An Audio-based Sequential Punctuation Model for ASR and its Effect on Human Readability

György Szaszák

Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics
Magyar tudósok krt. 2, H-1117 Budapest, Hungary
E-mail: szaszak@tmit.bme.hu

Abstract: Inserting punctuation marks into the word chain hypothesis produced by automatic speech recognition (ASR) has long been a neglected task. In several application domains of ASR, real-time punctuation is, however, vital to improve human readability. The paper proposes and evaluates a prosody inspired approach and a phrase sequence model implemented as a recurrent neural network to predict the punctuation marks from the audio. In a very basic and lightweight modeling framework, we show that punctuation is possible by state-of-the-art performance, solely based on the audio signal for speech close to read quality. We test the approach on more spontaneous speaking styles and on ASR transcripts which may contain word errors. A subjective evaluation is also carried out to quantify the benefits of the punctuation on human readability, and we also show that when a critical punctuation accuracy is reached, humans are not able to distinguish automatic and human produced punctuation, even if the former may contain punctuation errors.

Keywords: punctuation; prosody; speech recognition; recurrent neural network; human readability

1 Introduction

Most Automatic Speech Recognition (ASR) systems treat speech as a word sequence, and then, based on acoustic models (e.g. phonemes) and a language model (typically an N-gram), a so-called recognition network is created, along which the speech frames are aligned using the Viterbi-algorithm. The recognition hypothesis is yield by the most likely alignment path in the network, given the acoustic observation (e.g. speech frames). Despite the advanced search space reduction techniques (beam-search and pruning), decoding is still computationally expensive as the recognition network is usually quite complex for tasks such as dictation and closed captioning.

In this framework, inserting punctuation marks into the word sequence hypothesis has long been neglected, as research was mostly concerned by reducing word error rates and augmenting transcription accuracy for the word chain. On the other hand, punctuation is not relevant in applications where the text output is not directly required, but rather the system is expected to react according to the received commands or queries. In dictation systems, where punctuation is the most relevant, a telegraphic style alike explicit dictation of the punctuation marks was foreseen, similarly to commands intended to provide text formatting, i.e. "*SET_BOLD SET_INITIAL_CAPITALS dear mister smith COMMA SET_NORMAL NEWLINE ...*". Nevertheless, providing punctuation automatically is the only applicable approach in several use-cases, i.e. for closed captioning of audio data (subtitling), transcription of meeting records, audio indexing followed by text analysis, etc. In dictation systems, also, it is more natural and easier to speak normally, whereby the system automatically detects where punctuation is necessary.

In ASR, two main approaches can be applied or combined for punctuation insertion. *Text based punctuation* (hereafter TBP) approaches exploit word context dependency of the punctuation marks (for example, conjunction words are usually preceded by commas), whereas *audio based punctuation* approaches (hereafter ABP) exploit acoustic markers which correlate with clause or sentence boundaries.

Speech prosody is the most often used feature in ABP as prosody is known to reflect the information structure of the speech to some extent [16]. Features representing intonation, stress and pausing (F0 slopes and trends, pause durations) are found to be the most effective [3, 6]. ABP approaches have the advantage of being independent of ASR errors, albeit usually yield weaker performance than TBP approaches.

Regarding TBP, a straightforward way is to use N-gram language models enhanced with punctuation marks [5, 18], optionally complemented by involving the modeling of non-word events in the acoustic models [4]. A considerable drawback of using enhanced N-grams may be however, that punctuations are usually missing from human made speech transcripts associated with training corpora. Alternative approaches have been also proposed, based on the paradigm of sequence to sequence modelling with either Hidden Markov Models (HMM), maximum entropy models or conditional random fields, etc. [4, 10]. Recently, sequence to sequence approaches based on Recurrent Neural Networks (RNN) have been proposed, which first project the context words into an embedding space able to represent syntactic and semantic relations, and then predict the punctuation for a very long (~100) word sequence. Albeit still computationally expensive, these models yield the highest accuracy in state-of-the-art punctuation of transcripts by low word error rates. However, they often rely on *future* context, which obviously turns into high latency (wait for future tokens before processing the current ones) and hence, these models are not suitable for scenarios where either real-time operation or resource efficiency are required [23], or if the ASR works with higher word error rates.

