9th International Conference on Speech Prosody 2018
13-16 June 2018, Poznań, Poland

Distributed representation of melodic contours

Daniil Kocharov, Alla Menshikova

Saint Petersburg State University, Russia

[kocharov@phonetics.pu.ru, menshikova.alla2016@yandex.ru]

Abstract

We introduce a new computational model for melodic contours—melody embeddings. It is based on the approach of distributional semantics where embeddings represent units as continuous vectors in a multi-dimensional space based on hypothesis that units with similar meaning are used in similar contexts. This paradigm is applied to melodic contours and their segments. Melodic contours are represented by vectors of the same dimensionality independent on their length and shape. We successfully evaluated the ability of the proposed model to measure the distance between melodic contours. The results of applying the model for a task of prominent words detection have not showed the improvement over traditional prosodic features. Nevertheless we assume the model to be very promising. The possible applications for the proposed unsupervised prosodic model include processing of speech of under-resourced languages, modelling prosodic variability for text-to-speech synthesis, recognition and classification of prosodic events by means of deep-learning algorithms.

Index Terms: prosody, melody, unsupervised clustering, distributed representations, embeddings.

1. Introduction

We introduce a new computational method that models a melodic unit using the information about its context.

There is multiple evidence that melodic context is important for both perception and production of melodic contours. Cutler and colleagues showed that prosody helps to forecast focus location and intonation [1], [2]. Grosjean showed that listeners use various prosodic cues to predict the length of sentence given its beginning [3]. Linguistic research of different languages show that prenuclear and postnuclear melody corresponds to nuclear accent and is not arbitrary. Despite the fact that final accent is the most salient intonational cue, prenuclear intonation plays the role in distinguishing statements and questions in German [4], Greek [5] and Russian [6]. There is published evidence for European Portuguese [7] and Russian [8] that post-nuclear intonation can coincide with nuclear intonation.

Word meanings are successfully detected by contextual information in the field of computational semantics and language modelling. The most widely used approach for this is the word embedding method. Word embedding approach has become a state-of-the-art method since its introduction by Mikolov [9]. The word embeddings represent words as continuous vectors in a multi-dimensional space based on the hypothesis that words with similar meaning are used in similar contexts. The method has also been successfully applied to non-NLP tasks, e.g. modelling gene and protein sequences [10] and health care [11]. In the field of prosody modelling, word embeddings were applied for prosodic events recognition [12]. Ribeiro and colleagues used prosodic data to improve word embeddings applied in text-to-speech synthesis [13]. Their prosodic data included clustered information on fundamental frequency and energy. Authors declare that the proposed approach led to better text-to-speech synthesis in terms of subjective evaluation.

We propose to use the distributional approach for melody modelling, i.e. melody embeddings. The novelty of our approach is to train multidimensional melody embeddings purely on melodic information.

There are unsupervised methods that cluster prosodic information in multidimensional prosodic vectors, e.g. Principal Component Analysis [14], [15], Functional Data Analysis [16], [17], Self-Organizing Maps and Functional Principal Component Analysis [18]. However, none of them use prosodic context information as explicitly as the distributional approach.

The main problem for applying word embeddings to prosodic data is that it requires tokenization, while there are no clear tokens in melodic contours. Recently Schütze presented a tokenization-free method that processes a string of symbols with no assumption of any token boundaries [19].

This is why we applied embedding representation to melodic contours as it was defined by Schütze. We evaluated the proposed model in two tasks: (1) estimation of distance between two melodic contours and (2) using F_0 information for prosodic events detection.

2. Method

The procedure of building the melody embedding model consists of three steps: (1) calculating the stylized melodic contour; (2) coding the melodic information; (3) obtaining the vector representation of melody.

