INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

A phonological phrase sequence modelling approach for resource efficient and robust real-time punctuation recovery

Anna Moró[1], *György Szaszák*[1,2]

[1]Department of Telecommunications and Media Informatics,
Budapest University of Technology and Economics, Budapest, Hungary
[2]Spoken Language Systems Group (LSV), Saarland University, Saarbrücken, Germany
moro@tmit.bme.hu, gszaszak@lsv.uni-saarland.de

Abstract

For the automatic punctuation of Automatic Speech Recognition (ASR) output, both prosodic and text based features are used, often in combination. Pure prosody based approaches usually have low computation needs, introduce little latency (delay) and they are also more robust to ASR errors. Text based approaches usually yield better performance, they are however resource demanding (both regarding their training and computational needs), often introduce high time latency and are more sensitive to ASR errors. The present paper proposes a lightweight prosody based punctuation approach following a new paradigm: we argue in favour of an all-inclusive modelling of speech prosody instead of just relying on distinct acoustic markers: first, the entire phonological phrase structure is reconstructed, then its close correlation with punctuations is exploited in a sequence modelling approach with recurrent neural networks. With this tiny and easy to implement model we reach performance in Hungarian punctuation comparable to large, text based models for other languages by keeping resource requirements minimal and suitable for real-time operation with low latency.

Index Terms: speech prosody, punctuation, speech recognition, recurrent neural networks, bidirectional LSTM

1. Introduction

Punctuation recovery for Automatic Speech Recognition (ASR) output is a known problem in speech technology. The importance of having punctuations in automatically generated text has been outlined many times [1], [2], as this is important both for human readibility, but also for subsequent processing with text based tools. Two basic approaches are used for automatic punctuation, mostly in combination: (i) Prosody Based approaches (hereafter PB) and (ii) Text or lexical Based approaches (hereafter TB).

PB approaches exploit acoustic markers and attempt to link these to required punctuations in the text. Features derived from fundamental frequency (F0), energy and durations have been successfully used for automatic punctuation. In [3], the authors used pause and phoneme duration with F0 statistics in a finite state model and also in a Multi Layer Perceptron (MLP) approach for punctuation. In [4] Shriberg et al. report 82.4 % overall accuracy by 23.6% efficiency in a punctuation subtask of topic segmentation with disallowed lexical and promoted prosodic features in a decision tree approach. In [2], Batista et al. use word and syllable based statistics of prosodic features with a maximum entropy based classifier and report an overall impact of 69% of prosody in a punctuation task for English broadcast news ASR transcripts with prosodic and lexical features. A very often used acoustic-prosodic feature in hybrid PB-TB punctuation is pause duration, as it is easily available from the ASR step and especially as it shows high correlation with punctuation marks [3]. Statistics computed from F0, especially slopes come into the second place as they infer with intonation, crucial in identifying sentence ends, continuation rise (often followed by a comma) and interrogative modality [2].

Early TB approaches proposed to add punctuation marks to the N-gram language model of the ASR as hidden events [4], [5]. More sophisticated sequence modelling approaches were also inspired by this idea: a transducer alike approach getting a non punctuated text as input is capable of predicting punctuation as was presented in numerous works [2], [6], [7], with frameworks built on top of Hidden Markov Models (HMM), maximum entropy models or conditional random fields etc. Applying a monolingual translation paradigm for this sequence modelling task was proposed in [8]. Recently, sequence-to-sequence modelling deep neural network based solutions have been also presented: taking a large word-context and projecting the words via an embedding layer into a bidirectional Recurrent Neural Network (RNN), high quality punctuation could be achieved in English and Estonian [9].

TB approaches require large training corpora, especially if they benefit from word-embeddings, and are computationally more expensive than pure PB approaches. For agglutinating languages, such as Hungarian, TB approaches suffer from the exploding dictionary as well. TB approaches may introduce a latency which is not tolerable in real-time modes. As ASR also needs much computation and may have large memory requirements for decoding, computational load may also become crucial. In this paper, we intend to propose a lightweight and pure PB approach, which first captures the information structure of the utterance relying on speech prosody. We may observe that although the feature set is often called 'prosodic' in many punctuation applications, supra-segmental processing in the strict sense is rarely addressed; rather, static features are extracted around word boundaries to decide for a slot whether a punctuation mark is necessary. Compared to this, the proposed approach treats an utterance as a prosodically coherent entity, and performs a phonological phrase alignment via a simple and fast Viterbi-decoding with a lightweight HMM model. This step is supposed to capture a part of the information structure as far as reflected by prosody. By relying on the phrase sequence and timing, and also exploiting pause duration, a tiny RNN architecture is proposed to predict punctuation marks.

