Using Suprasegmental Information in Recognized Speech Punctuation Completion

Marek Boháč and Karel Blavka

Institute of Information Technology and Electronics,
Studentská 2/1402, 461 17 Liberec, Czech Republic
{marek.bohac,karel.blavka}@tul.cz
https://www.ite.tul.cz/itee/

Abstract. We propose a scheme to determine punctuation of the text produced by an automatic speech recognizer. We deal with the addition of commas based on the recognized text and we propose a full stop detection scheme using both – the textual and prosody information. We also propose an expanded scheme which utilizes enriched audio document information (e.g. speaker diarization, language detection etc.) to improve the sentence boundary detection. We compare the above mentioned schemes and its accuracy in terms of (in)correctly estimated punctuation markers and its ability to mark the positions of sentence boundaries. Hence we want to show it is better to incorporate all the relevant information sources in one reasonable scheme than to split the document processing into independent layers. Proposed schemes are evaluated over a set of recordings from the Czech (and Czechoslovak) radio broadcasts.

Keywords: punctuation completion, fundamental frequency detection, comma, full stop, automatic speech recognition, document segmentation.

1 Introduction

An increasing number of automatic speech recognition (ASR) applications aims to provide the access to different media sources and make it searchable. Some typical examples are on-line media monitoring systems [1], audio archive indexing engines [2] or lecture streaming [3]. Common feature of all mentioned applications is an interface presenting the ASR textual output to the user. When the presented plain text fragment is longer than one sentence, reading becomes very demanding and uncomfortable. As some languages (e.g. Slavic languages) have very loose form – especially their spoken form – it becomes essential to complement the ASR results with appropriate post-processing and punctuation.

We propose an approach to supplement the recognized text with punctuation markers and we show the importance of utilizing more information sources than to perform each post-processing level individually. From the whole post-processing task we focus on the modules estimating the commas and full stops. The comma estimation is based on N-gram language modeling. The full stop estimation employs the recognized text, prosodic information (speech fundamental frequency and non-speech events) and the document segmentation. We evaluate the consequences of employing or not employing concrete information sources.

P. Sojka et al. (Eds.): TSD 2014, LNAI 8655, pp. 555–562, 2014.
This is preprint prepared by Proceedings editor for Springer International Publishing Switzerland.

The next section introduces all the employed modules with emphasis on the full stop determination. Section 3 describes the experimental setup and results. In Section 4 we make conclusions and propose the future work.

2 Proposed Scheme

As can be seen in Fig. 1 the audio document processing can be generalized into four functional blocks: Automatic Speech Recognition system (ASR), Document Segmentation (Doc Seg), Comma completion and Full Stop completion. The complexity of concrete blocks may highly differ accordingly to the structure of the audio document (e.g. single vs. multiple-language documents) and to the demands of the user (e.g. one-pass vs. multi-pass ASR). In the following paragraphs we introduce the employed functional blocks with emphasis on the Full Stop completion as the other blocks were already published.