2.1. Calculating the melodic contour

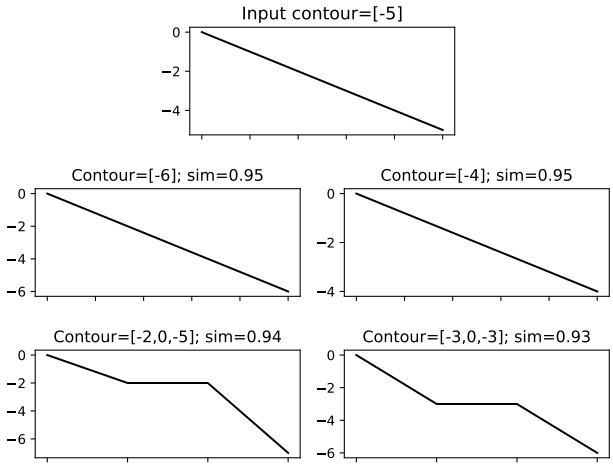
Fundamental frequency (F_0) is detected by the pitch tracking tool taken from Kaldi ASR toolkit [20]. The pitch tracker calculates F_0 and probability of voicing for each processing frame. We keep only the F_0 values with the probability of voicing above 0.85. The F_0 errors and microprosodic events are automatically detected and eliminated from melodic contours, then voiceless parts are bridged by means of linear interpolation. Finally, the contour is smoothed by Savitzky-Golay filtering using second order polynomial in 5 sample windows [21].

2.2. Coding

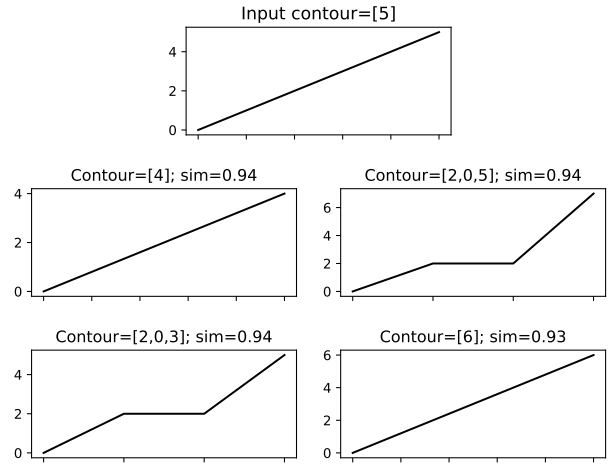
The goal of this step is to code F_0 movements with symbols so that we could apply text processing techniques.

The most common melody coding methods include ToBI [22], Tilt [23], INTSINT [24], SLAM [25]. We use none of the existing coding schemes as we consider them to be too general for our purpose: ToBI and INTSINT describe only the most important points of the melodic contour, whereas Tilt and SLAM describe the contour in terms of very few symbols and quantize F_0 values in broad intervals. Thus we applied our own coding scheme.

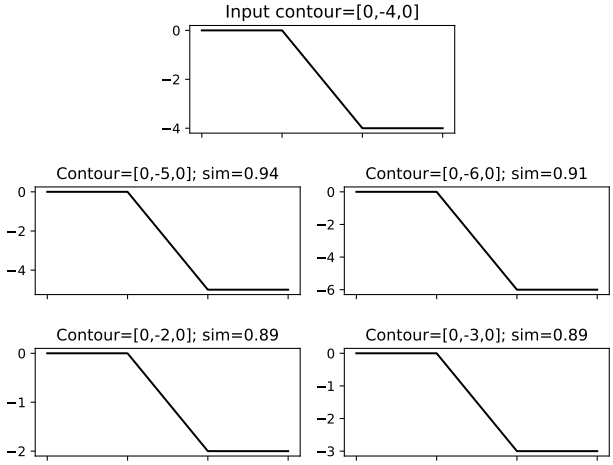
10.21437/SpeechProsody.2018-34



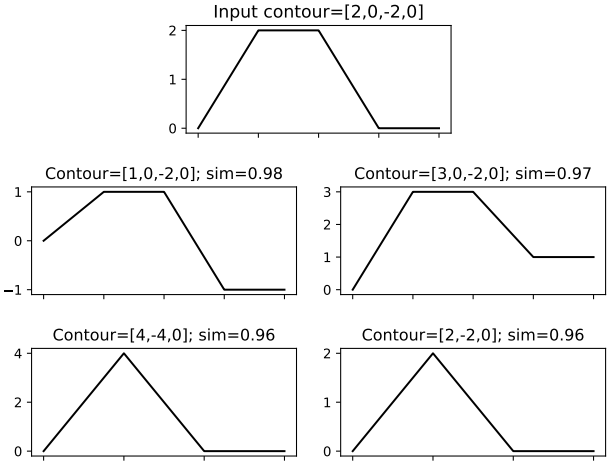
(a) *Nearest contours to a fall of 5 semitones*



(b) *Nearest contours to a rise of 5 semitones*



(c) *Nearest contours to a compound fall contour*



(d) *Nearest contours to a compound rise-fall contour*

Figure 1: *Nearest contours to the most frequent contours of different shape with indication of their cosine similarity*

The smoothed melodic contour is processed in non-overlapping 50 ms frames. The F_0 movement range within each frame is calculated in semitones. The movement is defined as rising, falling or level by the relative position of F_0 maximum and minimum within the frame. The contour is split into slopes—sequences of frames with identical direction of F_0 movement. The sum of F_0 values for each frame is a measure of F_0 movement of the slope.

