INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Sentence Boundary Detection Based on Parallel Lexical and Acoustic Models

Xiaoyin Che, Sheng Luo, Haojin Yang, Christoph Meinel

Hasso Plattner Institute, University of Potsdam
Prof.-Dr.-Helmert-Str. 2-3, 14482, Potsdam, Germany
{xiaoyin.che, sheng.luo, haojin.yang, christoph.meinel}@hpi.de

Abstract

In this paper we propose a solution that detects sentence boundary from speech transcript. First we train a pure lexical model with deep neural network, which takes word vectors as the only input feature. Then a simple acoustic model is also prepared. Because the models work independently, they can be trained with different data. In next step, the posterior probabilities of both lexical and acoustic models will be involved in a heuristic 2-stage joint decision scheme to classify the sentence boundary positions. This approach ensures that the models can be updated or switched freely in actual use. Evaluation on TED Talks shows that the proposed lexical model can achieve good results: 75.5% accuracy on error-involved ASR transcripts and 82.4% on error-free manual references. The joint decision scheme can further improve the accuracy by 3∼10% when acoustic data is available.

Index Terms: Sentence Boundary Detection, Parallel Models, Deep Neural Network, Word Vector

1. Introduction

Sentence boundary detection, or addressed as punctuation restoration, is an important task in ASR (*Automated Speech Recognition*) post-processing. With proper segmenting, the readability of the speech transcript is largely improved and some downstream NLP (*Natural Languages Processing*) tasks, such as machine translation, can also benefits from it [1, 2, 3]. In actual use, subtitling for example, the quality of the automatically generated subtitles might increase when the sentence boundaries detected are more accurate [4]. Perhaps it is still not good enough in user's point of view, but at least a semi-finished subtitle with better quality can definitely help the human subtitle producer.

Many efforts have already been made in sentence boundary detection. Generally the researchers focus on two types of resources: lexical features in the textual data and acoustic features in audio track. Most of the lexical approaches take LM scores (*Language Model*), tokens or POS tags (*Part-of-Speech*) of several continuous words as the features to train the lexical model [5, 6, 7, 8]. And the frequently used features in acoustic approaches include pause, pitch, energy, speaker switch and so on [9, 10, 11]. However, multi-modal approaches using both lexical and acoustic features are more popular.

In this paper, we first analyze the structure of those existing multi-modal solutions and propose our own sentence boundary detection framework. Then we introduce the independent lexical and acoustic models used in our framework and explain the 2-stage joint decision scheme in detail. These contents can be found in Section 2∼4 respectively. In the following evaluation phase, we evaluate the performance of proposed lexical models with both ASR and manual transcripts, test the acoustic models on ASR-available TED Talks and run experiments about our 2-stage joint decision scheme with different combination of models. In the end comes the conclusion.

2. Multi-Modal Structure Analysis

The structures of multi-modal solutions can be different and have been discussed before [12]. Some researchers propose a single hybrid model, which takes all possible features, no matter lexical or acoustic, together as the model input [13, 14, 15, 16], as shown in Figure 1-a. Some others apply a structure of sequential models, in which the output of model A is fed into model B, as shown in Figure 1-b, where model A accepts either lexical or acoustic features only, while model B takes the other type of features together with model A's output [17, 18, 19, 20, 21].

But all approaches with these two structures have a limitation: the training data must be word-level synchronized transcripts and audios, which largely limits the range of data collection. Many reports claimed that the classification performance can be largely influenced by the scale of training data. Furthermore, if ASR transcript is used as training data, the inevitable ASR errors, some of which are acoustically understandable but lexically ridiculous, such as misrecognizing "*how can we **find information** in the web*" as "*how can we **fight formation** in the web*", will definitely downgrade the functionality of the lexical model trained.

However, a structure of parallel models, as shown in Figure 1-c, can overcome this limitation. Models can be trained separately with different data. It means the lexical model can take any kind of textual materials as training data, which is almost endless, extremely easy to prepare and could be grammatically error-free. For the acoustic model, all available training data of previous two structures are still available. The next step is to fuse their posterior probabilities.

