Attentional Parallel RNNs for Generating Punctuation in Transcribed Speech

Alp Öktem[1], Mireia Farrús[1], and Leo Wanner[1,2]
{alp.oktem,mireia.farrus,leo.wanner}@upf.edu

[1] Universitat Pompeu Fabra, Barcelona, Spain
[2] Catalan Institute for Research and Advanced Studies (ICREA), Barcelona, Spain

Abstract. Until very recently, the generation of punctuation marks for automatic speech recognition (ASR) output has been mostly done by looking at the syntactic structure of the recognized utterances. Prosodic cues such as breaks, speech rate, pitch intonation that influence placing of punctuation marks on speech transcripts have been seldom used. We propose a method that uses recurrent neural networks, taking prosodic and lexical information into account in order to predict punctuation marks for raw ASR output. Our experiments show that an attention mechanism over parallel sequences of prosodic cues aligned with transcribed speech improves accuracy of punctuation generation.

Keywords: speech transcription, recurrent neural networks, prosody, punctuation generation, automatic speech recognition

1 Introduction

The introduction of punctuation marks into the automatic speech recognition (ASR) output is an important issue in applications such as automatic transcription/subtitling, speech-to-speech translation, language analysis, etc. Punctuation is essential for grammaticality, readability, and (in the case of a number of different tasks), subsequent processing. Thus, correct sentence segmentation and punctuation of recognized speech improves the quality of machine translation [6,7,24,26], and missing periods and commas in machine generated text results in suboptimal information extraction from speech [13,15]. Also, most of the data-driven parsing models use punctuation as features.

In spoken language, punctuation is influenced by two intertwined linguistic phenomena: (1) syntax and (2) prosody. Syntax determines the distribution of punctuation marks in accordance with the grammar of a language. Prosody realization in speech (such as, e.g., word grouping, pausing, emphasis, rising-falling intonation, etc.) tends also to signal the position and type of the punctuation marks. For instance, a pause after consecutive words might signal an enumeration, which requires comma, and rising intonation at the end of a sentence is a likely indicator of a question.

However, state-of-the-art approaches to punctuation generation are mainly driven by only syntactic (and lexical) criteria. In particular, recent data-driven approaches that use recurrent neural networks (RNN) proved to be competitive due to RNN's ability to capture long and short term syntactic dependencies. Models that account for prosodic features [30, 31] rely merely on pause duration between words; other prosodic features such as fundamental frequency (f0) and intensity information are ignored. Another shortcoming of the state-of-the-art is that the models are trained on either only written data [2] or on a combination of written and spoken data (with, again, a dominance of written material) [31]. This makes the trained models biased towards written data.

In what follows, we present a neural network setup that is able to process lexical and prosodic information in parallel for punctuation generation in raw speech data. This is different to, e.g., [31], which processes syntactic and prosodic information in sequence (and thus loses the linguistic evidence that both are intertwined). The proposed model makes it possible to integrate any desired feature (be it lexical, syntactic or prosodic) and allows us to test which prosodic features influence punctuation placement to what extent. Unlike previous works, we furthermore use in our experiments only spoken data and exploit various prosodic features that influence the usage of punctuation marks in automated transcriptions. The source code of our model is made publicly available together with a link to the dataset we used in our experiments in `https://github.com/TalnUPF/punkProse`.

The remainder of the paper is structured as follows. In Section 2, we describe the main architecture of our model. The experimental setup and the results of the experiments are outlined in Section 3 and discussed in Section 4. Section 5 summarizes briefly recent related work, and, finally, Section 6 concludes the paper and sketches some of the main lines proposed for future work.