Next, each slope is coded by a letter of Latin alphabet. The coding scheme is simple. Each value in semitones corresponds to a letter: 1—a, 2—b, 3—c, ..., 26—z. Positive values are coded by lower-case letters and negative values are coded by upper-case letters. Level slopes with the range of zero semitones are coded by the symbol '='. For instance, the melodic contour that consists of three slopes '*2, 0, -3*' is coded by the string '*b=C*'. The resulting coded melodic contour has no information on the duration of slopes, it contains only information on their range.

We have conducted a series of preliminary experiments on relatively small amount of speech data. They have shown several tendencies. First, it is better to discard slopes that are single-frame-long, as they add random noise to the final model. Second, adding temporal information to the coding significantly decreases the efficiency of the resulting embedding model. This might be due to a significant increase in the total number of basic symbols: from 53 to about 400 as in this case the 'symbol' consists of a letter and a value for slope duration (e.g. '*b4 =2 C3*').

2.3. Embedding

The coded string of symbols is used as input for the embedding procedure. We follow the tokenization-free approach of embedding representation [19]. We apply the method of multiple random segmentation—split a symbol representation of melodic contour into sequences of non-overlapping segments of random length ranging from k_{min} to k_{max} ('n-grams'). The result of such segmentation is a sequence of n-grams up to k_{max} symbols long. Each symbol representation is split *m* times in order to provide a better coverage of symbol n-grams. Results of all random segmentations are concatenated into one 'text'.

To learn the embeddings we use the skip-gram objective. The method attempts to predict surrounding n-grams from each symbol n-gram. The idea of the skip-gram model [9] is to

Table 1: *Comparison of various feature sets applied to prominent words detection*

Feature set	Precision	Recall	F_1-measure
F_0 maximum movement	64.9	78.0	**70.9**
F_0 melodic contour	62.6	71.7	66.8
ProsVec	61.2	70.7	65.6
Temporal features	59.9	77.0	67.4
Intensity features	55.8	62.2	58.8
F_0 max movement, ProsVec	66.6	75.9	70.9
F_0 melodic contour, ProsVec	62.2	72.0	66.8
F_0 max movement, F_0 melodic contour	69.6	77.0	**73.1**
F_0 max movement, F_0 melodic contour, ProsVec	69.3	75.9	72.5
F_0 max movement, F_0 melodic contour , temporal features, intensity features	74.4	83.0	**78.5**
F_0 max movement, F_0 melodic contour , temporal features, intensity features, ProsVec	74.1	82.9	78.3

predict the surrounding context words given the central word. Words are represented as n-dimensional vectors and the model is produced by a neural network with a single hidden layer. The network is trained to minimize the negative log-likelihood:

$$\begin{aligned} &-\log P(w_{c-h},...,w_{c-1},w_{c+1},...,w_{c+h}|w_c) \\ &= -\sum_{j=0,j\neq h}^{2h} u_{c-h+j}^T v_c + 2h\log\sum_{k=1}^{|V|}\exp(u_k^T v_c), \end{aligned} \quad (1)$$

where w_i is a word in the vocabulary V; word w_c is taken within each context $c-h,...,c+h$; v_c and u_c are the input and the output vector representations of word w_c.

3. Experimental Material

To train the melody embedding model we used 320 hours of speech. The recordings consisted of a set of fiction and non-fiction audio-books in mp3 format with various quality ranging from 92 kbps to 320 kbps. Mp3-files were converted to wav format by means of ffmpeg and than processed by the procedure described in Section 2.

To evaluate the proposed model in the experiments we used CORPRES (Corpus of Professionally Read Speech) developed at the Department of Phonetics, St. Petersburg State University [26]. The corpus contains recordings of read speech; the total duration is over 30 hours; it contains over 64 000 intonational phrases and 200 000 running words. The annotation consists of pitch, phonetic, orthographic and prosodic tiers with transcription and segmentation. The prosodic tier includes boundaries of intonational units, indications of phrase accents and prominent words.

We skipped the first step of automatic pitch detection described in Sec. 2.1, as the corpus annotation has a pitch tier.