The present study is interested in providing a lightweight, automatic punctuation approach with real-time capabilities. We rely exclusively on acoustic cues, and minimize latency and resource demand prior to maximizing punctuation accuracy. We do this inspired by the paradigm of cognitive infocommunication [1], i.e. we expect the human brain to "repair" part of the punctuation errors if a sufficient amount of punctuations is predicted correctly. We suppose that precision is more crucial in this sense than recall, i.e. false detections should be minimized, and human reader should rather be required to insert missing punctuations "in mind" than eliminating incorrect ones, the latter being more disturbing from a perception point-of-view [14]. In other words, we expect the human brain to interact with the automatic process and repair an amount of (tolerable) errors [2]. We suppose that a certain amount of punctuation errors is recoverable quasi unconsciously by the user, hence it is sufficient to provide a "good enough" punctuation for acceptable user satisfaction [21]. To further investigate this aspect, we also carry out subjective evaluation tests and hypothesize that (i) human readers are less sensitive to punctuation errors than to ASR errors; and that (ii) by low punctuation error rates, readers are not able to distinguish machine made (with some errors) and human made (error free) punctuation.

In this paper we first present an acoustic-prosodic phonological phrasing approach, which is used to extract prosodic markers, expected to reflect the information structure and hence punctuation of the word sequence. Thereafter, we propose a tiny RNN punctuation model exploiting the phonological phrase sequence and its characteristics. An experimental evaluation is presented for Hungarian, completed by subjective tests (Mean Opinion Score, MOS) to compare machine and human made punctuation from a perceptual point-of-view.

2 Feature Extraction

We do not use acoustic cues directly, but rather adopt an approach by which we obtain a phonological phrasing quickly and automatically. The phonological phrase (hereafter PP) is defined as a prosodic unit, which is characterized by a single occurrence of stress [16], in other words, it is a unit that lasts from stress to stress. In the prosodic hierarchy, PPs are situated between the better known intonational phrase and prosodic word levels. The strength and the place of the stress within the PP, as well as its intonational contour may vary, depending on higher, utterance or intonational phrase level constraints, leading us to a tiny inventory of PP types (see Table 1).

Table 1
PP inventory for Hungarian

Label	Stress	Location	Intonational shape
io	strong	IP initial	IP onset + descending
ss	strong	IP internal	Prominence + descending
ms	medium	IP internal	Prominence + descending
ie	medium	IP terminal	Prominence + descending
cr	medium	IP terminal	Prominence + ascending (continuation rise)
ls	neutral	IP initial	Descending (without initial stress)
sil	neutral	N.A.	Silence

2.1 Phonological Phrasing

In [24], a Hidden Markov Model (HMM) based approach was proposed, further enhanced by [19], to automatically recover the PP structure of speech utterances. The algorithm involves a modelling step carried out by machine learning for the 7 different PP models in Hungarian for declarative modality (as presented in Table 1, [19]), and an alignment step to recover the phrase structure.

The PP models use directly the acoustic-prosodic features, i.e. continuous F0 and energy streams, with added deltas calculated with several different time spans in order to represent short and long-term tendencies seen in the features (intonational slopes). Each PP type is modelled by a HMM / Gaussian Mixture Model (GMM) composite, where the HMM is responsible for dynamic time warping, and the GMM is used to derive matching likelihoods (or kind of similarity measures). The PP sequence corresponding to the utterance is obtained by Viterbi-alignment as the most likely path through an unweighted and looped network (phrase grammar) of singular PPs. Given the low dimensional acoustic feature set, the low number of mixture components in the GMMs and the simple phrase grammar, the PP alignment process has low resource demand and introduces low latency. The complete PP segmentation system, hereafter called Automatic Phrasing Module (APM) is thoroughly documented [19, 24], hence we refer the reader to these papers for further details and performance evaluation of the APM. Here we briefly mention that precision and recall of phrase boundary recovery is 0.89 for Hungarian on a read speech corpus (for the operation point characterized by equal precision and recall).