Gotoh *et al.* and Liu *et al.* trained models with different feature sources separately and interpolated their posterior probabilities for the final prediction [22, 23]. Lee and Glass applied log-linear model to combine outputs from different models [24], so did Cho *et al.* [25]. Pappu *et al.*, on the other hand, used logistic regression model for the fusion [26]. All these approaches offer predictions in one step and need an activation dataset to adjust fusion model parameters.

However, we would like to apply a 2-stage scheme. Different from some earlier multi-pass attempts [10, 27], which first predict punctuation positions and then distinguish punctuation mark types, the two stages in our decision scheme is like "segmenting" and "sub-segmenting". Therefore the lexical or acoustic model can be updated or switched freely in actual use. Details will be introduced in Section 4.

http://dx.doi.org/10.21437/Interspeech.2016-257

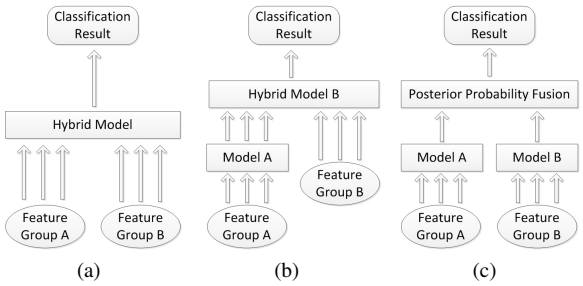

Figure 1: Three types of multi-modal structures

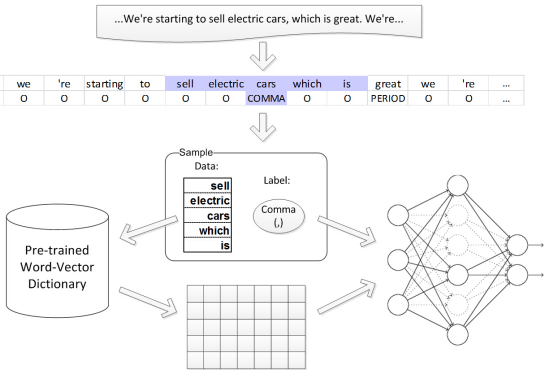

Figure 2: The process of data generation for lexical training

3. Proposed Models

3.1. Lexical Model

Word vectors are learned through neural language models [28]. Then a word can be represented by a real-valued vector, which is much lower dimensional when comparing with the traditional one-hot representation of words. It is suggested that the semantic distance between words can be measured by the mathematical distance of corresponding word vectors [29, 30]. Cho *et al.* has included word vector in punctuation prediction task together with many other lexical features [25]. However, the solo usage of word vectors has already been proven effective in various NLP applications [31, 32, 33] and will make the data preparation much easier. Therefore, word vector is used as the only feature for our lexical model.

The training data are extracted from punctuated textual files, which will be first transformed into a long word sequence with a parallel sequence of punctuation marks. Then an m-words sliding window will traverse the word sequence to create samples. The classification question is that whether there is a sentence boundary after the k-th word of a sample. Originally we introduced three categories to represent boundaries: Comma, Period and Question. All other punctuation marks will be switched into one of them or just ignored based on their functionalities. However, these categories can be combined freely when needed. "O" is used for "not a boundary" samples.

Then each word in the sample will be represented by an n-dimensional word vector which is stored in a pre-trained dictionary. A default vector will be taken as the substitute of any words out of the vocabulary. As the result, we obtain an $m \times n$ feature matrix as the lexical model input for the sample. During the training process, the value of all word vectors is kept static. The whole process is illustrated in Figure 2.

We choose a typical deep neural network for lexical training. Its structure is comparatively simple, with three sequential fully-connected hidden layers of 2048, 4096 and 2048 neurons respectively. Therefore, the computational cost for the training is not very high. In order to avoid co-adaptation, "dropout" is implemented on these layers, which randomly hides some neurons along with their connections during the training process [34]. Softmax function is applied for the output layer.