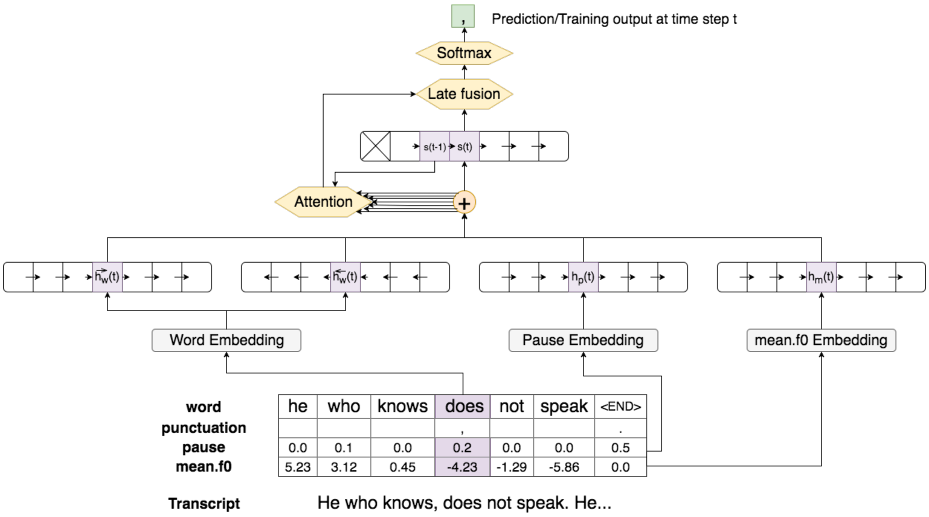
2 Our Model

Our model is inspired by Tilk et al.'s work [31]. Tilk et al. use a bidirectional recurrent network [27] for keeping track of the word context in two directions. Their model is a two-stage model. In the first stage, syntactic and lexical features are processed. In the second stage, pauses between words (as prosodic features) are also taken into account.

As Tilk et al., we use *gated recurrent units* (GRU) [8] for the RNN layers. Introduced as a simpler variate of *long short-term memory* (LSTM) units [11], GRUs make computation simpler by having fewer parameters. Number of gates in hidden units are reduced to two: (a) the *reset* gate determines whether the previous memory will be ignored, and (b) the *update* gate determines how much of the previous memory will be carried on.

Our modification to their proposal is that instead of passing continuous prosodic feature values to the second stage, we discretize the feature values and input them to the model through separate parallel GRU layers that are tuned in one single stage. Figure 1 illustrates our model.

For the sake of simplicity, we assume that the model is trained only with sequences of words (w), pause durations (p) and mean fundamental frequency (m). In this setting, the model has 4 GRU units: bidirectional layers for words,



word	he	who	knows	does	not	speak	<END>
punctuation				,			.
pause	0.0	0.1	0.0	0.2	0.0	0.0	0.5
mean.f0	5.23	3.12	0.45	-4.23	-1.29	-5.86	0.0

Transcript He who knows, does not speak. He...

Fig. 1: Our neural network architecture depicting processing of a speech data sample with pause and mean f0 features aligned at the word level

a unidirectional layer for pauses coming before the words, and a unidirectional layer for mean f0 values of words. GRU layers are preceded by embedding layers for words (W_e), pauses (W_p) and mean f0 (W_m). Inputs to the embedding layers are one-hot encoded vectors of sizes respective to their vocabulary sizes. The hidden states of the GRU layers at time step t are:

$$\begin{aligned}
\overrightarrow{h_w}(t) &= GRU(x(t)W_e,\ \overrightarrow{h_w}(t-1)) \\
\overleftarrow{h_w}(t) &= GRU(x(t)W_e,\ \overleftarrow{h_w}(t+1)) \\
h_p(t) &= GRU(p(t)W_p,\ h_p(t-1)) \\
h_m(t) &= GRU(m(t)W_m,\ h_m(t-1))
\end{aligned}$$

where $x(t)$, $p(t)$ and $m(t)$ are the word index, pause level and mean f0 level respectively at time step t. The parallel GRU states are concatenated to form the context vector $h(t)$ before being passed over as input to another unidirectional GRU layer:

$$\begin{aligned}
h(t) &= \left[\overrightarrow{h_w}(t), \overleftarrow{h_w}(t), h_p(t), h_m(t)\right] \\
s(t) &= GRU(h(t),\ s(t-1))
\end{aligned}$$

The attention mechanism combines all input states into a weighted context vector $a(t)$ which is then late-fused with the state $s(t)$ of the output GRU:

$$a(t) \;=\; \sum_{i=1}^{N} h(t)\alpha_{t,i}$$

$$f(t) \;=\; a(t)W_{fa} \bigodot \sigma(a(t)W_{fa}W_{ff} \;+\; s(t)W_{fs} \;+\; b_f) \;+\; s(t)$$

where $\alpha_{t,i}$ is the weight that determines the amount of influence of each input state to the current output and N is sequence size.