2.2 Phrase Density

Speech prosody, especially the F0 contour is characterized by prominent sections (local maxima can be spotted in the visualized F0 track). Prominence can be associated with prosodic stress (or accent in case of the F0 track), but micro-prosodic variation can also occur as a byproduct (noise) of the speech production process, especially voiced plosives may lead to a slight F0 peak. If the prominence is considerable, it can be regarded to infer stress exclusively. However, slight prominence may result either from secondary stress or microprosodic effects.

The sensitivity of the APM can be tuned whether it reacts to only the strong or also to the slight prominence. In the Viterbi-algorithm, this tuning parameter is called *insertion likelihood*. The higher this value is set, the more the PPs tend to split up to sub-phrases recursively, i.e. the denser the alignment will be. From ABP perspective, an optimization step is required to determine the optimal phrase density, which we will carry out and evaluate in the *Results* section.

2.3 Matching the PP Sequence with Word Boundaries

It is very important to notice that the boundaries of PPs usually coincide with word boundaries, especially in fixed stress languages such as Hungarian. Therefore, in the APM, we constrain PPs to start and end at word boundaries. This results in a word sequence, segmented for PPs: a PP may spread over several words, but contains at least one word. Readers interested in the correspondence between sentence level syntax and PP structure are referred to [12] and [20].

3 The Punctuation Model

The proposed punctuation model exploits the expected correlation between phonological phrasing and punctuation marks. As the phonological phrasing represents the building blocks of sentence level intonation, we model them as a sequence and map this sequence to the sequence of the punctuation marks. The most suitable machine learning framework for such tasks is using recurrent neural networks with Long-Short Term Memory cells (LSTM).

LSTM networks [17] are built up from cells which contain a memory unit, preserving past states of the cell. The memory unit itself, as well as the output of the cell combined from a weighted contribution of the current input and the memory unit, are regulated by the data flow. These regulating weights are learned during the training phase. Connecting LSTM cells sequentially leads to powerful sequential models, whereby typically each cell receives the features at a given time frame. It is common to incorporate future features into the processing framework, that is, the output of the network at time *t* depends on inputs ranging from *t-k* .. *t* .. *t+k*. This is usually more effective if we allow for a bidirectional (from past to future and from future to past) flow of the information within the network (e.g. Bidirectional LSTM, BiLSTM). Obviously, the future is not known, so technically such networks wait until future samples become available, and delay their output accordingly. For reasons explained in the *Introduction*, we have to limit this future context to preserve low latency operation of the model.

3.1 Phrase Sequence Features

We start from the automatic PP alignment and the word sequence, which are supposed to be known (as PP sequence hypothesis from APM and word sequence hypothesis from ASR). As said before, PPs are constrained to start and end on word boundaries, as punctuation marks may also be required at word boundaries (so called *slots*). Then, we extract the following features to be input to the RNN:

- the type of the PP (PP_{label})
- the duration of the PP (PP_{dur})
- the duration of short pause or silence following the PP (SIL_{dur})

These features build up a phrase sequence representation and are used as input to the RNN model.

3.2 The RNN Model

From the feature sequence, we use *k* samples (pp_1, pp_2, ... pp_k; $4 < k < 16$) at once, then we move on to the next sample (appending it) and drop the first one from the sequence. These are input to a bidirectional LSTM layer, followed by another similar layer. The first layer is composed of 20 LSTM units with sigmoid inner activation and RELU output activation. Dropout is set to 0.3. The second layer has 40 LSTM units and a dropout of 0.25 [13]. The output is derived from a fully connected layer using softmax activation, which yields posteriors for the modelled punctuation marks for the slot located between pp_{k-1} and pp_k. This means that the past context consists of *k-1* PPs, and a single PP represents the future context.

The RNN is trained with the Adam optimizer by using adaptive estimates of lower-order moments [7]. We perform up to 30 epochs, but also apply early stopping with a patience of 5 epochs to prevent overfitting. Class-weighting is applied to compensate for the imbalanced nature of the data, as there are more empty slots (without punctuation) than slots which require punctuation.

For such a lightweight network, training is not time-consuming; the network can be trained within 3-5 minutes even on CPU on a standard 8 core Intel(R) Core(TM) i5-6600K CPU @ 3.50 GHz workstation. Automatic punctuation requires feature extraction and a forward pass, both with low computational needs.

