Interspeech 2018
2-6 September 2018, Hyderabad

Expressive speech synthesis using sentiment embeddings

Igor Jauk[1], *Jaime Lorenzo-Trueba*[2], *Junichi Yamagishi*[2], *Antonio Bonafonte*[1]

[1]Universitat Politècnica de Catalunya, Barcelona, Spain
[2]National Institute of Informatics, Tokyo, Japan

ij.artium@gmail.com, jaime@nii.ac.jp,
jyamagis@nii.ac.jp, antonio.bonafonte@upc.edu

Abstract

In this paper we present a DNN based speech synthesis system trained on an audiobook including sentiment features predicted by the Stanford sentiment parser. The baseline system uses DNN to predict acoustic parameters based on conventional linguistic features, as they have been used in statistical parametric speech synthesis. The predicted parameters are transformed into speech using a conventional high-quality vocoder. In this paper, the conventional linguistic features are enriched using sentiment features. Different sentiment representations have been considered, combining sentiment probabilities with hierarchical distance and context. After preliminary analysis a listening experiment is conducted, where participants evaluate the different systems. The results show the usefulness of the proposed features and reveal differences between expert and non-expert TTS user.

Index Terms: Expressive speech synthesis, sentiment analysis, TTS, DNN

1. Introduction

Semantic vector representations of text have been used to perform a look-up in the training corpus for expressive speech data according to the textual input, such that, relying on semantic information, data clusters were used to train expressive voices via speaker adaptation, as for example in [1]. A logical evolution of this study is to use embeddings which are more dedicated to the expressiveness in text. The *Stanford Sentiment Parser* is such a tool, which provides vector embeddings reflecting the sentiment, i.e. the positiveness or the negativeness of the text. For more details refer to Section 2.1.

The Stanford parser is trained on labeled movie reviews, originally collected and published by [6]. The input to the Stanford parser is a textual unit, word level or more. First, the input is parsed and converted into a binary tree structure. Then, for each level the system predicts a sentiment. The format can be just a value, between *positive*, *negative* or *neutral*, a probability of belonging to one of the five categories *very positive*, *positive*, *very negative*, *negative* or *neutral*, or a vector embedding in a sentiment vector space.

In preliminary experiments, sentiment vectors were calculated for sentences of several corpora and a prosodic analysis was conducted examining the influence of sentiment on prosody. The results showed there is an actual effect, especially on F0.

A further improvement in comparison to work presented in [1] is the migration from HMM-based synthesis to DNN-based synthesis. A main drawback of the HMM-based synthesis is that the training data is clustered. This is a disadvantage, for clustering relies on extracted features, in this case representing expressiveness, however, even if the features are very good, there will always be an error. This will cause that data points which should belong to a certain training cluster are not inside, and others, which do not belong to the training cluster, are inside.

DNN-based synthesis, in certain manner, avoids this problem because the network sees the complete data set, and the neurons "decide" according to the training criterion, which output data (speech), corresponds to which input data (in this case, also to embeddings). In this sense, there is a kind of abstracted intern clustering optimized according to the training criterion.

In previous work, neural network based systems have already been combined with semantic vector input, though not for expressive speech. To name a few, Wang et al [13] use word embeddings to substitute TOBI and POS tags in RNN-based synthesis achieving significant system improvement. Wang et al [14] enhance the input to NN-based systems with continuous word embeddings, and also try to substitute the conventional linguistic input by the word embeddings. They do not achieve performance improvement, however, when they use phrase embeddings combined with phonetic context, they do achieve significant improvement in a DNN-based system. Wang et al [14] enhance word vectors with prosodic information, i.e. update them, achieving significant improvements.

In comparison to these systems, the system proposed here uses sentiment embeddings, i.e. the embeddings have an expressive meaning. Some speech synthesis systems have already used sentiment information. For instance, Trilla and Alias [11] already used sentiment analysis on sentence level for an expressive TTS. Vanmassenhove et al [12] also used sentiment combined with emotion labels for an HMM-based system. Sudhakar and Bensraj [9] implemented a TTS in Matlab which used sentiment information trained with fuzzy neural networks evaluated in a news domain. Differently from these systems, the proposed system uses sentiment for a DNN-based TTS in an audiobook domain, which is considerably open and rich in expressive speech.