The paper first presents the phonological phrasing approach, then introduces the RNN model built to predict punctuation based on the phrase sequence. Results obtained for punctuation in Hungarian are discussed and, finally, concluded.

http://dx.doi.org/10.21437/Interspeech.2017-204

2. Phonological Phrasing

A phonological phrase (PP) is defined as a prosodic unit, characterized by an own and singe stress and some following intonation contour [10]. In the prosodic hierarchy, PPs are placed between the better known intonational phrases and prosodic words. The strength and the place of the stress within the PP, as well as its intonational contour may vary, depending on higher (utterance or IP) level prosodic and additional syntactic or semantic/pragmatic constraints. We interpret PPs as linked to the word sequence, i.e. a PP always begins and ends at word boundaries and hence, can correspond to a single word or a group of words in the utterance [11]. PPs reflect the information structure [12] by providing a layering of the conveyed information, per se limited by the syntax/phonology interface, as prosodic and syntactic structure are closely related, but not identical [13].

In [14], a HMM approach was proposed, further enhanced by [11], 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 [11]). Hungarian has fixed stress on the first syllable.

Table 1: *The modelled phonological phrase (PP) types for Hungarian with their associated stress, intonational contour characteristics and location within the IP.*

Label	Stress	Location	Intonation pattern
io	strong	IP initial	IP onset + descending
ss	strong	IP internal	Stress + descending
ms	medium	IP internal	Stress + descending
ie	medium	IP terminal	Stress + low ending
cr	medium	IP terminal	Stress + ascending
ls	neutral	IP initial	No stress + desc.
sil	neutral	Betw. IPs	Silence

The HMM PP models use prosodic features – continuous F0 and energy streams, with added deltas calculated with several different time spans to reflect short and long term tendencies of the features. PP entities are treated in a very similar way to the acoustic models used in conventional ASR systems. Also, the Viterbi alignment which yields the PP alignment for the utterances is similar to the one known in ASR, but here, the recognition network is very simple, consisting of a simple loop over the 7 possible PP classes. Given the acoustic models of the PPs are also very simple (11-state left-to-right models with only up to 4 Gaussian mixtures for the 7 PP classes), automatic phonological phrasing has very low resource requirements. Just like in an ASR system, backtracking is possible at intermittent points if a longer continuous speech stream is processed. Details of the approach, including acoustic feature extraction, training data, parameter settings and exhaustive evaluation for automatic phrasing, stress detection and word-boundary detection were presented in [11], hence the reader is referred to [11] and [14] for more information. Here we briefly mention that precision and recall of phrase boundaries was 0.89 for Hungarian on a read speech corpus (for the operation point characterized by equal precision and recall).

As PPs are defined such that their boundaries coincide with word boundaries, and by taking into account their location and role within the IP (IP initial, IP terminal with descending contour or with continuation rise, IP internal with high/medium/little prosodic stress etc.), we observe and hypothesize, that the automatically aligned PP sequence reflects to some extent the information structure [12] and hence is expected to show a high correlation with punctuation marks required in the corresponding transcripts of the utterance. We intend to exploit this in automatic punctuation as explained in the subsequent sections.

We consider a novelty and a strength of the PP alignment approach that it treats an utterance as a meaningful entity in terms of its inherent prosodic structure and logic, and tries to interpret this on its own, i.e. without focusing only on slots around hypothesized word boundaries. In other words, the approach allows for a temporally contrastive (in the supra-segmental sense) and coherent processing of speech prosody.

3. Phrase Sequence Modelling and Punctuation

3.1. Phrase Sequence Features

In the first step toward automatic punctuation, the automatic PP alignment is performed as described in Section 2. The ASR output is put in parallel to the automatic PP alignment in order to have word hypothesis information (where the words begin and end, and whether there is a pause in between them). Based on these alignments (word and PP hypotheses), the following set of features is extracted: (1) the label of the PP (PP_{label}, out of the 7 possible classes); (2) the duration of the PP (PP_{dur}, in milliseconds); (3) the pause duration from the ASR, where the PP ends (SIL_{dur}, in milliseconds). The pause duration is set to zero if there is no word ending in the $\pm 250ms$ vicinity of the hypothesized end of the PP. If there are multiple pauses in this interval, the closest to PP ending wins.

