INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Leveraging a character, word and prosody triplet for an ASR error robust and agglutination friendly punctuation approach

György Szaszák[1,2], *Máté Ákos Tündik*[2,3]

[1]Telepathy Labs GmbH, Zürich, Switzerland
[2]Budapest University of Technology and Economics, Budapest, Hungary
[3]Nokia Solutions and Networks Ltd, Budapest, Hungary

{szaszak,tundik}@tmit.bme.hu

Abstract

Punctuating ASR transcript has received increasing attention recently, and well-performing approaches were presented based on sequence-to-sequence modelling, exploiting textual (word and character) and/or acoustic-prosodic features. In this work we propose to consider character, word and prosody based features all at once to provide a robust and highly language independent platform for punctuation recovery, which can deal also well with highly agglutinating languages with less constrained word order. We demonstrate that using such a feature triplet improves ASR error robustness of punctuation in two quite differently organized languages, English and Hungarian. Moreover, in the highly agglutinating Hungarian, where word-based approaches suffer from the exploding vocabulary (poorer semantic representation through embeddings) and less constrained word order, we show that prosodic cues and the character-based model can powerfully counteract this loss of information. We also perform a deep analysis of punctuation w.r.t. both ASR errors and agglutination to explain the improvements we observed on a solid basis.

Index Terms: punctuation, prosody, character model, rich transcription, agglutination

1. Introduction

Punctuating the produced transcripts is an important add-on in Automatic Speech Recognition (ASR), which is not inherently handled by most state-of-the-art ASR systems. The problem is well-known, therefore we overview here only the aspects relevant for our contribution, and refer the reader to [1, 2, 3] for further exploration of the topic.

Early attempts proposed adding punctuation marks to the language models as hidden events [4, 5] and hence incorporated this knowledge into the ASR itself. Recent trends favour distinct punctuation modules, added to the ASR pipeline as a post-processing step (or on top of the model in an all-neural alike approach). Transducer- or tagging-based approaches were presented in numerous works [6, 7, 8] using hidden Markov-model, maximum entropy (MaxEnt) or conditional random fields etc. Viewing the punctuation task as a Sequence-to-Sequence (S2S) problem, Recurrent Neural Networks (RNN) can be efficiently exploited for punctuation. RNNs constitute today's state-of-the-art [9], eventually with attention mechanism [10, 11] or relying on the machine translation paradigm [12, 13] and adopting hence an encoder-decoder approach.

Both text and acoustics can be exploited for punctuation; text-based models [2, 6, 8, 10, 12] are more complex, but yield better results than models based on prosodic-acoustic features [1, 7, 14], which are more resistant to ASR errors. The two types of features are used in parallel in some studies [7, 13, 11] and most of the time, pause duration is used by text-based systems.

Text-based models are mostly built using word tokens, but a character-level CNN model was proposed in [15], which is reported to outperform a MaxEnt baseline, but not the word-based RNNs. In [16], character- and word-based models are used in combination and an improvement is demonstrated over systems based on standalone features. In [13], a phoneme-based approach is also considered, motivated by obtaining higher ASR error robustness, but did not improve punctuation performance despite that ASR word error rate (WER) was over 30% on the respective task.

Recently, we have addressed punctuation of Hungarian and English using text (both on word and character basis) [2, 16] and prosodic-acoustic approaches [1]. We observed, that punctuation models for English [2, 9, 10, 13] consistently work better for periods (full stops) than commas. In contrast, for Hungarian, text-based punctuation works better for commas, whereas prosody yields better period recovery [1]. We believe that this is closely linked to some basic characteristics of the two languages (other than comma usage standards, which are quite similar), i.e. word order is basically fixed in English, whereas less constrained in Hungarian. Another difference is agglutination leading to extreme large vocabulary. We intend to explore how these can impact punctuation by analysing combinations of acoustic and textual cues, including the character-level as well.Our goal is to propose an approach that fits well both types of languages – English and the less constrained word order an highly agglutinating Hungarian. We will show that character-level and prosodic features significantly improve performance on real ASR output in both languages. To the best of our knowledge, our work is the first one that considers the character, word and prosody triple feature set for punctuation. We also consider as a valuable contribution of our work that the proposed modelling framework is more universal and fits a broader set of languages, yet by significantly improving performance in both involved languages.