The article is structured as follows. Section 2 presents the architecture of the proposed system, including sentiment vectors as described in 2.1. Section 3 describes the experiment and Section 3.1 presents the results. Finally, section 4 offers a discussion of the system and the results.

2. Expressive TTS System

The proposed architecture is basically an extension of a standard DNN TTS, where the DNN receives an additional input, the sentiment vectors, as shown in figure 1. The next section describes the *Stanford Sentiment Parser* which is used to generate the sentiment vectors. Afterwards, the system architecture is presented.

10.21437/Interspeech.2018-2467

2.1. Standford Sentiment Analysis

Socher et al [8] propose a recursive neural tensor network to create embeddings and to predict sentiment probabilities of terms. *Sentiment* is the valence, i.e. the positivity of the term. A term can be everything from word to sentence level. The network is trained on the labeled *Sentiment Treebank* which consists of a movie review database. The sentences are labeled as positive or negative reflecting the intention of the review publisher. Furthermore, reviews have been split in subphrases and annotated on a sentiment scale using Amazon Mechanical Turk. All sentences are parsed with the Stanford Parser, as by [5], and stored as binary trees.

The input to the sentiment parser is a sentence, the output can be the sentiment value (positive, negative, neutral), the probability, or the vector embedding of the sentiment for each binary node of the tree structure, from the top node down to the word level.

2.2. TTS Architecture

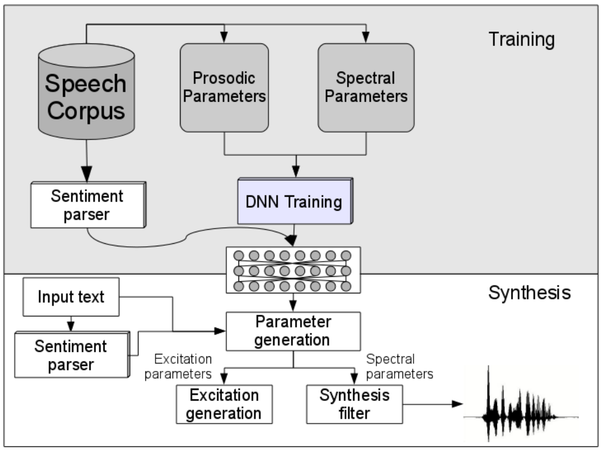

Figure 1: *Proposed DNN system architecture using sentiment embeddings.*

The underlying DNN system has the following specifications, as by [10]. For each utterance, a 60 dimensional MFCC vector, log F0, 25 dimensional band aperiodicity measures, and for each, dynamic and acceleration features are extracted. The log F0 is linearly interpolated and voiced/unvoiced marks are used as parameters. *Combilex*, by [7], is used to create context label files. First, an HMM-based training is performed, estimating phoneme boundaries. Then, the deep neural network is trained. The DNN is implemented with 5 hidden layers, each containing 1024 neurons. It is trained using Adagrad gradient optimization with minibatch size of 256. *Straight* vocoder, as by [4], is used to generate the waveform.

As proposed, an additional linguistic input is introduced, the sentiment predicted by the Stanford sentiment parser. Here, different input combinations are tested. Probability and embeddings are used alternatively in following configurations:

- **Without sentiment (ws)**: the standard DNN TTS without any embeddings.
- **Word level (wl)**: Word level probabilities and embeddings are used.
- **Word context and tree distance (wcd)**: Word context includes word level embeddings with two word embeddings on the left and on the right of the current word. It also includes the hierarchical tree distance for each word, i.e. the distance measured in number of tree nodes which separate two words. The aim is to stabilize the overall utterance prosody.

To visualize the input vectors, the Stanford parser probability vectors are composed as follows:

$$P = [p_{vneg}, p_{neg}, p_{neu}, p_{pos}, p_{vpos}] \tag{1}$$

where p_{vneg} is the probability of the category *very negative*, p_{neg} the probability of the category *negative*, etc. The probability vectors are provided on sentence level (sl) and word level (wl), in the respective cases. When word context was taken into account, probability vector of the word in question and the probability vectors of two words on the left and two words on the right were used. Also the tree distance, which is the hierarchical distance counted in the number of binary tree nodes between words is added, such that the input vector for each word for the system (v_wcd) is composed as follows:

$$P = \{P_{l_2}, P_{l_1}, P_c, P_{r_1}, P_{r_2}, D_t\} \tag{2}$$

where P_c is the probability vector for the current word, the P_{l_2} is the probability vector for the second word on the left, P_{l_1} is probability vector for the first word on the left, P_{r_1} is probability vector for the first word on the right and P_{r_2} is the probability vector for the second word on the right, each of the probability vectors as defined in equation 1. D is the hierarchical tree distance (distance in tree counted in nodes).