The attention mechanism is useful for the neural network to identify positions in a sequence where important information is concentrated [1]. For words, it helps to focus on positions of words and word combinations that signal the introduction of a punctuation mark. For prosodic features, it either remembers a salient point in the sequence or detects a certain movement that could help determining a punctuation mark at a certain position.

The output GRU layer uses a late-fusion approach, which lets the context gradient carry on easily by preventing it passing through many activation functions [33].

Finally, the late-fused context $f(t)$ is passed through a *Softmax* layer, which outputs the probability of the punctuation mark to be placed between the current and the previous word (starting from the second word in sequence):

$$y(t) \;=\; Softmax(f(t)W_y \;+\; b_y)$$

3 Experiments

3.1 Data

The experiments presented in this paper were performed on a corpus consisting of TED (Technology, Entertainment, Design) talks[1]. TED talks are a set of conference talks lasting approximately 15 minutes each that have been held worldwide in more than 100 languages. They include a large variety of topics, from technology and design to science, culture and academia. The corpus consists of 1046 talks by 884 English speakers, uttering a total amount of 156034 sentences. The corresponding transcripts, as well as audio and video files, are available on TED's website; they were created by volunteers and include punctuation and paragraph breaks [12]. The subtitle timings of TED transcripts do not always correspond to sentences in the transcript. To overcome this limitation, precise word timings were first obtained through Viterbi forced alignment using an automatic speech recognition system. The word timings were then further used to automatically obtain sentence boundaries and thus sentence timings [12].

As for the prosodic features, three main prosodic elements were extracted following the methodology in [12] in order to analyze their influence on punctuation generation: pauses, fundamental frequency (f0), and intensity. Pause

[1] http://www.ted.com

durations were extracted from the provided word timings, while f0 and intensity contours were extracted at 10 ms precision using Praat software [5] with linear interpolation and octave jump removal for fundamental frequency provided by Praat. f0 measurements were converted to semitones relative to speaker mean f0 value for normalization, while the speaker mean intensity over a talk was subtracted from the intensity values for the same purpose, so that speaker mean values were represented by zero values in both cases.

3.2 Features extraction and Preprocessing

The prosodic TED corpus is processed in order to be fed into the neural network. Firstly, the following aligned sequences are extracted for each talk:

word stands for the words that are uttered by the speaker. Abbreviations are decomposed into the letters they consist of (e.g., 'DIY' to three separate words 'D' 'I' 'Y'). Numbers are converted into text and separated (e.g., '93' to 'ninety three').

punctuation marks the symbol coming before the corresponding word. We limited the symbol vocabulary to period ('.'), comma(','), question mark ('?'), exclamation mark ('!'), colon(':'), semicolon(';'), dash('-') and 'no punctuation'. In the cases when more than one punctuation mark occur before a word (e.g., in a quotation), the most important punctuation mark is chosen as the symbol at that position.

reduced-punctuation is the reduced version of *punctuation.* Exclamation mark, dash, colon, and semicolon are mapped to a period.

pause holds the silence duration in milliseconds coming before the corresponding word. It is calculated from the word timings information obtained from speech alignment.

mean.f0 **and** *mean.i0* are the mean fundamental frequency and intensity values (in semitones) for the corresponding word.

range.f0 **and** *range.i0* are calculated by subtracting the minimum f0/intensity value from the maximum f0/intensity value for the corresponding word.

Secondly, taking into account that the number of words per sentence in our corpus is 15-20 in average, the data is sampled into sequences of size 50, each sample starting with a new sentence and ending with an END token. With this setting, more than one sentence fits into a sample. Sentences are placed in samples in the same order in the speech data. If the sample end is reached before the end of a sentence, the sentence portion that fits is kept in that sample and the next sample starts from the beginning of that sentence. We avoided putting together data from different talks in the same sample by discarding the last unfinished sample from a talk. Also, sentences with more than 50 words are discarded.

59811 samples were extracted this way. 70% percent (41867 samples) of this data were allocated for training, 15% for testing and 15% for validation (8971 samples each).

The word vocabulary is created with the tokens that occur more than 7 times in the corpus and two extra tokens: *out-of-vocabulary* and *end-of-sequence.*

This totaled up to 13830 tokens. The output punctuation vocabulary in our experiments is of size 4 (from the reduced punctuation set).