4 Punctuation Experiments

Implementing the feature extraction and the punctuation model presented so far, we intend to evaluate its performance. We use word error free speech transcription and ASR output test sets, the latter may contain word errors.

Table 2
The used corpora and number(#) of words, PP, comma and period slots

Corpus	Size	# words	# PP slots	# commas	# periods
BABEL	2k utts	20k	7-20k	3k	2k
BN	50 blocks	3k	1.5-3k	300	500

4.1 Speech Corpora

We use Hungarian BABEL [15], a read speech corpus recorded from non-professional native speakers; and a Broadcast News (BN) corpus [25]. BABEL is split up to train, validation and test sets (80%, 10%, 10% of utterances, respectively). The BN corpus is used for testing. Characteristics of the used data sets are presented in Table 2. Please note that the number of PP slots depends on PP density. The APM is also trained on BABEL train set, as well as the RNN punctuation model. The latter is validated on the validation set using the categorical cross-entropy loss function.

4.2 Performance Measures

The punctuation mark set consists of 3 elements: comma, period and empty (none). As question and exclamation marks are heavily underrepresented in the used corpora, we map these to period, as well as semicolons and colons. Dashes and terminal citation quotes are mapped to comma, whereas leading citation marks are removed. For revealing questions based on prosody, [3] proposed an approach; in this paper we focus only on phrasing related comma and sentence terminal period (full stop) recovery.

As performance measures we use retrieval statistics, i.e. precision (PRC), recall (RCL) and F-measure (F1). Actual values depend on the operating point of the system: regarding the extremities, permissive prediction leads to high recall, but also to high false alarm rate, translated into low precision; accepting only predictions with high confidence means high precision, but low recall. The RNN punctuation model yields posteriors for each punctuation class (comma, period, none). Based on these, operation characteristics can be plotted in the precision / recall space.

Additionally, there exists a measure designed uniquely to assess punctuation performance: the Slot Error Rate (SER) [11]. SER is obtained as the ratio of the correctly punctuated word slots vs. all word slots.

In the proposed approach, we predict only for word slots which are located at PP boundaries. All other slots are treated as being of 'none' punctuation. We define a measure, the Slot Miss Ratio (SMR), to evaluate the loss resulting from disregarding word slots with no PP boundary. SMR reflects the number of missed word slots which should have been punctuated with a non-empty mark (in the reference they carry a non-blank punctuation) versus all word slots with non-blank punctuation. Obviously, our goal is to keep SMR low.

5 Results

5.1 Sparse versus Dense PP Alignments

As explained in the respective section, by tuning the sensitivity of the APM, we can control how dense the resulting PP alignment becomes. We hope that the reader can easily deduce from the description provided so far in the paper, that in a dense alignment, phrases with a slight stress and a descending contour (*ms* in Table 1) will dominate in contrast to a sparse alignment, where PPs characteristic for intonational phrase or utterance onsets and endings will be found. Taking into account that we model the sequences of such phrases, PP density becomes a hyperparameter of our model to be optimized. It seems to be obvious that there is no point in augmenting the density of the PP alignment when trespassing a threshold, but still, we are interested in where this threshold can be found, and if there is significant difference in punctuation performance between using a sparse or a dense PP alignment.

Fig. 1 shows operation characteristics for comma and period punctuation based on a dense ($logP_{ins}=0$) and on a sparse ($logP_{ins}=-50$) PP alignment on the BABEL test set. In this scenario, we use reference transcription, (word error free), but perform a forced alignment [11] with the ASR to obtain the word boundaries.

In operating points more relevant for exploitation (high precision), not much difference is seen between a dense and a sparse alignment. From latency perspective, however, the denser the alignment, the lower the latency becomes, as we have to wait the k^{th} PP to terminate for punctuation prediction for the slot between the $k\text{-}1^{th}$ and k^{th} PPs. In a denser alignment, average PP length is lower.