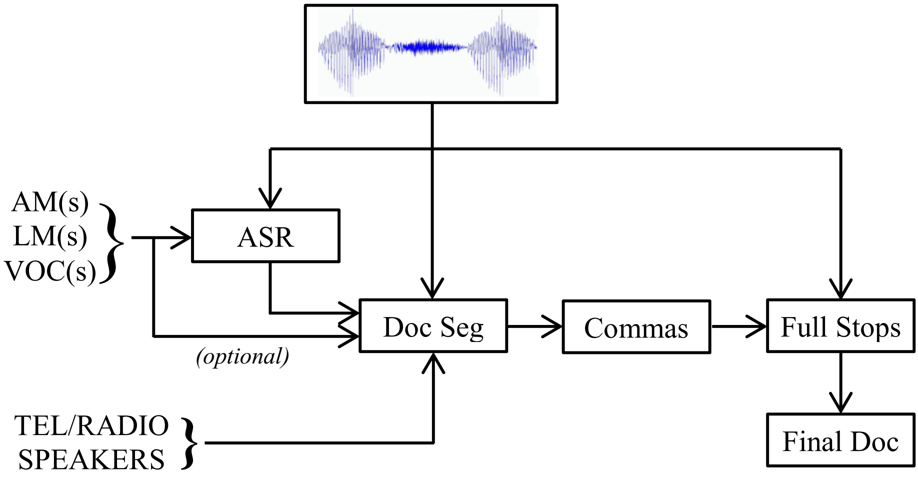

Fig. 1. Overall scheme of the audio document processing

2.1 ASR

The automatic speech recognizer (ASR) is our own one-pass time-synchronous Viterbi decoder. The HMM-based acoustic model contains speech phonemes as well as non-speech events (e.g. breathing, hesitation, click and cough). The features are 13-dimensional MFCCs with first and second derivate. The input audio presumes at least 16 kHz sampling frequency so the FFmpeg can convert it into standard PCM Wave format (mono, 16 bits per sample).

The acoustic models (AM), language models (LM) and vocabularies (VOC) are trained over different sets of data. The Czech vocabulary contains 550k items and the AM is trained using 300 h of recordings. The Slovak vocabulary contains 320k items, its AM is trained on 100 h and the combined Czech-Slovak vocabulary contains 50k+50k items and the AM is trained on 100 h+100 h of recordings. The language models use different amounts of stored (precomputed) word-pairs that can be mixed and weighted according to the application topic domain. In the case of Czech-Slovak LM there are special features preferring to keep the current language over changing it.

2.2 Document Segmentation

The document segmentation module (Doc Seg) employs several modules – some of them optional. The base and mandatory module is the speaker diarization. The optional modules are the channel classification (e.g. telephone vs. radio speech), the speaker identification and the jingle/song detection. All these partial information sources are passed to a logical layer which outputs smoothed language, speaker and channel homogeneous document segments (paragraphs).

Speaker Diarization The diarization module includes usual layers: i) voice activity detection (VAD), ii) speaker turn detection and iii) speaker clustering.

The VAD module is substituted by the usage of ASR output which already exists. This brings few benefits. In the case of noisy recordings the ASR-based speech detection grants higher robustness than a standalone solution. We can also limit the speaker turn points only to the borders of words so it decreases the computational demands and synchronizes the ASR output and the diarization.

Speaker turn point detection is performed by a variable-length window sliding along the frames of the parameterized recording. The test (derived from the Bayesian Information Criterion - BIC) compares the speech-labeled frames on the left and on the right from the examined turn point (for details see [4]). If the test exceeds the given threshold the speaker turn point is marked (we prefer over-splitting as it can be corrected in the clustering phase).

The speaker clustering is performed in two hierarchical steps. Firstly the segments are pre-clustered by the BIC-based classifier [5], secondly i-vector [6] representation of clusters with cosine similarity measure finalizes the clustering.

As the audio document diarization is a complex task and detailed description was already published, we kindly recommend to see [4,5,6] for details.

Speaker Identification & Channel and Gender Classification The tools performing the document diarization were adjusted so the document segments can be classified with regards to different criteria. If the document contains utterances of already known speakers we can train i-vector based speaker models and identify the speaker. The score returned by the tool can be used to find a threshold so we can verify the speaker (we can detect if the speaker was not observed before). Second tool is a GMM-based classifier used to distinguish between standard recordings and narrow-band recordings (typically telephone transmissions). The GMM models are also used to obtain the speaker's gender.

Language Identification & Document Segmentation Smoothing If there can be more than one language present in the recording (e.g. historical Czech and Slovak news) we apply special combined LM, AM and VOC in the first-pass recognition. For every speaker-and-channel homogenous segment the dominant language is determined (by the word count of the languages). If we can identify concrete speakers (not only diarization clusters), the language decision is smoothed over all speaker's utterances in the document.