Please note, that the PP alignment is not required and is not able to recover short pauses, as a 250 ms minimum constraint is applied in it to prevent misclassifications resulting from unvoiced, low energy speech segments. This is the rationale of including SIL_{dur} features from the ASR output.

Please note also some particularities of the proposed approach in contrast to standard punctuation approaches: usually, slots for punctuation are considered following each word in the text or transcript. We keep this definition for consistency with other works, but we assess slots for every PP. By definition, a PP starts and ends at word boundaries. Even supposing a 100% accurate automatic phrasing, not every word boundary is a PP boundary. Hence, we lose part of the potential slots for a punctuation mark, we hope these missed slots are mostly empty slots (with no required punctuation). In the evaluation section (in subsection 4.1), we present a methodology to evaluate this loss; and we also compensate for the differences in slot treatment and make our approach comparable to standard approaches assessing each slot following a word boundary. Of course, automatic phrasing is not perfect, errors may occur resulting in a PP boundary not coinciding with any word boundary. In this case, punctuation target is treated as empty both for training and testing of our punctuation system.

PP density can be controlled by an insertion log likelihood parameter. The more dense the PP alignment gets, the less latency results from waiting for the one additional PP after the target slot (see the RNN topology and the target slot of the output in subsection 3.2). Average PP duration is 270 ms by the densest alignment used in the experiments (this indeed may already be shorter than average word duration). We do not use alignments with density resulting in over 1 sec average PP duration.

3.2. RNN Phrase Sequence Model

The extracted features are regarded as a sequence, are windowed and fed into the RNN-based phrase sequence model. The window spans $W = 2..10$ PPs and their corresponding features. The single target slot (output of the RNN) for punctuation prediction is always the slot $W - 1$ one, preceding the last PP in the sequence. The window is shifted by one step once a new PP is available to the right. The window span is optimized later, hence it is left as a tunable parameter. PP_{label} features are one-hot encoded, PP_{dur} and SIL_{dur} are Z-normalized on utt basis prior to feeding them to the network.

Regarding punctuations, we assess periods and commas, as in the corpora available for the experiments, question marks and exclamation marks are heavily underrepresented. In order to prevent working with a severely imbalanced dataset, we map question and exclamation marks to periods. Semicolons and colons are also mapped to periods, leading citation quotes are removed, terminal ones are mapped to comma, together with dashes. For revealing questions based on prosody, [2] proposed an approach, in this paper we focus only on phrasing related comma and sentence terminal period (full stop) recovery.

The initial approach is to adopt a 2-layered recurrent neural network, its recurrent layers both consist of LSTM cells. During the development phase of the model, several different topologies and parameter-sets have been implemented. The window size and the number of feature steams used can be easily modified, they were tested alongside the different topologies. We started with unidirectional layers, but during training we observed better fitting with bidirectional layers, hence we moved on toward a BiLSTM. Despite targeting the $W - 1$ slot exclusively, a bidirectional approach using very limited future context slightly outperformed unidirectional models on the validation set. For a complete overview of bidirectional networks we refer the reader to [15].

The first layer of our final network is made out of 20 LSTM cells with inner sigmoid activations. We add a dropout layer with a dropout rate of 0.25, while the second BiLSTM layer is made out of 40 units with 0.3 dropout. The two recurrent layers are followed by a fully-connected layer, its activation function is defined as a softmax nonlinearity.

The network is trained using the Adam optimizer with a learning rate of 0.001 and by using adaptive estimates of lower-order moments [16]. 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.

For such a lightweight network, training is not time consuming; even on CPU the network can be trained within 3-5 minutes on a standard 8 core Intel(R) Core(TM) i5-6600K CPU @ 3.50GHz workstation.

3.3. Speech Corpora and ASR

Hungarian BABEL [17], a read speech database recorded from non-professional native speakers is used to train the RNN phrase sequence model. The corpus contains approx. 2K utterances, with 3K comma and 2K periods. There are almost 20K word slots and 7K - 20K PP slots, the latter depending on the density of the PP alignment controlled by the phrase insertion log likelihood parameter. A 10% subset of the training set is reserved for validation, and another 10% for testing.

We also test on a Broadcast News (BN) corpus [18] of 50 short blocks with 3K words, 300 comma and 500 period slots. This material is an ASR hypothesis obtained by WER=10.5%.