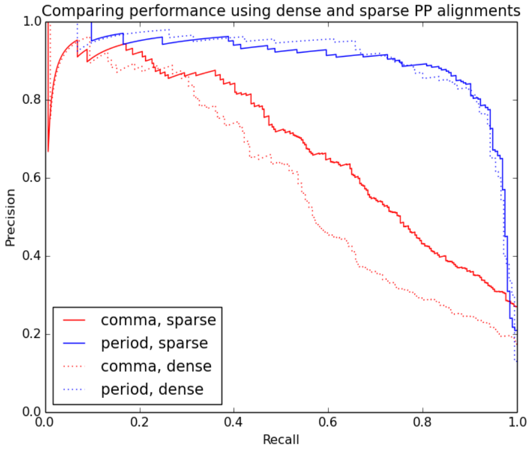

Figure 1
Precision and recall for commas and periods on BABEL based on dense and sparse PP alignments

We also report overall performance metrics for individual operating points completed by SER and SMR in the top 4 rows of Table 3. By decreasing PP density, SMR increases from 2% to 5%. At higher recall rates of the operation curve, especially regarding commas, sparse PP alignment performs better, although we consider that if precision is below a threshold, punctuation errors, even if associated with a higher recall, start to be disturbing for the user and hence we propose to maintain the system operating in the upper left quartile of the PR diagram. Using dense alignment is moreover advantageous from the perspective of SMR as well.

Table 3
Punctuation performance for 4 scenarios with sparse and dense PP alignment densities

Testset	PP density	comma			period			[%]	
		PRC	RCL	F1	PRC	RCL	F1	SER	SMR
BABEL, true transcript	dense	.83	.45	.58	.82	.89	.85	39.4	2.0
	sparse	.81	.42	.55	.85	.86	.85	40.3	5.1
BABEL, ASR transcripts	dense	.74	.44	.56	.83	.83	.83	39.1	6.5
	sparse	.72	.49	.59	.81	.82	.82	38.3	7.3
BN, ASR transcript	dense	.43	.38	.40	.76	.73	.75	51.2	7.2
	sparse	.45	.38	.41	.77	.77	.77	54.8	9.7
BN, ASR + adapt RNN	dense	.55	.32	.41	.80	.74	.77	45.5	6.5
	sparse	.82	.25	.38	.80	.76	.78	51.3	9.0

5.2 Feature Analysis

We are also interested in the contribution of the different features to punctuation performance. Therefore, in Fig. 2 we present operational characteristics for the cases when (i) only the type of the PP (PP_{label}) is used as RNN input, (ii) when we add the duration of the PP (PP_{dur}) and (iii) when we use the three alltogehter. In Fig. 2 we can see that we obtain a big ratio of classification power from SIL_{dur}, that is the length of the pause following the PP. The length of the PP itself does not lead to significant improvement when added to PP type feature.

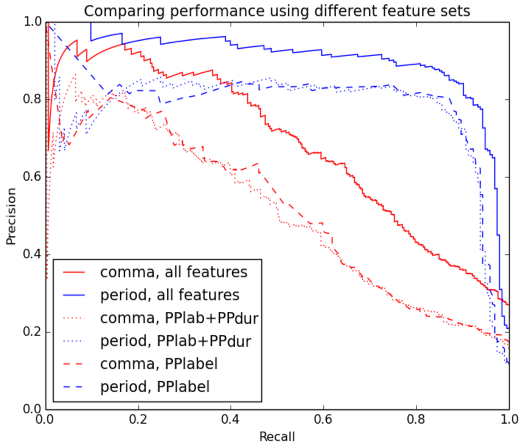

Figure 2
Precision and recall for commas and periods with different feature sets on BABEL test set

Regarding the length *k* of the input sequence, we observed modest impact on performance with *k=4* being a local maximum for most of the tested PP densities. Further augmenting the length of the sequence did not lead to significant improvement; hence it is worth to keep *k* as small as possible to favour a lightweight model.

5.3 Switching to ASR Transcripts

The realistic use-case for automatic punctuation is punctuating ASR output. Therefore, we evaluate our system on text converted from speech. Such text transcripts (ASR transcript) may contain ASR errors – word substitutions, word insertions or word deletions – and lack any punctuation mark and often also capital letters at sentence onsets. Due to word errors, we can expect a performance decrease of the punctuation when compared to the baseline used on error free text (falign – force aligned on true transcripts to obtain word slots). Results are presented in Fig. 3 and the respective rows of Table 3 for BABEL, where WER is 7.5%.