In order to input prosodic features to the RNNs, they had to be vocabularized as well. This is achieved by assigning a vocabulary index for certain ranges of the continuous feature values. The ranges were determined by dividing the feature value distribution according to the number of occurrences within that range. Via manual inspection, we divided the pause durations into 66 and semitone values distribution into 81 levels.

3.3 Experimental Setup

We used Theano [29] for implementing our models. In our experimental setup, the embedding vector sizes for words and prosodic features are set to 100 and 10 respectively. This is because prosodic feature vocabulary is significantly smaller than the word vocabulary. The hidden layer dimension of all GRU layers is also set to 100, except for pause durations, where a smaller dimension of 10 performed better in terms of validation scores, such that we set it to 10.

The model is trained in batches of size 128. The weight matrices are updated using the AdaGrad algorithm [10] with a learning rate of 0.05 for minimizing the negative log-likelihood of the predicted punctuation sequence.

3.4 Punctuation Generation Results

As the majority of the punctuation marks in our dataset consisted of the punctuation marks in the reduced set (comma, period and question mark), experiments were performed only with this set.

The two-stage method by Tilk et al. is used as a baseline by training over our data twice: first, only with text, and then together with the pause durations. Tilk et al.'s models are based on BRNN with an attention mechanism, which provided the best results when compared to other models [31].

In our single stage approach, the use of only lexical information (words) provided the same scores as the use of only words in the two-stages approach, since only one step is involved in both approaches. Then, in order to assess the contribution of new prosodic information to our model, the extracted prosodic features were added one by one. The pause duration feature was always kept while trying combinations of new features, i.e., means and ranges of both f0 and intensity. The outcomes of our experiments in generating periods, commas and question marks are presented in Table 1 in terms of precision (P), recall (R), and F_1 scores.

4 Discussion

A significant improvement is achieved with the proposed parallel RNNs approach compared to the two-stage model when trained with the same dataset. We observe an overall improvement in F_1 score of 5,8% when same features (word

Table 1: Punctuation generation results for two stages [31] and our single-stage approach

Model	Features	Comma			Period			Question			All		
		P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Two stages	word(w)	56.9	36.6	44.5	67.6	62.5	64.9	68.5	46.9	55.7	63.2	49.0	55.2
	w+pause(p)	51.0	51.6	51.3	68.6	57.8	62.8	66.8	48.9	56.5	58.9	54.4	56.6
Single stage	w+p	61.6	44.5	**65.6**	71.7	72.5	72.1	66.5	64.7	65.6	67.3	58.2	62.4
	w+p+range.f0	58.7	52.0	55.1	72.4	76.1	74.2	67.9	64.7	66.3	65.9	63.6	64.8
	w+p+mean.f0	59.3	53.3	56.1	74.9	75.9	75.4	65.2	**67.4**	66.3	67.2	**64.3**	**65.7**
	w+p+range.i0	55.0	**54.3**	54.6	**75.0**	70.3	72.5	70.0	58.7	63.9	64.5	61.9	63.2
	w+p+mean.i0	58.4	53.4	55.8	74.5	74.3	74.4	68.8	63.9	66.3	66.6	63.5	65.0
	w+p+range.f0+range.i0	60.9	45.5	52.1	71.9	76.0	73.9	71.5	61.0	65.9	67.3	60.2	63.6
	w+p+range.f0+mean.i0	61.2	46.6	53.0	72.9	77.6	75.2	**74.2**	63.1	**68.2**	68.0	61.6	64.7
	w+p+mean.f0+range.i0	61.6	47.9	53.9	73.1	**79.6**	**76.2**	74.1	62.0	67.5	68.2	63.1	65.6
	w+p+mean.f0+mean.i0	56.9	52.2	54.4	77.1	70.4	73.6	71.3	61.6	66.1	66.7	60.9	63.7
	w+p+mean.f0 +range.f0+range.i0	**63.4**	44.5	52.3	73.6	77.4	75.5	65.7	66.4	66.1	**69.2**	60.5	64.6

and pause durations) are used with our model. The model opens the way for a further improvement of 3.3% with the addition of mean f0 feature into the model, resulting in an overall F_1 score of 65.7%.