4. Results

4.1. Metrics

The commonly used approach in punctuation is to decide on punctuation for slots following each word in the text (word slots). In our approach we consider slots where PPs end (PP slots). As explained in section 3.3, thereby we may miss some word slots with punctuation and also insert slots which are not located between two words. In order to make our approach comparable to standard approaches which consider slots following each word, we adopt an evaluation scheme as follows: although our system is trained on and makes predictions for PP slots, we transform this to word slots by dropping all PP slots with unmatched word slot in time alignment (hereafter unmatched PP slots) and inserting a blank label prediction to non-covered word slots. During the normal operation of the punctuation system, only dropping of unmatched PP slots is necessary, inserting blank punctuations for uncovered word slots makes no sense, of course.

We use precision (Pr), recall (Rc), F1-measure (F1) and Slot Error Rate (SER) [19] as evaluation metrics, and present these numbers corrected for word slots. We also evaluate the Slot Miss Ratio (SMR), defined as the ratio of unmatched PP slots over all word slots with non-blank punctuation.

4.2. Evaluation of Punctuation

The RNN punctuation model yields posteriors for each punctuation class (comma, period). Based on these, an operation curve can be plotted for precision and recall. Fig. 1 shows operation curves 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 (word boundaries and SIL_{dur} obtained by forced alignment). In operating points more relevant for exploitation (high precision), not much difference is seen between a dense and a sparse alignment. By choosing the label with the highest posterior as the predicted punctuation, we also report overall performance metrics for individual operating points completed by SER and SMR in Table 2. 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 (reader) and hence we propose to maintain the system operating in the most upper quartile of the PR diagram. Using dense alignment is moreover advantageous from the perspective of SMR and latency, as the denser the PP alignment, the lower becomes the average PP_{dur}.

Regarding the window span for sequence generation at the output, we observed modest impact on performance with $W = 4$ being the optimal choice (for most of the tested PP densities). Longer windows did not result in improved performance, hence we propose to use a short window to make the punctuation model even simpler.

Beside a forced alignment version, we evaluated punctuation on data taken from ASR output for the BABEL test set (WER=7.5%). Given our feature streams interfere only with pause duration features from the ASR, and that PP alignment is independent, this influences only slightly punctuation results (see Table 2), and shows a very robust punctuation approach. Recall of commas is lower for ASR output by augmented SMR.

Table 2: *Punctuation performance on different test sets (BABEL and BN) and on ASR transcripts.*

Testset	comma			period			[%]	
	Pr	Rc	F1	Pr	Rc	F1	SER	SMR
BABEL force aligned, dense	0.83	0.45	0.58	0.82	0.89	0.85	39.4	2.0
BABEL force aligned, sparse	0.81	0.42	0.55	0.85	0.86	0.85	40.3	5.1
BABEL ASR transcript, dense	0.74	0.44	0.56	0.83	0.83	0.83	39.1	6.5
BABEL ASR transcript, sparse	0.72	0.49	0.59	0.81	0.82	0.82	38.3	7.3
BN ASR transcript, dense	0.43	0.38	0.40	0.76	0.73	0.75	51.2	7.2
BN ASR transcript, sparse	0.45	0.38	0.41	0.77	0.77	0.77	54.8	9.7
BN ASR + adapt RNN on BN, dense	0.55	0.32	0.41	0.80	0.74	0.77	45.5	6.5
BN ASR + adapt RNN on BN, sparse	0.82	0.25	0.38	0.80	0.76	0.78	51.3	9.0

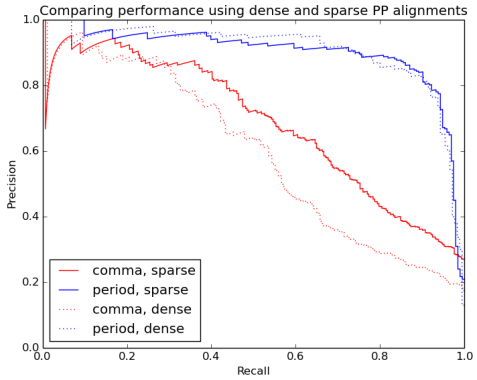

Figure 1: *Operation curves in the precision-recall space with dense and sparse PP segmentation for commas and periods.*