2. Modelling Principle

2.1. Word-level model

The word-level model (W) shown leftmost in Fig. 1 accepts a windowed non-overlapping word sequence (chunk) at its input and uses an embedding layer to map words to vectors. We use pre-trained embeddings (GloVe [17] for English and vectors from [18] for Hungarian); the word-level component module is adopted from [2], by using transfer learning. The vocabulary contains the k most common words in the training set (we

Copyright © 2019 ISCA

http://dx.doi.org/10.21437/Interspeech.2019-2132

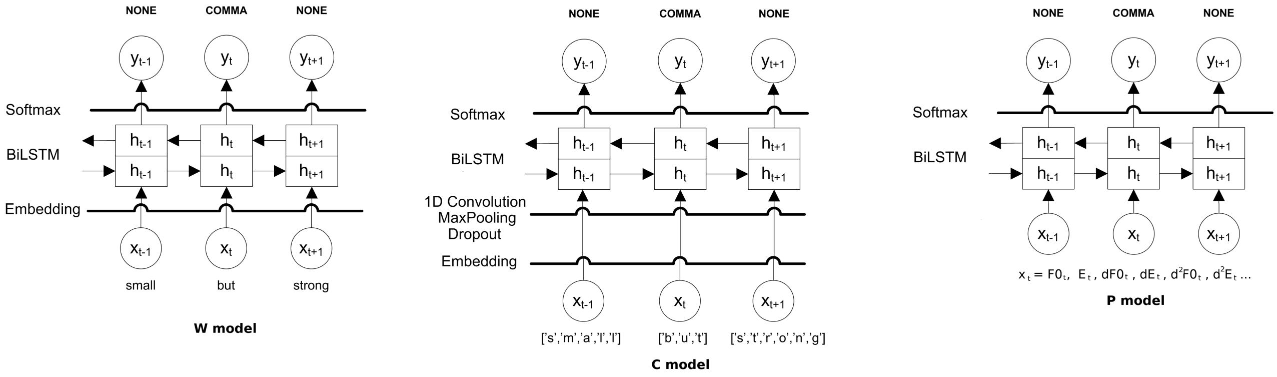

Figure 1: *The word (W), the character (C) and the prosody (P) models.*

also add an *unknown* entry to map rare words, and an *EOS* end-of-sequence symbol). The input is padded with zeros if it is shorter than chunk size. On top of the embeddings, a bidirectional LSTM layer is used, followed by a softmax layer to obtain the output (punctuation) with target classes of blank (0), period (.), comma (,), question (?) and exclamation mark (!). The model is described in more detail in [2], the hyperparameters of the model are summarized in Table 1.

2.2. Character-level model

The character-level model (C) represents the words as character sequences padded or cut for a fixed length. One-hot encoded characters get embedded via a dense layer. The C model is depicted in the centre of Fig. 1 and described in detail in [16]. The input has neither punctuation, nor capitalization. The structure is similar to the W model, but the embedding layer provides character-level embedding here, followed by a 1D convolutional layer of several filters and its outputs are downsampled using maxpooling. A bidirectional LSTM layer is added on top of this feature extractor, followed by a softmax to predict the punctuation mark out of $\{0 \ . \ , \ ? \ !\}$ for each character sequence in the input. Throughout the architecture, dropout layers are used to prevent overfitting. Please note that the character sequences for each word are first collapsed into fixed-length vectors and the LSTM already operates on word-by-word level. The reason for choosing this structure is to keep well aligned with the word-level and exploit character-level embedding instead of the word-level one, which allows for eliminating the Out-of-Vocabulary (OoV) problem in very large vocabulary Hungarian[1]. The hyperparameters of the model, optimized on the development set, are presented in Table 1.