In this approach we propose the lexical model with two configurations, addressed as LMC-1 and LMC-2 *(**L**exical **M**odel **C**onfiguration)*. Both of them use publicly available word vectors: LMC-1 uses GloVe.6B.50d vector set[1], as explained in [35], while LMC-2 applies Word2Vec-Google-300d[2]. Generally, LMC-1 is a light configuration while LMC-2 involves more data. The detailed settings can be found in Table 1. We use the word "this" as the default substitute for out-of-vocabulary words, since most of them are proper nouns which exist in special context only, such as "Word2Vec" we mentioned above, and "*I use **this** in...*" is grammatically similar to "*I use **Word2Vec** in...*" in purpose of sentence boundary detection. The neural network is constructed by CAFFE framework [36].

Table 1: Two configurations of our lexical training. ("Voc.", "n" , "m" and "k" represent vocabulary, vector dimension, sliding window size and supposed boundary position respectively)

Config.	Word Vector			Sample Size	
	Source	Voc.	n	m	k
LMC-1	GloVe	400k	50	5	3
LMC-2	Word2Vec	3M	300	8	4

3.2. Acoustic Model

In this approach we also apply two models to handle acoustic information. Different from the 4-classes lexical model, acoustic model outputs only 2 classes: boundary or not boundary. Our first acoustic model is a simplest one. For a word W_i in the ASR transcript, we simply calculate the pause duration p between W_i and W_{i+1} and use a variant of Sigmoid function:

$$P_a = \frac{1 - e^{-4p}}{1 + e^{-4p}}, \quad p \in [0, +\infty) \tag{1}$$

to project p into P_a, while $P_a \in [0, 1)$. Approximately when the pause is longer than 0.28 second, it will be acknowledged as a sentence boundary by this simplest model, which we would like to address as "Pause".

The second acoustic model takes more features into consideration. Pitch level and energy level are extracted from audio files by aubio[3] and Yaafe[4] toolkits. Then based on the time tags in the ASR transcripts, an average pitch level or energy level can be achieved for each recognized word. Similar to LMC-2 in lexical model, we also apply an 8-words context to form a sample, which classifies whether there is a sentence boundary after the 4th word.

Therefore, each sample for the second acoustic model contains 25 features: pitch and energy value for each word and 9 pauses available in the 8-words context. The features will also be fed into a neural network with 3 fully-connected layers for

[1] http://nlp.stanford.edu/projects/glove/
[2] https://code.google.com/p/word2vec/
[3] http://aubio.org/
[4] http://yaafe.sourceforge.net/

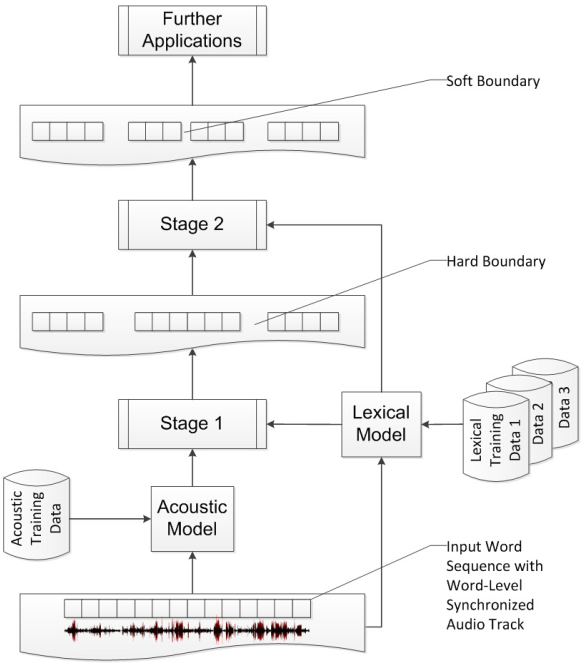

Figure 3: The workflow of proposed Joint Decision Scheme

training. Based on the features involved, we address this second acoustic model as "PPE" (***P**ause, **P**itch & **E**nergy*).

