustic model and a sub-word level language model. The language model is implemented on a sub-word level (using byte-pair encoding, BPE) and consists of 4-grams, 6-grams, and a recurrent neural network (RNNLM). However, in order to inject more errors into the resulting synthetic training data, only the 4-gram language model is used to produce synthetic data. As the raw speech recognition output contains numbers written with words not digits, the MT model would have to learn how to translate words into digits. Therefore, to simplify the training of the MT system, we apply a number normalization tool for Latvian that re-writes numbers as digits. If number normalization tools are not available for a given language, raw transcripts from the ASR system could also be used as the source text for MT training.

The speech processing workflow produced 4,407,364 synthetic sentences. After filtering, 1,921,043 sentences remained in the synthetic dataset. The rule-based synthetic noise generation workflow produced 753,693 sentences and 961,227 sentences when using and not using vocabulary validation. The vocabulary was built using the original training data.

For validation, we use the *NewsDev2017* dataset from WMT 2017. The dataset was also processed using the speech processing workflow. For validation during training, we use a combination of the clean and noisy validation sets. For evaluation, we use a dataset, which represents real-world data from an ASR application. The evaluation set

[3] http://statmt.org/wmt17/

[4] https://github.com/moses-smt/mosesdecoder

[5] normalize-punctuation.perl

[6] clean-corpus-n.perl

[7] tokenizer.perl

[8] train-truecaser.perl and truecase.perl

[9] https://github.com/rsennrich/subword-nmt

is based on a subset of data collected by Tilde's real-time Latvian ASR service. This subset contains 8,820 utterances, which amount to 37,782 words and 39 hours of audio (including silence). Utterances originate from various domains: queries, short messages, addresses, interaction with a voice-enabled educational app, etc. It contains also a lot of "noise": laughter, English and Russian speech, untranslatable or ambiguous utterances, etc. Therefore, several rounds of semi-automatic filtering were performed to select meaningful utterances from this dataset. This resulted in a final evaluation dataset of 1,159 Latvian utterances that were manually translated to English.

All NMT systems were trained using Transformer models from the Marian NMT toolkit [15]. All models were trained till the the validation loss did not improve for 10 consecutive validation iterations.

5. Results

The results in Table 2 show that for the ASR output, supplementing the original training data with synthetic noise allows increasing the MT quality by up to 1.66 BLEU points. A similar tendency is evident when translating human-created transcripts that also may contain orthographic speech noise (e.g., truncated words, incorrect syntax, wrong punctuation, etc.). Only when translating clean publishable transcripts, the systems that are trained on noisy data show lower results than the baseline system. Nevertheless, the synthetic noise generation strategies have been successful in handling ASR output better and achieving higher translation quality than the baseline system.

The results also show that the best results were achieved when combining the filtered synthetic data and the data that is generated using rules with vocabulary validation. The combined data allow increasing the translation quality by 1.87 BLEU points over the baseline system.

Table 2. Evaluation results (bold – highest score; † – improvement over the baseline is significant with p< 0.01)

Training data	**ASR output**		**Human transcripts**		**Transcripts + punct.**	
	BLEU	**ChrF2**	**BLEU**	**ChrF2**	**BLEU**	**ChrF2**
a. Original parallel data (baseline)	12.73	0.4395	14.67	0.4622	**20.90**	**0.5123**
b. Noisy synthetic data	12.61	0.4160	14.08	0.4306	13.94	0.4251
c. a + b	†14.33	0.4374	†16.08	0.4577	20.45	0.5068
d. Filtered synthetic data + a	†14.39	**0.4602**	†16.79	0.4854	19.37	0.4995
e. Rule-based data (no voc.) + a	12.08	0.4243	13.12	0.4377	18.58	0.4830
f. Rule-based data (with voc.) + a	11.47	0.4231	12.75	0.4367	18.94	0.4847
g. Rule-based data (no voc.) + d	13.72	0.4484	†16.00	0.4815	18.69	0.4973
h. Rule-based data (with voc.) + d	†**14.60**	0.4547	†**17.29**	**0.4907**	19.46	0.5028

6. Conclusion

We proposed data augmentation strategies for speech translation that introduce noise typical to ASR output in parallel data for NMT systems. We showed that the methods allow generating synthetic parallel data that allows improving speech translation quality.