The document segmentation is enhanced by the detection of jingles and strongly noised segments (e.g. music, terrain recordings). It is found by a sliding window traversing through the ASR transcription. Segments we want to detect have very high share of certain non-speech events (for details see [7]). Finally we smooth the document clustering with regards to the speakers' ID, gender, channel and language but also preventing over-segmentation. Segments are optionally recognized in the second pass using appropriate LM, AM and VOC [4].

2.3 Comma Completion

Our comma completion module is based on the textual input only. For the training we used a hand-made corpora (109 MB of Czech texts and 130 MB of Slovak texts) containing both – spontaneous (interviews, talk shows) and prepared speech (news broadcasts, public speeches). We searched the corpora for the most frequent words/phrases preceding and following the occurrence of commas. Then we carried out a statistical analysis of these rules so we defined the rules for Czech and Slovak comma completion (some of them produce the comma while others forbid it). These rules are word N-grams where $N \leq 3$ (e.g. prepositions, conjunctions and some common phrases). The application of the rules is performed via Weighted Finite State Transducers – WFST. One of the biggest advantages is the WFST ability to determine N-best solutions so it solves the cases of overlapping rules. The Czech rules are 1,243 phrases after comma (with 1,883 negatives) and 130 rules before comma (with 518 negatives). The Slovak rules are 2,518 rules after comma (with 5,071 negatives) and 333 rules before comma (with 5,752). The negatives prevent false commas by longer rule restricting the shorter one (which generates the comma).

2.4 Full Stop Determination

In the following paragraphs, we propose a prosody-based scheme for the full stop determination. It employs the ASR output to localize words and non-speech events. Speech melody (F0) is estimated using STFT and a dynamic programming-based decoder, choosing the most likely F0, that is searched for the potential full stop points. We also show how the ASR output and the diarization can be utilized to improve the full stop (and sentence boundary) determination.

Speech Localization & STFT To localize the audio segments containing speech we use the ASR output. As it provides the time stamps for all the words and non-speech events in the recording, the detection of speech segments is a straightforward task. The recording is processed word by word – inside every time span we compute the Short Time Fourier Transform (STFT). The STFT is computed within 20ms frames, 10ms overlap, zero-padded to 4096 samples and windowed with the Hamming window. From every frame we choose 5 most significant components (magnitude local maximums) in range 60–600 Hz (this interval should cover all – children, male and female speakers as shown in [8]). The detected components are given weights according to their significance (in our case 10, 9, 7, 5, 5) so we obtain a spectrogram with a kind of histogram equalization which is passed to the prosody decoder.

Fundamental Frequency Decoding The F0 decoder solves few tasks at once. It chooses the best fitting F0 according to the spectrogram and it performs a kind of F0 smoothing also. The algorithm is based on dynamic programming and optimizes a path between averaged three last frames and three first frames. Between frames inside the word borders it can pass directly between the significant components or can keep the component from the preceding frame. Passing between the significant components is favored by its weights (so the most significant components will most probably form the prosody). Big steps between the components are penalized (even a trained singer has very limited speed of prosody change) as well as the keeping of preceding component has its penalization. The best path through the spectrogram matrix is the detected prosody.

Prosody-based Sentence Boundary Determination As the Czech and Slovak prosody is not very distinctive (when compared for example with English or French native speakers) we can extract weaker cues to detect the sentence borders. Generally, we observed that pitch declines as the sentence end approaches and the new sentence starts at a higher pitch. The last word of the sentence has usually very changing pitch while the words inside usually keep flatter pitch trend. We decided to detect this behavior by two features – the mean pitch of the word $\overline{P}$ and a normalized difference between maximum and minimum pitch P of the word as shown in (1).