4. 2-Stage Joint Decision Scheme

As already mentioned in Section 2, the relation between the two stages of proposed joint decision scheme is like "segmenting" and "sub-segmenting". Stage-1 takes the posterior probabilities of both lexical and acoustic models as input and detects the sentence boundary position (*Hard Boundary*). Then the complete word sequence can be split by these positions into segments. Each segment will be further checked by the adjusted lexical model output for potential sub-segmenting (*Soft Boundary*) in stage-2. Figure 3 illustrates this procedure.

In stage-1, acoustic model output is the foundation. Ideally the sentence boundaries in the speech should always result in something detectable in acoustic features, especially the pauses. But actually the speaker may hesitate or be interrupted by unexpected events. These phenomena result in false positive sentence boundaries from acoustic analysis. Therefore, lexical probability is employed here to "filter" those false positive boundaries. The basic idea is that if an acoustically supposed boundary position is strongly opposed by the lexical model, it will be denied. And the lexical denial threshold is associated with the confidence of acoustic prediction by a simple linear function. Here we use P_a and P_l to represent the posterior probability of "being a boundary" from acoustic and lexical model respectively, then a hard boundary will be confirmed if $1 - P_l < P_a \times 0.25 + 0.7$ and $P_a > 0.05$.

In stage-2, only lexical model output is used. The goal here is to recover the sentence boundaries which have no acoustic hint. Since many boundaries have already been detected in stage-1, we set very strict restriction on stage-2 classification. Therefore, the posterior probability is adjusted by

$$P_l' = P_l \times e^{(\frac{L}{\hat{L}} - \lambda)} \times \frac{d \times (L - d)}{(\frac{L}{2})^2} \quad (2)$$

where L is the length of the input segment, d is the distance between current word and previous detected boundary, $\hat{L}$ is the expected length between adjacent boundaries and λ is the restriction coefficient. This adjustment generally reduces the P_l. In practice the value of $\hat{L}$ and λ are fixed and the extent of reducing becomes smaller when L gets larger and d approaches $L/2$, which means a soft boundary is supposed to be found in the middle position of a long input segment. In extreme case, the adjustment might even increase P_l, but L needs to be more than λ times larger than $\hat{L}$, which happens rarely. Generally, only positions with very strong lexical evidence to be boundaries can be acknowledged after the adjustment as (2).

Basically the joint decision scheme works with only 2 classes: boundary or not boundary. But if punctuation marks need to be restored, it can be fulfilled in stage-2 based on the lexical model in use. Suppose n types of punctuation marks are available in the lexical model trained, then $P_l = \sum_{i=1}^{n} P_i$. As long as a position has already been confirmed as a boundary, no matter hard boundary or soft boundary, the i-th type of punctuation mark will be chosen when P_i is the largest in $\{P_1, P_2, ..., P_n\}$.

5. Evaluation

5.1. Data Collection

We collected all the data from IWSLT datasets, which can be found online. For the lexical model, we aim to evaluate its performance with both ASR transcripts and manual references. The test set is the "tst2011" package for IWSLT 2012 ASR Track, which consists of 8 TED Talks and has both ASR and manual transcripts, containing around 12k words each. The training dataset consists of the manual transcripts of 1710 TED Talks and comes originally from the in-domain training data of IWSLT 2012 MT Track. We further split it into training set and development set, with 2.1M and 296k words respectively. Based on the TalkID, we make sure there is no overlapping between training and test sets. The average length between two adjacent boundaries in the 2.1M samples of the training set is 7.8, which would be taken as the $\hat{L}$ in the joint decision scheme, while λ was set to 3.

For the acoustic model, the range of data selecting is quite limited. We also used the "tst2011" ASR transcripts as the test set, with 8 TED Talks in total. And we managed to find 70 other TED Talks with ASR transcripts and audio files from different IWSLT datasets as the training data. There is no development set for acoustic model. We also evaluated the proposed joint decision scheme with "tst2011" test set. Additionally, we built a special small lexical training dataset with the transcripts of the 70 TED Talks used in acoustic training, which contains approximately 80k instances in total, in order to figure out how the lexical model can perform with limited training data.