2.3. Acoustic-prosodic model

In [1], the authors evaluated punctuation based on prosodic features derived from phonological phrase alignment. Since this approach involves a separate Hidden Markov-model to obtain the phonological phrase segmentation [19], we replace this with an end-to-end pipeline in this work, and let most of the modelling job be done by a neural network, but we keep the acoustic-prosodic feature pre-processing steps. We extract F0 and overall energy (for energy we use a 150 ms long window and do not decompose it into mel bins) by 10 ms frame rate and smooth the signals by applying a 5-points median filter. We compute first and second order derivatives, obtained by approximating the derivatives d_t of x_t in a $W = 30$ frame context by the following regression formula:

$$d_t = \frac{\sum_{i=1}^{W/2} i(x_{t+i} - x_{t-i})}{2\sum_{i=1}^{W/2} i^2} \qquad (1)$$

Wherever a word boundary is hypothesized by the ASR (or the alignment of the transcripts), two 15 frame long portions of this 6-dimensional feature sequence are extracted preceding and following the word boundary. Statistics composed of the minima, maxima and means of the 6 values are computed for the two portions separately and added to the input vector of the punctuation module. The input vector is augmented by the durations of the preceding word and of the pause at the word boundary. This input vector is fed into a bidirectional LSTM layer, followed by a softmax to predict the punctuation. This model is kept small (see Table 1 and Fig. 1) to allow for use with low amount of training data as we have limited audio transcribed with punctuation. This is also the reason for not using an embedding for the P model, although theoretically it would be possible in the acoustic space as well.

Compared to the method presented in [1], we require word boundary information, but since we intend to combine this prosodic punctuation module to the character- and word-based ones, this requirement does not add considerable complexity to our approach. Although we do not use a hierarchical encoder as in [13], or a phrase segmentation module as in [1] to map frame-based acoustic-prosodic features to the word-level, we still believe our approach is quite robust and yet simpler, as through the dynamic (delta and acceleration) prosodic features we capture most of the relevant prosodic features and their historic trajectories related to word boundaries (local stress patterns, intonation and pause duration).

2.4. Hybrid models

We will analyse the best combination of the individual character (C), word (W) and prosody (P) based models pairwise (C+W, C+P, W+P) and also into a triple (C+W+P) hybrid model by concatenating the respective LSTM hidden states of C and/or W models with the input of the P model. We drop the final softmax output layer of the single models, add a new second shared bidirectional LSTM layer and a new softmax layer, and train the network altogether.

[1] Please note that ASR dictionary may be larger than the vocabulary for the W-model, in order to keep word vectors good enough w.r.t. the available training data.

Table 1: *Hyperparameters of the individual models; where different from Hungarian, English is given in ()*

Input	Model	Chunk Size	Vocab. Size	Embedding dimension	#LSTM cells	Batch size	Optimizer	Filter length	#Filters	Stride	MaxPooling Window	Patience
Words	W	200	100,000 (27,244)	300 (100)	512	128	RMSProp	N/A	N/A	N/A	N/A	3 (2)
Chars	C	200	100	80 (70)	512	128	RMSProp	6 (5)	70	2 (1)	25 (20)	3 (2)
Prosody	P	200	N/A	N/A	512	16	RMSProp	N/A	N/A	N/A	N/A	3

3. Data

The **Hungarian dataset** is derived from public service and commercial TV broadcasts. This dataset contains various genres (weather forecasts, news, conversations, magazines, sport). The dataset used for pre-training the character- and word-level models is a subset with manual transcription including punctuation containing 12M, 3M and 136k words for the train, validation and test sets, respectively [20]. The punctuation marks addressed in the experiments include commas, periods, question marks and exclamation marks. Audio is associated with a smaller part of the data covering mainly broadcast news genres [21] of 3.5 hours. This data is augmented by samples from Hungarian BABEL [22] to provide more question and exclamation samples associated with audio. The acoustic-based model is trained from scratch on this part, whereas we applied transfer learning method on character- and word-level models.