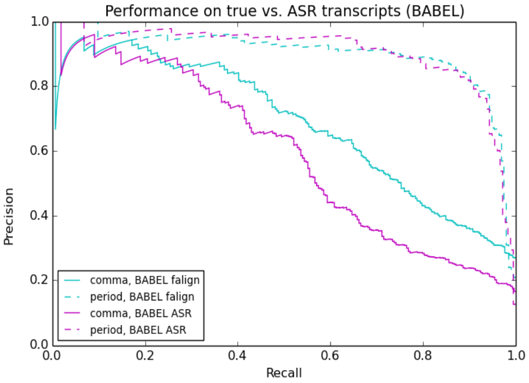

Figure 3

Precision and recall for commas and periods with true error free transcripts (falign) and ASR transcripts (ASR) on BABEL test set

In case of ASR transcripts, word recognition errors may propagate further into the processing pipeline. Periods seem to be resistant to these, whereas we can observe a modest performance drop for commas, which we explain partly by the propagated errors originated in the ASR.

5.4 Switching to Broadcast News

As we saw, punctuation results for commas dropped when using ASR transcripts (Fig. 3). When using BN data, where we have only ASR transcripts available obtained by WER=10.5%, this gap gets significantly larger: the curves for comma show lower precision and recall. Observing speech characteristics shows us that speaking style in the BN corpus is different to the BABEL one. We observe that BN utterances have consistently less characteristic acoustic-prosodic marking of comma slots. Therefore, we attempt an adaptation of the punctuation model by transferring parameters trained on BABEL and run 10 epochs on a held out set from BN data (25 blocks). Given the lightweight network, we let all parameters to learn. Fig. 4 shows results before and after this adaptation (validated and tested on the remaining 10+15 blocks). We notice a modest improvement only in period precision (for commas, only the operation point is shifted by closely the same F1). We think that signal level acoustic mismatch between BABEL and BN influences less the performance of the RNN punctuation model than does the speaking style: we suppose that the poorer comma recovery in the BN case is caused by the speaking style, i.e. acoustic-prosodic marking of comma slots is less characteristic. In the lack of these, the only way to restore punctuation is to use a TBP method.

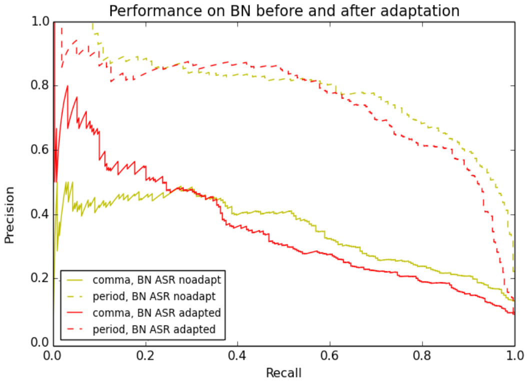

Figure 4

Precision and recall for commas and periods for ASR transcripts (noadapt) and for ASR transcripts (ASR) on BN test set

6 Subjective Assessment of Punctuation

It is an interesting question how the users themselves perceive punctuation accuracy and quality. All the measures used so far are objective measures and these are computed from comparing the automatic punctuation to the reference one. Although rarely, but it may happen that the same utterance has several correct punctuation patterns, such as in the well-known letter of the archbishop of Esztergom, John from Merania. His sentence, written in latin, "*Reginam occidere nolite timere bonum est si omnes consentiunt ego non contradico",* has two opposite interpretations based on where commas are inserted. Moreover, using punctuations when writing is a less conscious process than correct spelling of words, especially humans will not always agree, where to put commas into an unpunctuated text. We hypothesize that some eventual punctuation errors are even not spotted by the user.

Taking as an example the closed captioning of live video or audio with ASR, from a user perception point-of-view, subtitles are visible for some seconds, whereas the user concentrates on getting the meaning and following the video as well. In other words, it is more important, what is written, than how it is written. In addition, we may suppose that an unconscious error repair mechanism [21] is functioning, which, just like in self repairing coding, restores the correct punctuation sequence or ignores the errors in it, as far as error ratios are below a critical threshold.