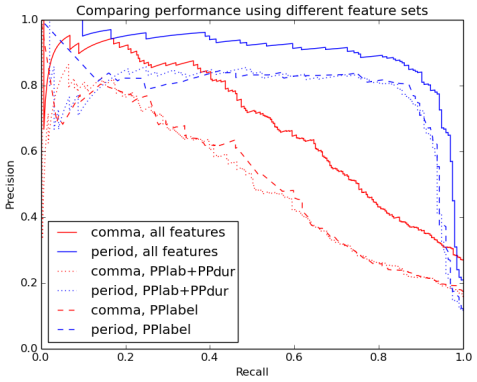

Figure 2: *Effect of using reduced feature sets. BABEL test set, sparse PP segmentation.*

With the broadcast news (BN) corpus (ASR with WER=10.5%), punctuation results for commas drop drastically (Table 2). By adapting the RNN models (running +10 epochs on adaptation data) on held out BN data, we notice modest improvement only in period precision (for commas, only the operation point is shifted by 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. We suppose that syntax can convey the structural information as news tend to have clear and standard syntax. Given the missing acoustic cues, such comma slots seem to be addressable only by adding text based features to our approach in the BN case.

4.3. Contribution of Feature Streams

We examined the contribution of the different feature streams to punctuation. Fig. 2 shows results with using PP_{label} only, $PP_{label} + PP_{dur}$ and $PP_{label} + PP_{dur} + SIL_{dur}$ features. It is remarkable that without the SIL_{dur} features, performance saturates around 0.80 precision and cannot improve further. We were experimenting with a fourth feature stream by adding acoustic confidence of the aligned PPs, but observed that this deteriorates the performance (introduces noise). We were also experimenting with a direct word slot based feature extraction, but this gave us unsatisfactory results. Therefore we also argue that supra-segmental structure and phrase sequence as is are important cues for prosody based punctuation which are mostly missed when extraction of prosodic features is considered only locally (around hypothesized word boundaries) and not in a supra-segmental approach.

5. Conclusions

In this paper we proposed a prosody based punctuation scheme based on automatically recovering the PP structure, and model this coupled with ASR pause duration features in a sequential approach with a small BiLSTM neural network. We obtained excellent punctuation recovery on a lecture alike corpus. The proposed method is fast to train and introduces little latency (with a single target slot preceding the last PP in the sequence) by being robust against ASR errors. We outlined a slight impact of PP alignment density and sequence window on punctuation. Switching to a broadcast news corpus, punctuation power was maintainable for periods using some adaptation cycles for the RNN model, but we noticed a considerable performance drop for commas. Indeed, acoustic evidence seems to be limited regarding comma slots in broadcast news, which can be counteracted by including text based features in a hybrid approach (the proposed RNN module can also yield posteriors for this). We believe the strengths of the proposed prosody based punctuation approach are its low computational needs, automatic PP recovery, ASR error robustness, RNN approach and little latency.

6. Acknowledgements

The authors were funded by the National Research, Development and Innovation Office of Hungary (PD-112598) and by EU MALORCA Project (Grant agreement No: 698824). The authors are grateful for the Titan GPU provided by NVIDIA.

7. References

[1] M. Paulik, S. Rao, I. Lane, S. Vogel, and T. Schultz, "Sentence segmentation and punctuation recovery for spoken language translation," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008, pp. 5105–5108.

[2] 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*, vol. 20, no. 2, pp. 474–485, 2012.

[3] 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.

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

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

[6] D. Beeferman, A. Berger, and J. Lafferty, "Cyberpunc: A lightweight punctuation annotation system for speech," in *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, vol. 2. IEEE, 1998, pp. 689–692.

[7] 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.

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

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

[10] E. Selkirk, "The syntax-phonology interface," in *International Encyclopaedia of the Social and Behavioural Sciences*. Oxford: Pergamon, 2001, pp. 15 407–15 412.

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

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

[13] S. Millotte, R. Wales, and A. Christophe, "Phrasal prosody disambiguates syntax," *Language and cognitive processes*, vol. 22, no. 6, pp. 898–909, 2007.

[14] K. Vicsi and G. Szaszák, "Using prosody to improve automatic speech recognition," *Speech Communication*, vol. 52, no. 5, pp. 413–426, 2010.

[15] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[16] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

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

[18] J. Žibert, F. Mihelič, J. P. Martens, H. Meinedo, J. P. D. S. Neto, L. Docio, C. G. Garcia-Mateo, P. David, J. Žďánskỳ, M. Pleva *et al.*, "The cost278 broadcast news segmentation and speaker clustering evaluation-overview, methodology, systems, results," in *9th European Conference on Speech Communication and Technology*, 2005.

[19] 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.