The sentence boundaries are proposed as subsequent word pairs where the mean pitch declines and the normalized pitch difference of the second word exceeds a given threshold. Such a word pair must be followed by a word with a higher mean pitch or by a non-speech event (e.g. breath, laughing).

$$NormDiff = \frac{\max(P) - \min(P)}{\overline{P}} \tag{1}$$

ASR and Segmentation Utilization To decide if the proposed sentence boundary induces a full stop we firstly check if there is a comma. Secondly we check the ASR output around the proposed sentence boundary – some words do not occur at the sentence end/begin (we found these words statistically in the previously mentioned corpora). If the previous fulfilled we place the full stop.

As the recordings, we process suffer from background music and noises, the prosody information is not as reliable as for clear recordings. Hence we prefer a setup with high precision and lower recall. Thus we need to detect more full stops using other information sources. Segmentation information is a natural choice. We place full stops at the ends of document segments – paragraphs.

The last source of full stops is the ASR output. We detect non-speech events within long sequences without a sentence delimiter. We place a full stop to every non-speech event which constitutes a sentence longer than 12 words (as it is the average lengths of Czech sentences – see [7]).

3 Experimental Evaluation

In this section we describe the evaluation data and define the metrics. As we want to show the impact of using different information sources, we evaluate the punctuation scheme in five setups. All the schemes have the same comma detection module but they differ in the full stop determination stage: i) using the prosody and ASR output (fs_pros), ii) using prosody, ASR and speaker-turn points (fs_turns), iii) using all – the prosody, ASR, speaker-turn point and heuristics splitting of too long sentences (fs_full), iv) using only the speaker-turn points (fs_trn) and v) using speaker-turn points and heuristic splitting (fs_trhe).

3.1 Evaluation Data and Metrics

The evaluation data consists of 21 radio broadcasts recorded within years 1971 – 2005. Total duration of the recordings is 9 hours 21 minutes. In 10 of them Czech and Slovak occurs (Slovak forms 10%–40% of the concrete recordings). Some parts are recorded outside (e.g. telephone entry, street interview, etc.) so there are strongly noised parts as well as jingles and background music.

The reference transcripts were made manually. As there can be some differences between the reference and ASR-recognized text, we aligned it using the word time stamps obtained via ASR and forced alignment of the references [9]. However there are still two minor drawbacks of this approach. First one is possible misalignment between the reference and the ASR. We manually checked the data and the error is negligible. The second is that different annotators usually mark the same positions as sentence boundaries but the inter-annotator agreement on concrete punctuation marker is low – see [7]. As every document was transcribed by one annotator we must consider it the ground truth.

We use these evaluation metrics: accuracy (2), precision (3), recall (4), detection rate (5) and false alarm rate (6), where *TP* stands for true positives (correctly marked positions), *FP* stands for false positives (false alarms), *FN* stands for false negatives (missing markers) and *CF* stands for confused markers (e.g. annotator makes a full stop and the system generates comma). Subscript *com* denotes commas and *fs* stands for full stops.

$$ACC = \frac{TP}{TP + FP + FN} \tag{2}$$

$$PRC = \frac{TP}{TP + FP} \tag{3}$$

$$REC = \frac{TP}{TP + FN} \tag{4}$$

$$DR = \frac{TP_{fs} + TP_{com} + CF}{TP_{fs} + FN_{fs} + TP_{com} + FN_{com}} \tag{5}$$

$$FA = \frac{FP_{fs} + FP_{com}}{TP_{fs} + FN_{fs} + TP_{com} + FN_{com}} \tag{6}$$

3.2 Experimental Results

As the commas and full stops occupy the same set of positions we evaluate the experiments together – see Table 1. This is clearly shown by the results of comma detection. Results of the same module differ as *CF* is not the same (there are full stops instead of missing commas).

The comma completion results show over 80% precision (low false alarm rate). The problem is with lower recall (we can mark approx. 50% of the positions). Similar is the situation with the prosody based full stop detection. It has very good precision (over 85%) but low recall (approx. 25%). If we presume the application of "full scheme", this is what we need – place the markers where we are sure and pass those which can be placed by other knowledge sources.