The **English dataset** consists of IWSLT TED Talk transcripts, which is a commonly used benchmark dataset for English punctuation models [10, 12, 23]. We use the predefined train, validation and test sets, containing 2.1M, 296K and 13K words respectively, and dealing with three types of punctuations (comma, period and question mark). This corpus contains no audio, hence audio is derived from the IWSLT2011 talk translation dataset [24] containing 6 hours of speech.

4. Results and Discussion

We use the F1-score computed from precision and recall for evaluation, corresponding to the common practice for the punctuation task. We decided not to provide Slot Error Rates (SER) in detail, just for the best models, as differences seen in SER were found consistent with differences in F1 among the tested models.

Results for Hungarian are shown in Fig. 2 (overall WER of the ASR is 34.6%). Text-based (C/W) models yield good comma recovery, but they perform weaker for periods, both on reference (REF) and ASR transcripts. On the other hand, as sentence endings tend to be more marked by prosody, the P model performs well with period prediction, but has weak comma prediction capabilities. Combining them hence leads to overall improvement (with C+P, W+P and C+W+P). In Fig. 3 we present Venn-diagrams showing the share (in %) of the individual models on correctly recovered punctuation marks. When working on ASR output, as expected, the amount of slots missed by W increases considerably, letting the P model represent alone almost 1/3 of the punctuations! The C model roughly preserves its share when switching to ASR transcripts. This shows that, as can be expected, the C and especially the P models are more ASR error robust. Overall the best models were C+W+P on reference ($F1 = 65.2\%; SER = 53.8\%$) and C+P on ASR ($F1 = 43.9\%; SER = 88.6\%$).

Question and exclamation marks are better predicted based on C/W, than on prosody (P). Analysing confusion matrices and listening to the particular utterances showed that intonation often does not correspond to the interrogative or exclamative modalities, but is realized with declarative patterns. Question specific words on the other hand can help C/W models to better predict these. This has been also observed in [1] and [13] – despite prosody should theoretically be an ideal candidate to detect questions, practice does not confirm this.

Results for English are presented in Fig. 4 (by WER: 18.7% for ASR). The C+W+P model yields the best overall performance on reference ($F1 = 53.6\%; SER = 65.5\%$), and the W+P model performs the best on English ASR ($F1 = 52.8\%; SER = 70.2\%$). Taking a look at the Venn-diagrams representing the share of each model in correct punctuation predictions in Fig. 5, REF and ASR cases are more balanced than in Hungarian, there is no significant gain from adding prosody. This can be explained partly by the lower WER for English, but differences between the two languages likely interplay, too. Indeed, WER is often higher in agglutinating languages compared to English in corresponding, similar tasks [25]. In English, periods are efficiently recovered by C/W models, and performance