On the technical side, the vectors are always inserted on frame level. So for instance, when using word level probabilities, the embeddings were the same for all frames within a word, changing on word boundaries.

3. DNN-sentiment evaluation

For this experiment, two systems, *word level* and *word context and tree distance*, were chosen to synthesize 12 sentences in comparison to a system without sentiment, a total of 36 samples. The synthesized sentences are listed in Table 1.

The system, which architecture is shown in Figure 1, and which specifications are stated in Section 2, is trained with a clean portion of an audiobook corpus read by a semiprofessional male reader of American English. The audiobook portion contains 5039 sentences and is approximately 5 hours long. Apart of the features extracted for the DNN system, as stated in Section 2, for each of the sentences, a sentiment probability vector is calculated, using the Stanford sentiment parser, and added in the combinations described above as additional input to the system on frame level (except for the case without sentiment).

The sentiment is determined by the Stanford parser. The participants have no information whether a sentence is supposed to be positive or negative, they have to intuit it from the semantics. The task is to rate the systems, between 1 and 3, being 1 the best option and 3 the worst. The participants can rate the systems equally, if they consider them to be equally good or bad. They also have the option to disqualify a system, if they think that it is not adequate for a sentence at all.

3.1. Perceptual results

A total of 20 persons participated in the experiment, 12 of them reported to be experts in speech technology development, two have experience as users with speech technology, the others do not have experience with speech technology, one of them was native US-English speaker. Table 2 shows the average rankings and variances for the systems. As can be seen, the best performing system is the *word level* system, however, with a high variance. The system without sentiment was disqualified 1 time, the word level system 3 times, and the *word context and tree distance* system 0 times.

Table 3 shows the P-values for one- and two-tailed t-tests with $\alpha = 0.05$. The tests show that there is a significant difference between the system without sentiment and the *word level* system, but no significant difference between the system without sentiment and the *word context and tree distance* (although it is close), nor between the *word level* and the *word context and tree distance*.

Table 4 shows the preferences divided by the sentiment. For positive and negative sentences, the *word level* system performed best, although for negative sentences with high variance. For neutral sentences, the *word context and tree distance* system performed best. Possibly it is due to the fact that it probably has an equilibrating effect.

Table 5 shows the P-values for the t-tests for negative, neutral and positive sentences. For negative sentences, there is a significant difference between the system without sentiment and the *word level* system, and no significant difference for the other systems. For neutral sentences, there is a significant difference between the system without sentiment and the *word context and tree distance* system, but not for the other systems. For positive sentences, there is only significant difference for the one-tailed t-test between the system without sentiment and the *word level* system.

Table 1: *Synthesized sentences for the main experiment.*

neg1	*And if you fail I will kill you.*
neg2	*I indicated that dreadful lee shore.*
neg3	*I exclaimed startled out of myself by the picture.*
neg4	*The awful soundtrack was disgusting and made me puke.*
neu1	*My house is green with a big yellow door.*
neu2	*The movie is there and I am here.*
neu3	*It is the first day of June.*
neu4	*Each glass bottle has been paid for each metal can.*
pos1	*A woman's hair is wonderful.*
pos2	*The mate's strength was amazing.*
pos3	*Ellie was an inspiration to her friends and family.*
pos4	*I was extremely happy with the movie.*