5.2. Lexical Model Evaluation

We would test the lexical model on ASR transcripts and manual references of TED Talks, which are addressed as "TED-ASR" and "TED-Ref" respectively. In lexical evaluation, we first reported the statistics when comma, period and question marks are treated as separate classes, addressed as 4-Classes test. However, in the sentence boundary detection task, the specific type of punctuation mark might not be as important as the punctuation position. Therefore we further combined all punctuated classes together as "Boundary" for a 2-Classes test.

Table 2: Lexical Model Evaluation (in Percentage)

Test Set	Model	Tr-Size	4-Classes			2-Classes		
			Precision	Recall	F_1	Precision	Recall	F_1
TED-ASR	LSTM-[21]	2.1M	49.1	43.7	46.2	69.3	61.6	65.2
	LMC-1	2.1M	54.4	45.6	49.6	77.5	64.9	70.7
	LMC-2	2.1M	54.0	52.2	**53.1**	76.8	74.2	**75.5**
	LMC-2-80k	80k	45.6	23.5	31.0	77.8	40.1	52.9
TED-Ref	LSTM-[21]	2.1M	55.0	47.3	50.8	75.3	64.6	69.5
	LMC-1	2.1M	60.3	48.6	53.8	85.8	69.2	76.6
	LMC-2	2.1M	60.4	55.8	**58.0**	85.8	79.3	**82.4**

Table 3: Acoustic Model and Joint Solution Evaluation (in Percentage)

Models	Lexical	Acoustic	Joint-S1			Joint-S2		
	(F_1)	(F_1)	Precision	Recall	F_1	Precision	Recall	F_1
LMC-1 + Pause	70.7	60.9	82.8	62.3	71.1	79.2	76.0	77.6
LMC-2 + Pause	75.5	60.9	85.5	62.1	71.9	78.8	79.5	**79.2**
LMC-2-80k + Pause	52.9	60.9	83.5	58.0	68.5	79.2	65.5	71.7
LMC-1 + PPE	70.7	61.0	77.1	67.4	72.0	75.8	76.6	76.2
LMC-2 + PPE	75.5	61.0	79.7	67.5	73.1	76.5	80.6	**78.5**
LMC-2-80k + PPE	52.9	61.0	78.4	62.1	69.3	75.9	66.4	70.8

On both test sets we reported the performances of LMC-1 and LMC-2. Additionally, we also did the tests with the toolkit introduced in [21], which is addressed as LSTM, with exactly same datasets. Please note that the model described in [21] can be divided into 2 stages: stage-1 uses pure lexical features, while stage-2 involves pause information as well. In the lexical model evaluation here, we refered [21] with stage-1 only and the LSTM model contains hidden neurons in hundred-level. Besides, we ran a test on TED-ASR with LMC-2 but the special "80k" training set. Detailed results can be found in Table 2. Please note that all stats we reported have excluded the quantitatively dominating "true negative" samples, which means correctly recognized "not a boundary" occasions. Otherwise the classification accuracy can reach around 89∼93%.

From the statistics we can easily find out that with same dataset, both LMC-1 and LMC-2 outperform LSTM approach. But it should not be neglected that our models contain several thousands of hidden neurons, which is more complicated than the LSTM model in [21]. And the performance of LMC-2 is apparently better than LMC-1, which is also quite understandable. The special test with LMC-2 but only 80k training data on TED-ASR shows clearly that sufficient training data for a lexical approach in this task is crucially important.

5.3. Acoustic Model and Joint Solution Evaluation

We put the acoustic model evaluation together with the joint solution in this chapter. The test set is the same as TED-ASR in lexical evaluation, and the results of the "Pause" and "PPE" acoustic models can be found in "Acoustic" column of Table 3. Since there are only 2 classes available for the acoustic models, we also apply the 2-classes lexical posterior probabilities as the input for the fusion. Based on the lexical and acoustic models we proposed, the testing results of 6 possible combinations are presented in two phases: "Joint-S1" shows the result after the decision scheme stage-1, and "Joint-S2" is the final result.