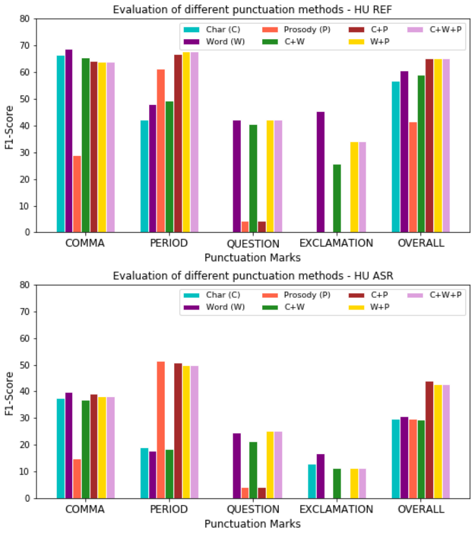

Figure 2: *Punctuation on REF / ASR transcripts in Hungarian*

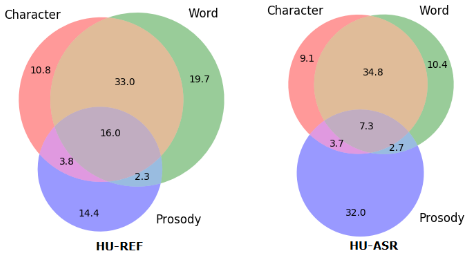

Figure 3: *Contribution of the individual models to correctly predicted punctuations (Hungarian)*

for commas is weaker than for periods for all models. This is opposite to Hungarian. We explain differences between English and Hungarian by their fixed vs relatively free word order and also by the highly agglutinating nature of Hungarian resulting in extreme large vocabularies[2]. Comma usage rules are quite similar for these two languages.

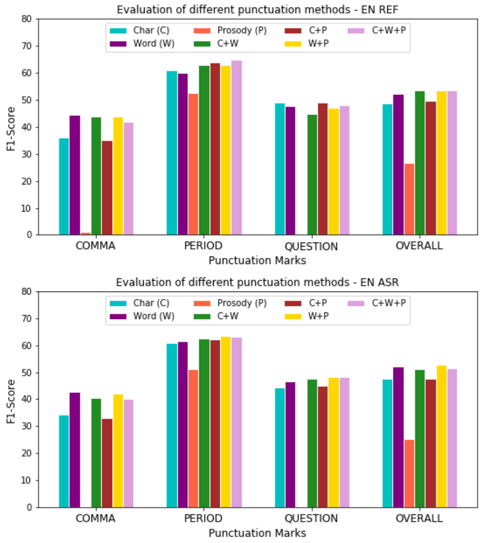

Figure 4: *Punctuation on REF and ASR transcripts in English*

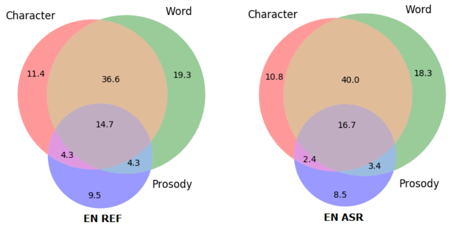

Figure 5: *Contribution of the individual models to correctly predicted punctuations (English)*

Considering the W model, word embeddings operate by capturing the semantic (and also pragmatic) relations of individual words. In a language with constrained word order such as English, the set of words and the role of the words occurring at sentence boundaries is less variable than in Hungarian. Another aspect of the phenomenon results from the fact that word embeddings are less powerful for a language of intense agglutination. The higher number of possible word forms means that the embeddings have more OOV words, and their estimation is less robust resulting from the higher overall variability of the data. The latter constitutes the main problem. Using word embeddings enhanced by character N-grams and matching embeddings to ASR vocabulary can help the OOV problem, and it also improves semantic accuracy as shown for other languages in [26]. But the unconstrained word order problem still remains an issue, and we observed that the Hungarian word embeddings enhanced by character N-grams still showed drastically less absolute semantic accuracy than the English ones [26] despite training them on large corpora. (By using a larger skip-gram context, again the estimate could be less accurate, but we have not tested this so far.) On the other hand, through the C model we also can exploit character-level information with a framework optimized for the task, hence we did not use embeddings enhanced by character N-grams in this work, and had an OOV rate of 7% for Hungarian and 4% for English.

Seeing higher prediction power for commas and lower for periods with the W model in Hungarian is fairly in line with the above hypothesis: if we consider the most common cases where a comma is used – to separate clauses where most often a conjunction word known by the embedding occurs; or in enumerations where typically words with high similarity in the semantic space are involved – embeddings can perform well for these situations. Sentence endings however show high variability resulting from the less constrained word order.