We believe the methods will be beneficial when developing NMT systems for speech translation purposes.

Future work on data augmentation strategies may be directed in two directions. Generation of a wider variety of ASR errors using rule-based methods (in these experiments, we covered only 26 rules) as well as investigating how much can be achieved by stripping most punctuation and symbols that are not supported by ASR systems from the parallel data.

Relevant code for re-producing the the synthetic data filtering and rule-based error generation results is published on GitHub[10].

Acknowledgements

This research has been supported by the ICT Competence Centre (www.itkc.lv) within the project "2.2. Adaptive Multimodal Neural Machine Translation" of EU Structural funds, ID n° 1.2.1.1/18/A/003. The work presented in this paper has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement no. 812997 and under the name CLEOPATRA (Cross-lingual Event-centric Open Analytics Research Academy). The presented research has been funded by the European Union's Horizon 2020 research and innovation programme project COMPRISE (http://www.compriseh2020.eu/) under grant agreement No. 3081705. The work has been carried out during Diego Alves' internship in Tilde.

References

[1] Duong L, Anastasopoulos A, Chiang D, Bird S, Cohn T. An attentional model for speech translation without transcription. In: Proceedings of NAACL 2016; 2016. p. 949–959.

[2] Bansal S, Kamper H, Lopez A, Goldwater S. Towards speech-to-text translation without speech recognition. In: Proceedings of EACL 2017; 2017. p. 474–479.

[3] Bansal S, Kamper H, Livescu K, Lopez A, Goldwater S. Low-Resource Speech-to-Text Translation. Proceedings of Interspeech 2018. 2018:1298–1302.

[4] Suhm B, Geutner P, Kemp T, Lavie A, Mayfield L, Mcnair AE, et al.. Janus: Towards Multilingual Spoken Language Translation; 1995.

[5] Waibel A, Fugen C. Spoken language translation. IEEE Signal Processing Magazine. 2008;25(3):70–79.

[6] Liu D, Liu J, Guo W, Xiong S, Ma Z, Song R, et al. The USTC-NEL speech translation system at IWSLT 2018. arXiv preprint arXiv:181202455. 2018.

[7] Ruiz N, Di Gangi MA, Bertoldi N, Federico M. Assessing the Tolerance of Neural Machine Translation Systems Against Speech Recognition Errors. 2017.

[8] Sperber M, Niehues J, Waibel A. Toward robust neural machine translation for noisy input sequences. In: Proceedings of IWSLT 2017; 2017. p. 1–7.

[9] Simonnet E, Ghannay S, Camelin N, Estève Y. Simulating ASR errors for training SLU systems. In: Proceedings of LREC 2018; 2018. p. 3157–3162.

[10] Sperber M, Neubig G, Pham NQ, Waibel A. Self-Attentional Models for Lattice Inputs. In: Proceedings of ACL 2019; 2019. p. 1185–1197.

[11] Levenshtein VI. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. Soviet Physics Doklady. 1966;10(8):707–710.

[12] Koehn P, Hoang H, Birch A, Callison-Burch C, Federico M, Bertoldi N, et al. Moses: Open Source Toolkit for Statistical Machine Translation. In: Proceedings of ACL 2007. ACL '07. Stroudsburg, PA, USA: Association for Computational Linguistics; 2007. p. 177–180.

[10] https://github.com/dfvalio/Speech_Translation

[13] Gage P. A new algorithm for data compression. C Users Journal. 1994;12(2):23–38.

[14] Sennrich R, Haddow B, Birch A. Neural Machine Translation of Rare Words with Subword Units. In: Proceedings of ACL 2016. Berlin, Germany: Association for Computational Linguistics; 2016. p. 1715–1725.

[15] Junczys-Dowmunt M, Grundkiewicz R, Dwojak T, Hoang H, Heafield K, Neckermann T, et al. Marian: Fast Neural Machine Translation in C++. In: arXiv preprint arXiv:1804.00344; 2018. p. 116–121.