Although we cannot carry out a throughout testing to determine this threshold for punctuation, [22] found a similar behaviour in human perception of audio, where phone errors were inserted in a gradually ascending manner. Within the present work, we undertake a comparison of automatically punctuated texts versus error free reference punctuated texts. A similar comparison is run for reference transcripts versus ASR transcripts, in order to compare the effect on human perception of ASR and punctuation errors. We use the Mean Opinion Score (MOS) metric, which we compute as the average of user ratings.

To carry out the subjective tests, we select 4 samples, composed of 5-7 coherent sentences from the BN corpus, and prepare 3 types of text for each: (i) a reference transcript with automatic punctuation (AP), (ii) an ASR transcript with reference punctuation (AT), and (iii) reference text with reference punctuation as a control set (CTRL). Users are asked to rate the text on a scale from 1 to 5 according to the following guideline: "*In the following text word or punctuation errors may appear. To what extent do these errors influence your ease of understanding? "*. During the evaluation, we contrast AP with CTRL and AT with CTRL. WER of the AT is 5.5%, SER of the AP is 6.4% in the selected blocks overall.

35 subjects, 28 male and 7 female with 29,6 years mean age took part in the tests, assessing two types of text out of the three possible. Most of them were university students or tercier sector employees. The subjects got the texts on a sheet and they had to read through once the 2 short blocks. One of the blocks tested for word errors, the other one for punctuation errors. The users were unaware of whether they receive a correct (reference or 100% accurate automatic) text or an incorrect text with eventual errors. They had to rate the texts according to how disturbing the errors were regarding the interpretation of the meaning (with score 5 = not disturbing at all to score 1 = text not understandable due to the errors).

Table 4

Mean Opinion Score and chi-square test results

Text set	**MOS**	**chi-square**	**p (significance)**
AP	4.28	14.0497	.00089
AT	4.05	5.1826	.07492
CTRL	4.19	N.A.	N.A.

Results are summarized in Table 4. On the ratings we calculated MOS and performed a chi-square test to see whether differences are statistically significant. Surprisingly, MOS for AP is higher than for the control blocks, but it is more important that even by 1% significance level (*p<.01*), subjects were not able to make a difference between correct and erroneous texts in terms of punctuation. Spotting ASR errors is easier, regarding the AT vs. CTRL task we found a statistically significant difference in ratings by 5% significance level (*p>.05*).

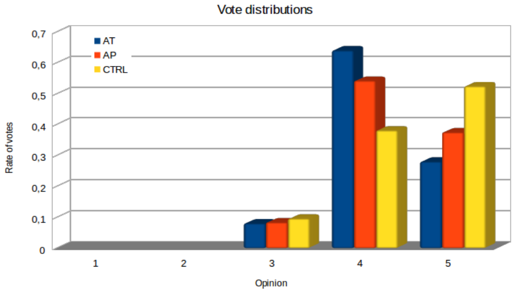

Figure 5
Distribution of subjective ratings for AT, AP and CTRL

Fig. 5 shows the distribution of votes. It is a bit surprising to observe that control and hence error free texts are evaluated as score "5" in only about 50% of the cases. We explain this by two factors: (i) subjects knew that they had to rate possibly erroneous text samples, which made them more "suspicious" and biased the rating; (ii) humans are not 100% accurate in correct spelling and correct punctuation and they are not able to spot all of the errors, hence they favour the rate "4", "*acceptable with minor errors*".

Overall, results confirm our initial hypotheses: if punctuation error is low, humans are not able to locate punctuation errors, whereas they are more sensitive to ASR errors than to punctuation errors. Although we did not assess MOS for texts without any punctuation, based on our experience we regarded these as hard to follow and understand if provided on a word-by-word basis in sequence.

Conclusions

In this paper we presented a novel prosody based automatic punctuation approach and evaluated it in realistic use-case scenarios. The model relies on phrase sequence information, exploited in a recurrent neural network framework. The model is implemented such that it has minimal latency and resource demand, in order to allow for real-time exploitation. Additionally, we performed subjective tests to assess whether errors affect readability and text understanding in texts with automatic punctuation. Results showed that humans are less able to spot punctuation errors and they are less sensitive to these kinds of errors than to ASR errors; hence, a "good-enough" punctuation may be sufficient is several cases when ASR is used for speech to text conversion.

Acknowledgements

This work was supported by the National Research, Development and Innovation Office of Hungary under contracts PD-112598 and FK-124413. The experiments were run on NVIDIA Titan GPU provided by NVIDIA.