4. Experimental Results

In our experiments, word length was from k_{min}=1 to k_{max}=4 symbols. The corpus was segmented 20 times. The dimension of v_c and u_c vectors was set to 50. The considered context length for the skip-gram model equalled to 3 words.

We used a Python implementation of word2vec, Gensim [27], for learning embeddings of letter n-grams. We trained a word2vec skip-gram model on the material, with intonation units presented as sentences, and its 'melodic segments' (short melodic contours from 1 to 4 slopes long) as words. The resulting material contained 4.7 million 'melodic segments'. The vocabulary of the model has about 82 000 'melodic segments' including 55 000 'melodic segments' found less than 10 times.

To estimate the efficiency of the proposed prosodic model in two typical prosody modelling tasks we conducted two experiments: (1) estimation of the distance between two contours and (2) using F_0 information for prosodic events detection.

4.1. Distance calculation

As melodic contours are represented by vectors, we were able to calculate the distance between different contours. As a measure of distance between two vectors we used cosine similarity. Its values range from '0' (most distant) to '1' (nearest).

For this experiment we took fifty most frequent contours with length ranging from 1 to 4 slopes. Then we calculated the contours whose representation vectors were closest to the vector of each frequent contour. For each frequent contour we plotted and analysed four nearest contours. Fig. 1 illustrates these plots of the most frequent contours. The plots include the information about cosine similarity between the input contour and its neighbours. The analysis showed that the nearest contours of each frequent contour seem reasonable.

The most positive thing about the proposed model is that it is able to calculate the distance between contours of different length, e.g. on Fig. 1a among the nearest contours there are both simple falls and compound falls with level F_0 in the middle.

4.2. Detection of prominent words

We applied the proposed model to a very common detection task—prominent words detection. Prominent words are considered to be words prosodically emphasized by the speaker and thus standing out of the surrounding words. We tested embeddings performance against commonly used melodic, temporal and intensity features:

- F_0 maximum movement. It is the speed of the most prominent F_0 movement within the word (in semitones per frame). The range of F_0 maximum movement in semitones was divided by the number of 50 ms frames; it was negative for falling contours and positive for rising contours.
- F_0 melodic contour. A sequence of three F_0 movements within the word (in semitones). If the number of F_0 movements within the word exceeded three, the sequence with the maximum sum of absolute values was chosen.
- Temporal features: maximum, minimum and mean

sound duration within the word, normalized for the speaker.

- Intensity features: maximum, minimum and mean values within the word, normalized for the speaker.

As embedding feature (ProsVec) we used position embeddings, proposed in [19] and [28], since it was shown that this implementation allows to preserve sequence information better, as well as compensate for melody units non present in the vocabulary of the embedding model. For each word rise-fall or fall-rise F_0 movement of the largest amplitude was detected. Then all sequences presented in the word and containing the detected sequence as a sub-sequence were selected. The resulting embedding vector was calculated as a sum of vectors representing the selected sequences.

We trained and tested a support-vector machine classifier [29] on the set of 211 384 words (68 645 of them were marked as prominent), using 5-fold cross-validation. We tested and compared different feature combinations. The results are presented in the Table 1.

The efficiency of the embedding feature is almost the same but still lower than the efficiency of the feature it is aimed to substitute, i.e. the melodic contour. Both of them have significantly lower efficiency than the speed of the most prominent F_0 movement. The results are disappointing considering that the first experiment showed that the embedding model is suitable for melodic contour comparison.

There are several possible reasons why melody embeddings neither outperform pure melodic information nor increase the efficiency when used as a complimentary feature. First, the embedding feature is a 50-dimensional vector that significantly increases dimensionality of the feature space making it very sparse when there is a constant number of classified objects. Second, the position-embedding approach that works well for text-mining tasks may be unsuitable for prosody events classification. Nevertheless, one can see that vectors are well distributed and there are several clear clusters, see Fig. 2 that shows a visualization of position-embedding vectors that have been used for classification mapped to two-dimensional space by means of t-SNE algorithm [30].