The performances of two acoustic models are almost the same, both of which are higher than LMC-2-80k, but lower than the others. The results of all combinations after stage-1 are around 10% better than the acoustic benchmark, but when best lexical performer LMC-2 is adopted, the "Joint-S1" result cannot compete with the lexical-only performance yet. We believe it is reasonable, because the stage-1 of the joint decision scheme generally only filters the false positive acoustic detections, resulting in a comparatively high precision but low recall rate, just as shown in Table 3. However, the recall rate can be largely improved by stage-2. In the end, the performance of a joint solution is better than either lexical or acoustic model.

In many previous works, the researchers claimed that pause feature is the dominant acoustic feature in sentence boundary detection task [21, 22, 37, 38]. It is also what our experiment tells us. The "Pause" model works as good as "PPE" model independently, although "PPE" model involves much more information. When working jointly with lexical model, the combinations with a simpler "Pause" model manage to achieve even better result.

In our parallel model structure, the lexical model is more important. With fixed acoustic model, the combination with better lexical model always achieves better result. However, the differences between the performances become smaller after fusing the lexical result with the acoustic model output.

6. Conclusions

In this paper we aim to detect sentence boundaries from unpunctuated speech transcripts. First we developed a lexical model with word vector as only input feature. Then we introduced two simple acoustic models. These models can be applied independently with different training data, but we further proposed a 2-stage joint decision scheme to fuse posterior probabilities. Evaluation shows the high accuracy of the lexical model and the effectiveness of the joint decision scheme. In the future we intend to upgrade our models for better accuracies and extend the availabilities with different languages.

7. Acknowledgements

Hereby we would like to express our gratitude to Mr. Ottokar Tilk, Institute of Cybernetics at Tallinn University of Technology, for sharing their model and experiment results.

8. References

[1] M. Stevenson and R. Gaizauskas, "Experiments on sentence boundary detection," in *Proceedings of the sixth conference on Applied natural language processing*. Association for Computational Linguistics, 2000, pp. 84–89.

[2] E. Matusov, D. Hillard, M. Magimai-Doss, D. Z. Hakkani-Tür, M. Ostendorf, and H. Ney, "Improving speech translation with automatic boundary prediction." in *INTERSPEECH*, vol. 7, 2007, pp. 2449–2452.

[3] C. Fügen and M. Kolss, "The influence of utterance chunking on machine translation performance." in *INTERSPEECH*, 2007, pp. 2837–2840.

[4] J. Tiedemann, "Improved sentence alignment for movie subtitles," in *Proceedings of RANLP*, vol. 7, 2007.

[5] A. Gravano, M. Jansche, and M. Bacchiani, "Restoring punctuation and capitalization in transcribed speech," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 4741–4744.

[6] W. Lu and H. T. Ng, "Better punctuation prediction with dynamic conditional random fields," in *Proceedings of the 2010 conference on empirical methods in natural language processing*. Association for Computational Linguistics, 2010, pp. 177–186.

[7] N. Ueffing, M. Bisani, and P. Vozila, "Improved models for automatic punctuation prediction for spoken and written text." in *INTERSPEECH*, 2013, pp. 3097–3101.

[8] D. Zhang, S. Wu, N. Yang, and M. Li, "Punctuation prediction with transition-based parsing." in *ACL (1)*, 2013, pp. 752–760.

[9] L. Xie, C. Xu, and X. Wang, "Prosody-based sentence boundary detection in chinese broadcast news," in *Chinese Spoken Language Processing (ISCSLP), 2012 8th International Symposium on*. IEEE, 2012, pp. 261–265.

[10] T. Levy, V. Silber-Varod, and A. Moyal, "The effect of pitch, intensity and pause duration in punctuation detection," in *Electrical & Electronics Engineers in Israel (IEEEI), 2012 IEEE 27th Convention of*. IEEE, 2012, pp. 1–4.