Summarizing the results, for the highly agglutinating and relatively free word order Hungarian, we can obtain significant improvement in overall punctuation over the W baseline by adding the C and P models. Considering the most important use-case, where ASR transcripts are input, C+P helps by rel. 18% in Hungarian over the W baseline (significant by $p < 0.01$). For English, adding prosody lead to an improvement of 4.4% over the W baseline on ASR transcripts (still significant by $p < 0.05$).

5. Conclusions

We proposed a punctuation approach combining character, word and prosody features, and illustrated that C and P models can contribute to by a large margin better punctuation performance in agglutinating Hungarian, where WER is often higher and W punctuation models rely on less effective word embeddings due to the less constrained word order. Although to a smaller extent, but still significant improvements were seen in English as well. Our C+W+P ensemble outperformed all individual systems on reference in Hungarian and English, while on ASR transcripts, the C+P hybrid gave the overall best results in Hungarian and the W+P one in English. We also demonstrated and quantified how the prosody based model provides robustness when also ASR errors degrade the word-based model's performance. In agglutinating Hungarian, P is powerful in recovering periods, while commas are better predicted by C/W features. We explained this difference compared to English by word embeddings' semantic modelling capabilities, weaker for agglutinating languages (less constrained word order, large vocabulary). These results should be applicable to all similar (agglutinating) languages and highlight the importance of C/P features for such languages. We argue that the proposed framework can be applicable for a wide range of languages, as it integrates all features relevant for punctuation prediction. The contribution of the individual features is per se language dependent, which can however give an interesting insight into the inner structure and nature of agglutinating languages, i.e. to illustrate how the increased contribution of prosody can counteract the higher variability in word forms and word order to maintain understandability at a high level in speech communication.

6. Acknowledgements

Training was performed on GPUs kindly provided by NVIDIA. Research was supported by the Hungarian Research, Development and Innovation Office (NKFIH, contract ID *FK-124413*).

[2] Agglutinating is linked to free word order too: since case endings make clear grammatical relations in a sentence, the words can be moved around more freely by preserving almost the same core meaning.

7. References

[1] A. Moró and G. Szaszák, "A phonological phrase sequence modelling approach for resource efficient and robust real-time punctuation recovery," in *Proceedings of Interspeech*, 2017.

[2] M. Á. Tündik, B. Tarján, and G. Szaszák, "Low Latency MaxEnt- and RNN-Based Word Sequence Models for Punctuation Restoration of Closed Caption Data," in *International Conference on Statistical Language and Speech Processing*. Springer, 2017, pp. 155–166.

[3] O. Klejch, P. Bell, and S. Renals, "Punctuated transcription of multi-genre broadcasts using acoustic and lexical approaches." in *SLT*, 2016, pp. 433–440.

[4] E. Shriberg, A. Stolcke, and D. Baron, "Can Prosody Aid the Automatic Processing of Multi-Party Meetings? Evidence from Predicting Punctuation, Disfluencies, and Overlapping Speech," in *ISCA Tutorial and Research Workshop (ITRW) on Prosody in Speech Recognition and Understanding*, 2001.

[5] C. J. Chen, "Speech recognition with automatic punctuation," in *Proceedings of Eurospeech*, 1999, pp. 447–450.

[6] D. Beeferman, A. Berger, and J. Lafferty, "Cyberpunc: A lightweight punctuation annotation system for speech," in *Proceedings of ICASSP*. IEEE, 1998, pp. 689–692.

[7] F. Batista, H. Moniz, I. Trancoso, and N. Mamede, "Bilingual experiments on automatic recovery of capitalization and punctuation of automatic speech transcripts," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 474–485, 2012.

[8] W. Lu and H. T. Ng, "Better punctuation prediction with dynamic conditional random fields," in *Proceedings of EMNLP*. ACL, 2010, pp. 177–186.

[9] O. Tilk and T. Alumäe, "LSTM for punctuation restoration in speech transcripts," in *Proceedings of Interspeech*, 2015, pp. 683–687.