Table 2: *System preferences. ws: without sentiment, wcd: word context and tree distance, wl: word level*

	ws	wcd	wl
mean	1.97	1.88	1.84
variance	0.59	0.68	0.86

Table 3: *One- and two-tailed t-test results, P-values. ws: without sentiment, wcd: word context and tree distance, wl: word level,* $\alpha = 0.05$

	one-tailed P	two-tailed P
ws/wcd	0.06	0.12
ws/wl	0.01	0.01
wl/wcd	0.28	0.55

Table 4: *System preferences for positive, negative and neutral sentences. ws: without sentiment, wcd: word context and tree distance, wl: word level*

	ws	wcd	wl
positive mean	1.84	1.85	1.71
positive variance	0.54	0.76	0.54
negative mean	2.06	1.96	1.84
negative variance	0.52	0.67	1.1
neutral mean	2	1.83	1.96
neutral variance	0.71	0.6	0.95

Table 5: *One- and two-tailed t-test results for positive, negative and neutral sentences, P-values. ws: without sentiment, wcd: word context and tree distance, wl: word level,* $\alpha = 0.05$

	Negative		Neutral		Positive	
	1-t.	2-t.	1-t.	2-t.	1-t.	2-t.
ws/wcd	0.12	0.24	0.01	0.02	0.46	0.92
ws/wl	0.01	0.02	0.36	0.72	0.04	0.08
wl/wcd	0.17	0.34	0.15	0.3	0.08	0.15

Among the comments of the participants, several stated that in some cases it was difficult to decide which system was better. Looking at the results of the only native speaker, he prefers the *word level* system with an average rank of 1.64, and he mostly discards the *word context and tree distance* system, with an average rank of 2.42. Only for neutral sentences he prefers the system without sentiment with an average rank of 1.50

Table 6: *System preferences between developer participants, user participants, and participants without experience with speech technology. ws: without sentiment, wcd: word context and tree distance, wl: word level*

	ws	wcd	wl
developer mean	2.01	1.79	1.77
developer variance	0.67	0.6	0.74
user mean	1.75	1.79	1.88
user variance	0.46	0.69	0.72
unexpert mean	1.94	2.08	1.96
unexpert variance	0.48	0.78	1.17

Table 6 shows preference results for users with different experience levels. Developer participants generally follow the tendency of the overall results, evaluating better the systems with

Table 7: *One- and two-tailed t-test results for developer participants, P-values. ws: without sentiment, wcd: word context and tree distance, wl: word level, $\alpha = 0.05$*

	one-tailed P	two-tailed P
ws/wcd	0.00	0.00
ws/wl	0.00	0.00
wl/wcd	0.22	0.44

Table 8: *One- and two-tailed t-test results for no-expert participants, P-values. ws: without sentiment, wcd: word context and tree distance, e wl: word level, $\alpha = 0.05$*

	one-tailed P	two-tailed P
ws/wcd	0.02	0.03
ws/wl	0.44	0.87
wl/wcd	0.06	0.12

sentiment than without. The P-values of the t-test for the developer participants are listed in Table 7. There are significant differences between both systems with sentiment and the system without sentiment, but no significant differences between the two systems with sentiment.

The user participants, on contrary, prefer the system without sentiment. However, the general tendency of the user participants is a rather good ranking of all systems, i.e. they considered more often that several systems were equally good. In any case, only two persons reported to be experienced user, with no further details how far this experience goes, which has no statistical importance, therefore no t-test is performed for the user participants.

The participants without experience preferred the system without sentiment and the system with *word level* sentiment, and pretty much discarded the system with *word context and tree distance*. The t-test results for the no-expert participants are listed in Table 8. The results show that there is a significant difference between the system without sentiment and the *word context and tree distance*, but no significant difference in other combinations. However, although the difference between the *word level* and the *word context and tree distance* is not significant, it is much bigger than the difference between the system without sentiment and the *word level* system. In general, and especially for participants without experience, the *word level* system has the highest variance.

4. Discussion

This work was dedicated to expressive speech synthesis with deep neural networks. For this, a DNN based speech synthesis system was trained on an audiobook, where additionally, sentiment input predicted by the Stanford sentiment parser was added to train the system. Three different configurations were tested, among them including the sentiment probability on word level, including word context and hierarchical tree distance, and without sentiment.