[11] M. Sinclair, P. Bell, A. Birch, and F. McInnes, "A semi-markov model for speech segmentation with an utterance-break prior." in *INTERSPEECH*, 2014, pp. 2351–2355.

[12] E. Shriberg, A. Stolcke, D. Hakkani-Tür, and G. Tür, "Prosody-based automatic segmentation of speech into sentences and topics," *Speech communication*, vol. 32, no. 1, pp. 127–154, 2000.

[13] B. Favre, D. Hakkani-Tür, S. Petrov, and D. Klein, "Efficient sentence segmentation using syntactic features," in *Spoken Language Technology Workshop, 2008. SLT 2008. IEEE*. IEEE, 2008, pp. 77–80.

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

[15] X. Wang, H. T. Ng, and K. C. Sim, "Dynamic conditional random fields for joint sentence boundary and punctuation prediction." in *INTERSPEECH*, 2012, pp. 1384–1387.

[16] M. Hasan, R. Doddipatla, and T. Hain, "Multi-pass sentence-end detection of lecture speech." in *INTERSPEECH*, 2014, pp. 2902–2906.

[17] Y. Liu, A. Stolcke, E. Shriberg, and M. Harper, "Using conditional random fields for sentence boundary detection in speech," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2005, pp. 451–458.

[18] M. Zimmerman, D. Hakkani-Tür, J. Fung, N. Mirghafori, L. Gottlieb, E. Shriberg, and Y. Liu, "The icsi+ multilingual sentence segmentation system," DTIC Document, Tech. Rep., 2006.

[19] J. Kolár and L. Lamel, "Development and evaluation of automatic punctuation for french and english speech-to-text." in *INTERSPEECH*, 2012, pp. 1376–1379.

[20] C. Xu, L. Xie, G. Huang, X. Xiao, E. Chng, and H. Li, "A deep neural network approach for sentence boundary detection in broadcast news." in *INTERSPEECH*, 2014, pp. 2887–2891.

[21] O. Tilk and T. Alumäe, "Lstm for punctuation restoration in speech transcripts," in *Sixteenth Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2015.

[22] Y. Gotoh and S. Renals, "Sentence boundary detection in broadcast speech transcripts," 2000.

[23] Y. Liu, E. Shriberg, A. Stolcke, D. Hillard, M. Ostendorf, and M. Harper, "Enriching speech recognition with automatic detection of sentence boundaries and disfluencies," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 5, pp. 1526–1540, 2006.

[24] A. Lee and J. R. Glass, "Sentence detection using multiple annotations." in *INTERSPEECH*, 2012, pp. 1848–1851.

[25] E. Cho, K. Kilgour, J. Niehues, and A. Waibel, "Combination of nn and crf models for joint detection of punctuation and disfluencies," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[26] A. Pappu and A. Stent, "Automatic formatted transcripts for videos," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[27] O. Khomitsevich, P. Chistikov, T. Krivosheeva, N. Epimakhova, and I. Chernykh, "Combining prosodic and lexical classifiers for two-pass punctuation detection in a russian asr system," in *Speech and Computer*. Springer, 2015, pp. 161–169.

[28] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *The Journal of Machine Learning Research*, vol. 3, pp. 1137–1155, 2003.

[29] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations." in *HLT-NAACL*, 2013, pp. 746–751.

[30] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, vol. 12, pp. 1532–1543, 2014.

[31] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *The Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[32] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[33] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics (ACL), 2014.

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

[35] X. Che, C. Wang, H. Yang, and C. Meinel, "Punctuation prediction for unsegmented transcript based on word vector," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, 2016.

[36] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the ACM International Conference on Multimedia*. ACM, 2014, pp. 675–678.

[37] 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.

[38] J. Kolář, J. Švec, and J. Psutka, "Automatic punctuation annotation in czech broadcast news speech," *SPECOM´2004*, 2004.