The results of employing the document segmentation are predictable – it improves the full stop determination. The decrease of precision is caused by over-segmentation of the document (some longer paragraphs are interrupted). The additional "heuristic" completion of commas naturally carries decrease of precision but the impact on the recall and sentence boundary detection far outweights it. This can be clearly seen in the increase of sentence boundaries detection rate. Our experience also proves that slightly over-segmented text is more reader-friendly than a text with very long sentences.

Table 1. Experimental results – full stops, commas and sentence boundaries

	Full Stops			Commas			Sentence Boundaries	
scheme	ACC [%]	PRC [%]	REC [%]	ACC [%]	PRC [%]	REC [%]	DR [%]	FA [%]
fs_pros	23.32	86.85	24.18	43.78	83.12	48.06	40.90	7.03
fs_turns	29.65	81.42	31.81	44.90	83.18	49.40	44.94	8.74
fs_full	48.53	61.02	70.35	50.86	83.18	56.70	74.47	27.95
fs_trn	16.88	78.41	17.70	43.71	83.28	47.92	36.50	7.35
fs_trhe	43.93	59.21	62.98	49.80	83.28	55.14	70.06	27.68

4 Conclusions & Future Work

We presented a scheme for comma and full stop completion. Our results show the advantage of combining all the available knowledge (text, prosody, segmentation) against separated independent layers. Our punctuation scheme is sufficient to make the document easily readable although there are some reserves.

A closer view showed us that to mark the missing comma positions we would have to carry out the semantic analysis of the text. Another future work is to redefine the stop-lists (words implying that the sentence continues) using more linguistic knowledge (not only the statistics). Our main future interest lies in better definition of heuristics for additional full stop placing – especially in utilizing verb detection in the text (and so preventing the false alarms).

Acknowledgments. This work was supported by Czech Ministry of Culture (project DF11P01OVV013 in program NAKI).

References

1. Pawlaczyk, L., Bosky, P.: Skrybot – a system for automatic speech recognition of polish language. In: Man-Machine Interactions. Volume 59. Springer Berlin Heidelberg (2009) 381–387
2. Ordelman, R., de Jong, F., Huijbregts, M., van Leeuwen, D.: Robust audio indexing for dutch spoken-word collections. In: XVIth International Conference of the Association for History and Computing, Amsterdam, KNAW (2005) 215–223
3. Cerva, P., Silovsky, J., Zdansky, J., Nouza, J., Malek, J.: Real-time lecture transcription using asr for czech hearing impaired or deaf students. In: 13th Annual Conference of the International-Speech-Communication-Association, Portland, ISCA (2012) 762–765
4. Cerva, P., Silovsky, J., Zdansky, J., Nouza, J., Seps, L.: Speaker-adaptive speech recognition using speaker diarization for improved transcription of large spoken archives. Speech Communication **55** (2013) 1033–1046
5. Chen, S., Gopalakrishnan, P.: Speaker, environment and channel change detection and clustering via the bayesian information criterion. In: Proc. DARPA Broadcast News Transcription and Understanding Workshop, Virginia, USA (1998) 127–132
6. Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., Ouellet, P.: Front-end factor analysis for speaker verification. Audio, Speech, and Language Processing, IEEE Transactions on **19** (2011) 788–798
7. Bohac, M., Blavka, K., Kucharova, M., Skodova, S.: Post-processing of the recognized speech for web presentation of large audio archive. In: Telecommunications and Signal Processing, 2012 35th International Conference on. (2012) 441–445
8. Atassi, H.: Metody detekce základního tónu řeči - methods for speech pitch detection. Elektrorevue (2008)
9. Bohac, M., Blavka, K.: Text-to-speech alignment for imperfect transcriptions. In Habernal, I., Matoušek, V., eds.: Text, Speech, and Dialogue. Volume 8082 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2013) 536–543