[10] O. Tilk and T. Alumäe, "Bidirectional Recurrent Neural Network with Attention Mechanism for Punctuation Restoration," in *Proceedings of Interspeech*, 2016, pp. 3047–3051.

[11] A. Öktem, M. Farrús, and L. Wanner, "Attentional parallel RNNs for generating punctuation in transcribed speech," in *International Conference on Statistical Language and Speech Processing*. Springer, 2017, pp. 131–142.

[12] E. Cho, J. Niehues, K. Kilgour, and A. Waibel, "Punctuation insertion for real-time spoken language translation," in *Proceedings of the Eleventh International Workshop on Spoken Language Translation*, 2015.

[13] O. Klejch, P. Bell, and S. Renals, "Sequence-to-sequence models for punctuated transcription combining lexical and acoustic features," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5700–5704.

[14] H. Christensen, Y. Gotoh, and S. Renals, "Punctuation annotation using statistical prosody models," in *ISCA Tutorial and Research Workshop (ITRW) on Prosody in Speech Recognition and Understanding*, 2001.

[15] W. Gale and S. Parthasarathy, "Experiments in character-level neural network models for punctuation," in *Proceedings Interspeech*, 2017, pp. 2794–2798.

[16] M. A. Tündik and G. Szaszák, "Joint Word-and Character-level Embedding CNN-RNN Models for Punctuation Restoration," in *Cognitive Infocommunications (CogInfoCom), 2018 9th IEEE International Conference on*. IEEE, 2018.

[17] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation." in *Proceedings of EMNLP*, 2014, pp. 1532–1543.

[18] M. Makrai, "Filtering Wiktionary triangles by linear mapping between distributed models," in *Proceedings of LREC*, 2016, pp. 2776–2770.

[19] G. Szaszák and A. Beke, "Exploiting prosody for automatic syntactic phrase boundary detection in speech," *Journal of Language Modeling*, vol. 0, no. 1, pp. 143–172, 2012.

[20] Á. Varga, B. Tarján, Z. Tobler, G. Szaszák, T. Fegyó, C. Bordás, and P. Mihajlik, "Automatic Close Captioning for Live Hungarian Television Broadcast Speech: A Fast and Resource-Efficient Approach," in *Proceedings of SPECOM*. Springer, 2015, pp. 105–112.

[21] C. Teleki, S. Velkei, S. L. Tóth, and K. Vicsi, "Development and evaluation of a Hungarian Broadcast News Database," in *Forum Acusticum*, 2005.

[22] P. S. Roach, S. Amfield, W. Bany, J. Baltova, M. Boldea, A. Fourcin, W. Goner, R. Gubrynowicz, E. Hallum, L. Lamep, K. Marasek, A. Marchal, E. Meiste, and K. Vicsi, "Babel: An eastern european multi-language database," in *International Conf. on Speech and Language*, 1996, pp. 1033–1036.

[23] V. Pahuja, A. Laha, S. Mirkin, V. Raykar, L. Kotlerman, and G. Lev, "Joint Learning of Correlated Sequence Labelling Tasks Using Bidirectional Recurrent Neural Networks," *arXiv preprint arXiv:1703.04650*, 2017.

[24] M. Federico, S. Stüker, L. Bentivogli, M. Paul, M. Cettolo, T. Herrmann, J. Niehues, and G. Moretti, "The IWSLT 2011 evaluation campaign on automatic talk translation," in *International Conference on Language Resources and Evaluation (LREC)*, 2012, pp. 3543–3550.

[25] M. Kurimo, A. Puurula, E. Arisoy, V. Siivola, T. Hirsimäki, J. Pylkkönen, T. Alumäe, and M. Saraclar, "Unlimited vocabulary speech recognition for agglutinative languages," in *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics -*. Morristown, NJ, USA: Association for Computational Linguistics, 2006, pp. 487–494.

[26] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.