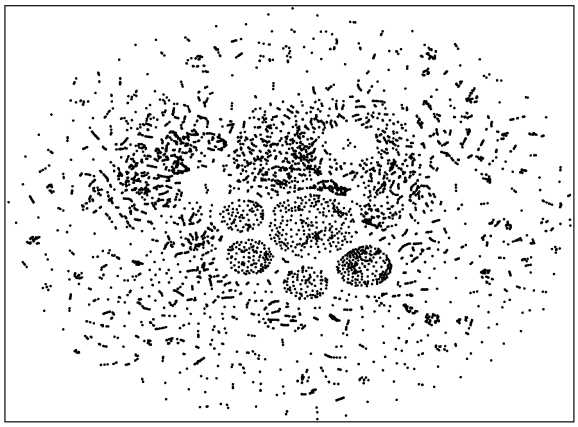


Figure 2: *Distribution of melody position-embeddings mapped to 2D space*

5. Discussion

We have presented the prosody embedding model—a new computational model of prosody. It models melodic unit based on statistical information on its context.

We tested it in two experiments and obtained controversial results. On the one hand, the model is able to detect nearest neighbours of frequent melodic contours. On the other hand, it does not increase efficiency compared with pure melodic information when used for prominent words detection.

The first and easy explanation could be that it just does not work for prosody as it does for lexical information. Lexical information is categorical and has no numerical structure. Melodic contours may be represented as linear sequences of numeral values and may be compared with each other. Nevertheless, the results of the first experiment demonstrate that it works for frequent melodic contours. The embedding feature bears context information and might be considered as complementary.

We think that the most crucial reason for the failure of the model in the second experiment is that it was trained on a small amount of data. It is well-known that the quality of the embedding model increases significantly with amount of the training data. Dhingra and colleagues showed that using training corpora smaller than billions of tokens leads to the same efficiency in solving the reading comprehension task as using random vectors [31]. The amount of our training data was 320 hours of speech. It was large enough to perform prosody modelling before, but for training embedding vectors of good quality this is certainly not enough. We have 4.7 million melody tokens in 320 hours of speech. In comparison, Wikipedia, which is the basic dataset in text processing, contains 3 billion words. Thus, we have to use a 100 times larger corpus to achieve comparable quality.

There is a number of possible improvements and evolutionary steps to enhance the quality of the modelling method. First, one can increase the quality of data used for model training: (1) we used a simple stylization procedure that may be improved further; (2) our coding procedure was also very simple. The coding scheme could be changed to decrease the number of quantization intervals. We coded each melodic movement but one may want to code accents only.

Second, currently there is a number of methods to train embeddings: skip-gram, continuous bag-of-words [9] and GloVe [32]. We have tested only one of them, the skip-gram model, but one can test the others.

Third, current models do not use information on the relative position of context tokens around the given one, and this seems important for prosody. Mikolov and colleagues have started to use word n-grams to enhance embedding results in distributional semantics [33] and one could do similar for melodic data.

We are going to test the proposed improvements in the nearest future and present the results. We assume that this method is very promising, our decision is grounded on its ability to define nearest melodic contours. We see many possible applications for the proposed unsupervised prosodic model including processing of speech in under-resourced languages, modelling prosodic variability for text-to-speech synthesis, recognition and classification of prosodic events by means of deep-learning algorithms.

6. References

[1] A. Cutler, "Phoneme-monitoring reaction time as a function of preceding intonation contour," *Perception and Psychophysics*, vol. 20, no. 1, pp. 55–60, 1976.

[2] M. H. K. Ip and A. Cutler, "Intonation facilitates prediction of focus even in the presence of lexical tones," in *Proceedings of Interspeech 2017*, 2017, pp. 1218–1222.

[3] F. Grosjean, "How long is the sentence? Prediction and prosody in the on-line processing of language," *Linguistics*, vol. 21, pp. 501–529, 1983.

[4] C. Petrone and O. Niebuhr, "On the intonation of German intonation questions: The role of the prenuclear region," *Language and Speech*, vol. 57, no. 1, pp. 108–146, 2014.

[5] M. Baltazani, E. Kainada, K. Nicolaidis, and A. Lengeris, "The prenuclear field matters: Questions and statements in standard modern Greek," in *Proceedings of 18th International Congress of Phonetic Sciences*, 2015.

[6] D. Kocharov, N. Volskaya, and P. Skrelin, "F0 declination in Russian revised," in *Proceedings of 18th International Congress of Phonetic Sciences*, 2015.

[7] S. Frota, "Nuclear falls and rises in European Portuguese: A phonological analysis of declarative and question intonation," *Probus*, vol. 14, no. 1, pp. 113–146, 2006.

[8] N. Volskaya and T. Kachkovskaia, "Prosodic annotation in the new corpus of Russian spontaneous speech CoRuSS," in *Proceedings of Speech Prosody 2016*, 2016.