References

[1] P. Baranyi and A. Csapo, “Cognitive infocommunications: CogInfoCom,” in Computational Intelligence and Informatics (CINTI), 2010 11th International Symposium on. IEEE, 2010, pp. 141-146

[2] P. Baranyi, A. Csapo, and G. Sallai, “Cognitive Infocommunications (CogInfoCom)” Springer, 2015

[3] F. Batista, H. Moniz, I. Trancoso, and N. Mamede, “Bilingual experiments on automatic recovery of capitalization and punctuation of automatic speech transcripts,” Transactions on Audio, Speech and Language Processing, 20(2), pp. 474-485, 2012

[4] D. Beeferman, A. Berger, and J. Lafferty, “Cyberpunc: A lightweight punctuation annotation system for speech,” in Proc. International Conference on Acoustics, Speech and Signal Processing, IEEE, Vol. 2. 1998, pp. 689-692

[5] C. J. Chen, “Speech recognition with automatic punctuation,” in Proceedings of Eurospeech, 1999

[6] 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

[7] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014

[8] G. Kiss, D. Sztahó, and K. Vicsi, “Language independent automatic speech segmentation into phoneme-like units on the base of acoustic distinctive features,” in Proc. Cognitive Infocommunications (CogInfoCom), IEEE, 2013, pp. 579-582

[9] W. J. Levelt, “Monitoring and self-repair in speech”. Cognition Vol. 14, pp. 41-104

[10] 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, 2010, pp. 177-186

[11] J. Makhoul, F. Kubala, R. Schwartz, and R. Weischedel, “Performance measures for information extraction,” in Proceedings of DARPA broadcast news workshop, 1999, pp. 249-252

[12] S. Millotte, R. Wales, and A. Christophe, “Phrasal prosody disambiguates syntax,” Language and cognitive processes, 22(6), pp. 898-909, 2007

[13] A. Moró and G. Szaszák, “A prosody inspired RNN approach for punctuation of machine produced speech transcripts to improve human readability”, Cognitive Infocommunications (CogInfoCom) IEEE, 2017

[14] A. Postma, "Detection of errors during speech production: a review of speech monitoring models," Cognition, 77, pp. 97-131, 2000

[15] P. S. Roach et al., "BABEL: An Eastern European multi-language database," in International Conf. on Speech and Language, 1996, pp. 1033-1036

[16] E. Selkirk, "The syntax-phonology interface," in International Encyclopaedia of the Social and Behavioural Sciences. Oxford: Pergamon, 2001, pp. 15407-15412

[17] M. Schuster and K. Paliwal, "Bidirectional recurrent neural networks," IEEE Trans. on Signal Processing, Vol. 45, No. 11, pp. 2673-2681, 1997

[18] 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 ITRW on Prosody in Speech Recognition and Understanding, 2001

[19] G. Szaszák and A. Beke, "Exploiting prosody for automatic syntactic phrase boundary detection in speech," Journal of Language Modeling, 0(1), pp. 143-172, 2012

[20] G. Szaszák, K. Nagy, and A. Beke, "Analysing the correspondence between automatic prosodic segmentation and syntactic structure." in Proc. Interspeech, 2011, pp. 1057-1060

[21] A. S. Szőllősy and G. Vitályos, "Pragmatics in the usability discipline," in Cognitive Infocommunications (CogInfoCom) IEEE, 2012, pp. 359-364

[22] L. Tóth, "Benchmarking Human Performance on the Acoustic and Linguistic Subtasks of ASR Systems". Proc. Interspeech 2007, Antwerp, Belgium, pp. 382-85, 2007

[23] A. Varga et al., "Automatic closed captioning for live Hungarian television broadcast speech: A fast and resource-efficient approach," in International Conference on Speech and Computer. Springer, 2015, pp. 105-112

[24] K. Vicsi and G. Szaszák, "Using prosody to improve automatic speech recognition," Speech Communication, 52(5), pp. 413-426, 2010

[25] J. Žibert, et al., "The COST 278 broadcast news segmentation and speaker clustering evaluation-overview, methodology, systems, results," in 9th European Conference on Speech Communication and Technology, 2005