We also see from the results that the inclusion of f0- and intensity-related prosodic features —apart from pauses— into the neural network improves the generation score for period and question marks. An improvement of 4,1% in F_1 score is observed for periods with the inclusion of mean f0 and intensity range features on top of pause features. For question marks, the best F_1 score is achieved with f0 range and mean intensity features on top of pause durations (improvement of 2,6%). For commas, we observe that precision and recall improve with different settings but when looked at the F_1 score best feature combination stays to be words and pause durations.

The best performing set of features seems to be the combination of pause and mean f0 when looked at the overall F_1 score. However, we see that each punctuation mark has a different set of features that improve their generation results the most. Combination of f0 range and mean of intensity gives best results for generating question marks (68.2% in F_1 score). For period, using mean f0 and intensity range features together yields the best result (76.2% in F_1 score). Recall that colons, semicolons, dashes and exclamation marks in our dataset are also mapped to periods.

It has to be stated that our evaluation method for the baseline does not corroborate the design decision of Tilk et al. Their two stage training helps building a more solid lexical model by training on a larger text corpus. In [31], they report an overall F_1 score of 72.2% trained only on written textual data, which further improves to an F_1 score of 77% with additional training on pause-annotated corpus. However for our purely spoken data, their model performs with an F_1 score of 56.6% which shows only an 1.4% improvement after the addition of pause features.

Our initial guess was that training with four prosodic features at once would oversaturate the model; however, the results for the feature set consisting of mean f0, f0 range and intensity range combined gives promising results. The best overall precision score (69.2%) and precision for generating commas (63.4%) are achieved with this feature set.

5 Related Work

The problem of punctuation determination has been addressed in several works in the literature — as has been the closely-related issue of boundary detection. Both problems have been tackled from diverse perspectives, and many of them only take into consideration the recognized ASR output text, ignoring the speech related information contained in the original speech, or they simply tackle the problem for textual data in which the correct punctuation is missing, e.g., in a sentence generation or a grammatical correction scenario. In [16], for instance, the punctuation detection is addressed from a syntax-based perspective by using the output of an adapted chart parser, which provides information on the expected punctuation placement. Also in [32] and in [23] the punctuation generation task is carried out without taking prosodic cues into account. In the former, several textual features including language model scores, token n-grams, sentence length and syntactic information extracted from parse trees are combined using conditional random fields (CRF). In the latter, the task is based on dynamic conditional random fields and applied to a conversational speech domain. A more recent work [2] introduces a language-independent model with a transition-based algorithm using LSTMs [11], without any additional syntactic features.

Overall, it has been shown that prosodic features are highly indicative of sentence boundaries as well as of punctuation placement. Therefore, a great deal of effort has been put in several works into the use of prosodic features when original speech is available. In [3], sentence boundaries are characterized by prosodic features and modeled by decision tree classifiers. In [20], the authors successfully detect automatically full stops by using a neural network to estimate the weights assigned to pauses, f0 changes and amplitude range, which are later used by a punctuation mark classifier; commas are shown to be more difficult to detect.

Other studies, such as [17], combine prosodic, word and grammatical features by using SVM and CRF classifiers, and test the prediction experiments on different speech styles, validating the hypothesis that the punctuation problem is much more difficult to address in ASR output than in manual transcripts. Prosodic and textual cues are also combined in [22] and implemented in a decision tree classifier with the goal to detect sentence boundaries. A combination of lexical-, prosodic-, and speaker-based features is also found in [4] for the detection of full stops, commas, and question marks in a bilingual English-Portuguese broadcast news data, while [19] focuses on Czech broadcast news speech to detect commas and sentence boundaries by using a prosodic model in a decision trees and a multi-layer perceptron and N-gram models for language modeling. Similar works deal with the punctuation generation problem by using statistical models of prosodic features [9], the combination of both textual and prosodic features based on adaptive boosting [18], and a cross-linguistic study of prosodic features through two different approaches for feature selection: a forward search wrapper and feature filtering [14]. Although not using prosodic features strictly speaking, [25] takes advantage of the transcriptions of multiple parallel speech streams in four different languages in order to increase punctuation generation accuracy.