[9] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of Neural Information Processing Systems 2013*, 2013.

[10] E. Asgari and M. R. K. Mofrad, "Continuous distributed representation of biological sequences for deep proteomics and genomics," *PLOS ONE*, vol. 10, pp. 1–15, 2015.

[11] E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart, and J. Sun, "Doctor AI: Predicting clinical events via recurrent neural networks," in *Proceedings of the 1st Machine Learning for Healthcare Conference*, ser. Proceedings of Machine Learning Research, vol. 56, 2016, pp. 301–318.

[12] A. Rendel, R. Fernandez, R. Hoory, and B. Ramabhadran, "Using continuous lexical embeddings to improve symbolic-prosody prediction in a text-to-speech front-end," in *Proceedings of ICASSP 2016*, 2016, pp. 5655–5659.

[13] M. S. Ribeiro, O. Watts, and J. Yamagishi, "Learning word vector representations based on acoustic counts," in *Proceedings of Interspeech 2017*, 2017, pp. 799–803.

[14] N. G. Ward, "Automatic discovery of simply-composable prosodic elements," in *Proceedings of Speech Prosody 2014*, 2014, pp. 915–919.

[15] N. G. Ward, S. D. Werner, F. Garcia, and E. Sanchis, "A prosody-based vector-space model of dialog activity for information retrieval," *Speech Communication*, vol. 68, pp. 85–96, 2015.

[16] B. Parrell, S. Lee, and D. Byrd, "Evaluation of prosodic juncture strength using functional data analysis," *Journal of Phonetics*, vol. 41, pp. 442–452, 2013.

[17] O. Jokisch and G. Pintér, "Intonation-based classification of language proficiency using FDA," in *Proceedings of Speech Prosody 2014*, 2014, pp. 795–799.

[18] Y. Asano, M. Gubian, and D. Sacha, "Cutting down on manual pitch contour annotation using data modelling," in *Proceedings of Speech Prosody 2016*, 2016, pp. 282–286.

[19] H. Schütze, "Nonsymbolic text representation," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, vol. 1, 2017, pp. 785–796.

[20] P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Trmal, and S. Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 2494–2498.

[21] A. Savitzky and M. J. E. Golay, "Smoothing and differentiation of data by simplified least squares procedures," *Analytical Chemistry*, vol. 39, no. 8, pp. 1627–1639, 1964.

[22] K. Silverman, M. Beckman, J. Pitrelli, M. Ostendorf, C. Wightman, P. Price, J. Pierrehumbert, and J. Hirschberg, "ToBI: A standard for labeling English prosody," in *The Second International Conference on Spoken Language Processing, ICSLP 1992*, 1992, pp. 867–870.

[23] P. Taylor, "The rise/fall/connection model of intonation," *Speech Communication*, vol. 15, no. 1, pp. 169–186, 1994.

[24] D. Hirst, A. Di Cristo, and R. Espesser, "Levels of representation and levels of analysis for the description of intonation systems," in *Prosody: Theory and Experiment*, M. Horne, Ed. Springer Netherlands, 2000, pp. 51–87.

[25] N. Obin, J. Beliao, C. Veaux, and A. Lacheret, "SLAM: Automatic stylization and labelling of speech melody," in *Proceedings of Speech Prosody 2014*, 2014.

[26] P. A. Skrelin, N. B. Volskaya, D. Kocharov, K. Evgrafova, O. Glotova, and V. Evdokimova, "A fully annotated corpus of Russian speech," in *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, 2010, pp. 109–112.

[27] R. Řehůřek and P. Sojka, "Software framework for topic modelling with large corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 2010, pp. 45–50.

[28] N. Kalchbrenner, L. Espeholt, K. Simonyan, A. van den Oord, A. Graves, and K. Kavukcuoglu, "Neural machine translation in linear time," *CoRR*, vol. abs/1610.10099, 2016. [Online]. Available: http://arxiv.org/abs/1610.10099

[29] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[30] L. v. d. Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, no. 9, pp. 2579–2605, 2008.

[31] B. Dhingra, H. Liu, R. Salakhutdinov, and W. W. Cohen, "A comparative study of word embeddings for reading comprehension," *CoRR*, vol. abs/1703.00993, 2017.

[32] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proceedings of EMNLP 2014*, 2014.

[33] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin, "Advances in pre-training distributed word representations," *CoRR*, vol. abs/1712.09405, 2017.