A perceptual experiment was conducted with test sentences synthesized using the different sentiment input configurations. It compared two sentiment systems with a system without sentiment. The overall results yield that the systems with sentiment are better. Also, there are differences between positive, negative, and neutral sentences. However, when the results are separated by the experience of the participants with speech technology, there are important differences between the groups. The developer confirm and accentuate the overall results. The participants without any experience often preferred the system without sentiment features. Those with user experience had a different tendency, although there were only two of them, making the interpretation of their results statistically irrelevant. The best performing system, the word level sentiment system, has also the highest variance. This is probably due to the fact that this system yields the strongest and most varied accentuations since it is driven by word-level sentiment. This can be perceived sometimes as good and sometimes as bad.

The results obtained in the experiments show the general potential of neural network based synthesis in combination with expressive information derived from text. The results show the general preference for the best performing system using this information. However, they also show that different designs of the input yield very different results in system performance, which probably means, that there is a lot more room for improvement.

Furthermore, the sentiment parser is trained on movie reviews, and the acoustic model on an audiobook. The consequence is that many sentences which are positive or negative in one domain, are different in the other domain. Also, movie reviews are usually written, and even if spoken, often with neutral voice. This discrepancy probably lowers the quality of the prediction, of the training, and of the synthesis. On the other hand, the original audiobook by itself, is not very expressive.

Future work should aim, first, at improving these conditions, the database and the sentiment parser. After that, the way how the sentiment information is used in the system should be studied and improved. One of the main point regarding this is that there should be a connection between the sentiment (or other) sentence embeddings and the actual acoustics. A good investigation could be to train the sentiment analysis in such a way that the sentiment output is adjusted not only to the labels on text level, but also to the acoustics. Features like i-vectors or i-vector based combinations proposed in [3, 2] could be used instead of labels, automatizing the process. This technique could also work for other semantic embeddings adjusting them to the acoustics and improving them for the expressiveness.

5. References

[1] I. Jauk and A. Bonafonte. Direct expressive voice training based on semantic selection. In *Proceedings of Interspeech*, pages 3181–3185, 2016.

[2] I. Jauk and A. Bonafonte. Prosodic and spectral ivectors for expressive speech synthesis. In *Proceedings of Speech Synthesis Workshop 9*, pages 59–63, 2016.

[3] I. Jauk, A. Bonafonte, P. López-Otero, and L. Docio-Fernandez. Creating expressive synthetic voices by unsupervised clustering of audiobooks. In *Interspeech 2015*, pages 3380–3384, 2015.

[4] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigne. Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based f0 extraction: Possible role of a repetitive structure in sounds. *Speech Communication*, 27(3-4):187–207, 1999.

[5] D. Klein and C.D. Manning. Accurate unlexicalized parsing. *ACL*, pages 423–430, 2003.

[6] B. Pang and L. Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *ACL*, pages 115–124, 2005.

[7] K. Richmond, R.A. Clark, and S. Fitt. Robust lts rules with the combilex speech technology lexicon. *Proceedings of Interspeech*, pages 1295–1298, 2009.

[8] R. Socher, A. Perelygin, J.Y. Wu, J. Chuang, C.D. Manning, A.Y. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013.

[9] B. Sudhakar and R. Bensraj. An efficient sentence-based sentiment analysis for expressive text-to-speech using fuzzy neural network. *Research Journal of Applied Sciences, Engineering and Technology*, 8(3):378–386, 2014.

[10] S. Takaki and J. Yamagishi. Constructing a deep neural network based spectral model for statistical speech synthesis. *Recent Advances in Nonlinear Speech Processing*, 48:117–125, 2016.

[11] A. Trilla and F. Alias. Sentence based sentiment analysis for expressive text-to-speech. *IEEE Transactions on Audio, Speech and Language Processing*, 21(2):223–233, 2013.

[12] E. Vanmassenhove, J. Cabral, and F. Haider. Prediction of emotions from text using sentiment analysis for expressive speech synthesis. *Proceedings of Speech Synthesis Workshop (SSW9)*, pages 119–124, 2016.

[13] P. Wang, Y. Qian, F.K. Soong, L. He, and H. Zhao. Word embedding for recurrent neural network based tts synthesis. In *Proceedings of International conference on acoustics, speech and signal processing (ICASSP)*, pages 4879–4883, 2015.

[14] X. Wang, S. Takaki, and J. Yamagishi. Investigating of using continuous representation of various linguistic units in neural network based text-to-speech synthesis. *IEICE Transactions on Information and Systems*, E99-D(10):2471–2480, 2016.