More recently, the already mentioned work by Tilk et al. addresses the use of textual features and pauses (as sole prosodic feature) in an LSTM recurrent neural network [30] and in a bidirectional recurrent neural network [31] in order to detect full stops and commas in the former, and also question marks in the latter. As already discussed above, Tilk et al. 's methodology combines syntactic and prosodic features in a two-stage model. Only textual features are learned from a large non-spoken text corpus in a first stage. Then, in a second stage, the model is retrained with pause durations on a smaller corpus. This approach follows the work from [28], in which the language model can be trained on large amounts of textual data—lacking of the corresponding spoken data—, while the acoustic model —also based only on pause duration—is trained on a smaller corpus.

6 Conclusions and Future Work

In this work, we have presented a recurrent neural network architecture that processes lexical and prosodic information in parallel for the generation of punctuation, avoiding the dominance of written data, and thus the bias of trained models towards written material. Our proposed model allows the integration of any desired feature (lexical, syntactic or prosodic) and thus a further analysis of the impact of every feature used on the punctuation generation. In addition, the current model achieves a significant improvement over previous works that used two stages and were biased to written data.

The results are significantly better also when prosodic features are added to the lexical information. Solely pauses — when trained with a separate RNN — improve considerably the lexical-based scores. Moreover, f0- and intensity-based prosodic features help to achieve a better period and question mark detection in terms of F_1 measure, and comma detection is improved in terms of precision and recall in some specific settings. All in all, the best combination of prosodic features is when our model is trained on words together with the preceding pause durations and their normalized mean f0 values.

As future work, we plan to experiment with more prosodic features (such as speech rate) and their combinations and also see whether other RNN types such as LSTM help solve the problem better. Also, a model that gives attention to different prosodic features for different punctuation marks is a field to explore.

Our model trains word embeddings together with the whole architecture. We believe that pre-trained word embeddings extracted from a larger speech corpus would improve the scores. Also it has been recently shown that character-based encodings improve results in neural network based applications by largely decreasing the word vocabulary size [21].

7 Acknowledgements

We would like to thank Francesco Barbieri for offering his technical insights throughout this work. This work is part of the KRISTINA project, which has received funding from the *European Union's Horizon 2020 Research and Innovation Programme* under the Grant Agreement number H2020-RIA-645012. The second author is partially funded by the Spanish Ministry of Economy, Industry and Competitiveness through the *Ramón y Cajal* program.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. CoRR abs/1409.0473 (2014), `http://arxiv.org/abs/1409.0473`
2. Ballesteros, M., Wanner, L.: A neural network architecture for multilingual punctuation generation. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (2016)
3. Baron, D., Shriberg, E., Stolcke, A.: Automatic punctuation and disfluency detection in multi-party meetings using prosodic and lexical cues. Channels 20(61), 41 (2002)
4. Batista, F., Moniz, H., Trancoso, I., Mamede, N.: Bilingual experiments on automatic recovery of capitalization and punctuation of automatic speech transcripts. IEEE Transactions on Audio, Speech, and Language Processing 20(2), 474–485 (2012)
5. Boersma, P., Weenink, D.: Praat: doing phonetics by computer [computer program]. `http://www.praat.org/` (2016)
6. Cho, E., Niehues, J., Kilgour, K., Waibel, A.: Punctuation insertion for real-time spoken language translation. In: Proceedings of the Eleventh International Workshop on Spoken Language Translation (2015)
7. Cho, E., Niehues, J., Waibel, A.: Segmentation and punctuation prediction in speech language translation using a monolingual translation system. In: International Workshop on Spoken Language Translation (IWSLT) 2012 (2012)
8. Cho, K., van Merrienboer, B., Gülçehre, Ç., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. CoRR abs/1406.1078 (2014), `http://arxiv.org/abs/1406.1078`
9. Christensen, H., Gotoh, Y., Renals, S.: Punctuation annotation using statistical prosody models. In: in Proc. ISCA Workshop on Prosody in Speech Recognition and Understanding. pp. 35–40 (2001)
10. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. 12, 2121–2159 (Jul 2011), `http://dl.acm.org/citation.cfm?id=1953048.2021068`
11. Dyer, C., Ballesteros, M., Ling, W., Matthews, A., Smith, N.A.: Transition-based dependency parsing with stack long short-term memory. CoRR abs/1505.08075 (2015), `http://arxiv.org/abs/1505.08075`

12. Farrús, M., Lai, C., Moore, J.D.: Paragraph-based Prosodic Cues for Speech Synthesis Applications. In: Proceedings of the 8th International Conference on Speech Prosody (2016)
13. Favre, B., Grishman, R., Hillard, D., Ji, H., Hakkani-Tur, D., Ostendorf, M.: Punctuating speech for information extraction. In: Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on. pp. 5013–5016. IEEE (2008)
14. Fung, J.G., Hakkani-Tür, D., Magimai-Doss, M., Shriberg, E., Cuendet, S., Mirghafori, N.: Cross-linguistic analysis of prosodic features for sentence segmentation. In: Eighth Annual Conference of the International Speech Communication Association (2007)
15. Hillard, D., Huang, Z., Ji, H., Grishman, R., Hakkani-Tur, D., Harper, M., Ostendorf, M., Wang, W.: Impact of automatic comma prediction on pos/name tagging of speech. In: Spoken Language Technology Workshop, 2006. IEEE. pp. 58–61. IEEE (2006)
16. Jakubıcek, M., Horák, A.: Punctuation detection with full syntactic parsing. Special issue: Natural Language Processing and its Applications p. 335 (2010)
17. Khomitsevich, O., Chistikov, P., Krivosheeva, T., Epimakhova, N., Chernykh, I.: Combining prosodic and lexical classifiers for two-pass punctuation detection in a Russian ASR system. In: International Conference on Speech and Computer. pp. 161–169. Springer (2015)
18. Kolář, J., Lamel, L.: Development and evaluation of automatic punctuation for french and english speech-to-text. In: Proceedings of Interspeech. pp. 1376–1379 (2012)
19. Kolář, J., Švec, J., Psutka, J.: Automatic punctuation annotation in Czech broadcast news speech. In: in Proceedings of the SPECOM (2004)
20. Levy, T., Silber-Varod, V., Moyal, A.: The effect of pitch, intensity and pause duration in punctuation detection. In: Electrical & Electronics Engineers in Israel (IEEEI), 2012 IEEE 27th Convention of. pp. 1–4. IEEE (2012)
21. Ling, W., Trancoso, I., Dyer, C., Black, A.W.: Character-based neural machine translation. CoRR abs/1511.04586 (2015)
22. Liu, Y., Chawla, N.V., Harper, M.P., Shriberg, E., Stolcke, A.: A study in machine learning from imbalanced data for sentence boundary detection in speech. Computer Speech & Language 20(4), 468–494 (2006)
23. Lu, W., Ng, H.T.: Better punctuation prediction with dynamic conditional random fields. In: Proceedings of the 2010 conference on empirical methods in natural language processing. pp. 177–186. Association for Computational Linguistics (2010)
24. Matusov, E., Mauser, A., Ney, H.: Automatic sentence segmentation and punctuation prediction for spoken language translation. In: International Workshop on Spoken Language Translation (IWSLT) 2006 (2006)
25. Miranda, J., Neto, J.P., Black, A.W.: Improved punctuation recovery through combination of multiple speech streams. In: Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on. pp. 132–137. IEEE (2013)
26. Peitz, S., Freitag, M., Mauser, A., Ney, H.: Modeling punctuation prediction as machine translation. In: International Workshop on Spoken Language Translation (IWSLT) 2011 (2011)
27. Schuster, M., Paliwal, K.K., General, A.: Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing (1997)
28. Shen, W., Yu, R.P., Seide, F., Wu, J.: Automatic punctuation generation for speech. In: Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on. pp. 586–589. IEEE (2009)

29. Theano Development Team: Theano: A Python framework for fast computation of mathematical expressions. arXiv e-prints abs/1605.02688 (May 2016), http://arxiv.org/abs/1605.02688
30. Tilk, O., Alumäe, T.: LSTM for punctuation restoration in speech transcripts. In: Proceedings of Interspeech. pp. 683–687 (2015)
31. Tilk, O., Alumäe, T.: Bidirectional recurrent neural network with attention mechanism for punctuation restoration. In: Proceedings of Interspeech. pp. 3047–3051 (2016)
32. Ueffing, N., Bisani, M., Vozila, P.: Improved models for automatic punctuation prediction for spoken and written text. In: INTERSPEECH. pp. 3097–3101 (2013)
33. Wang, T., Cho, K.: Larger-context language modelling. CoRR abs/1511.03729 (2015), http://arxiv.org/abs/1511.03729