Learning word vector representations based on acoustic counts

Citation for published version:

Digital Object Identifier (DOI):
10.21437/Interspeech.2017-1340

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Interspeech 2017

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
Learning word vector representations based on acoustic counts

M. Sam Ribeiro¹, Oliver Watts¹, Junichi Yamagishi¹²

¹Centre for Speech Technology Research, University of Edinburgh, UK
²National Institute of Informatics, Tokyo, Japan
m.f.s.ribeiro@sms.ed.ac.uk, owatts@inf.ed.ac.uk, jyamagis@inf.ed.ac.uk

Abstract

This paper presents a simple count-based approach to learning word vector representations by leveraging statistics of co-occurrences between text and speech. This type of representation requires two discrete sequences of units defined across modalities. Two possible methods for the discretization of an acoustic signal are presented, which are then applied to fundamental frequency and energy contours of a transcribed corpus of speech, yielding a sequence of textual objects (e.g. words, syllables) aligned with a sequence of discrete acoustic events. Constructing a matrix recording the co-occurrence of textual objects with acoustic events and reducing its dimensionality with matrix decomposition results in a set of context-independent representations of word types. These are applied to the task of acoustic modelling for speech synthesis; objective and subjective results indicate that these representations are useful for the generation of acoustic parameters in a text-to-speech (TTS) system. In general, we observe that the more discretization approaches, acoustic signals, and levels of linguistic analysis are incorporated into a TTS system via these count-based representations, the better that TTS system performs.

Index Terms: speech synthesis, text-to-speech, vector representations, word embeddings, deep neural networks

1. Introduction

In statistical parametric speech synthesis, acoustic parameters are generated by an acoustic model and then used to drive a vocoder in order to obtain an artificially-generated speech waveform. The acoustic model has, in recent years, typically taken the form of a deep neural network (DNN) [1, 2]. The input to this model is often referred to as the linguistic specification, which is a representation designed to bridge the gap between text and speech. Common feature sets for English data, such as the one described in [3], mostly involve context-dependent phones, syllable stress, word part-of-speech, as well as various positional features describing phonetic and prosodic contexts within a text sentence. The group of modules that processes a text sentence and generates the corresponding linguistic specification is often termed the front-end.

However, earlier work in the context of HMM-based speech synthesis found that features defined at linguistic levels above the syllable have little impact on the prediction of acoustic parameters [4]. Good representations of higher-level phenomena (often related to syllables or words) are essential for accurate generation of natural speech prosody, especially in the context of expressive audiobook speech synthesis, where speech is expected to be more fluid and pleasing.

In this work, we investigate a method to learn acoustically-motivated representations of words and syllables for text-to-speech synthesis. For this task, we explore vector space models (VSM), which are a well-established approach for obtaining semantic representations in the field of Natural Language Processing (NLP). VSMs are rooted in the distributional hypothesis, which claims that words that have similar contexts tend to have similar meanings [5, 6]. The approaches chosen to learn such representations can be grouped into two main classes. These can be termed, following the terminology of [7], count models and predictive models.

The first class of models is defined by extracting co-occurrence statistics over large text corpora. Various transformations can be applied to the raw counts, such as context weighting or dimensionality reduction techniques [8, 9, 10]. Conversely, the second class of models frames the problem as a context prediction task. That is, given a word, it is the model’s objective to determine the context with which it occurs. Therefore, it is expected that words that have similar contexts will be mapped to similar representations in the low-dimensional dense space learned by the model [11, 12, 13, 14]. Investigations have been made into these two approaches, comparing them with various configurations on a set of semantic tasks. Although earlier work showed a clear preference for predictive models [7], recent work showed that their superiority might not be as obvious [15].

In terms of their application to speech synthesis, various approaches have been proposed. Representations learned with count-based methods have been explored as input features to modules within a TTS front-end (e.g. phrase-break prediction [16]), replacement of those modules (e.g. part-of-speech tagging [17]), or as direct input for acoustic modelling [17, 18]. With recent developments in neural network architectures, predictive approaches have gained popularity. Recent work investigated representations of words derived from large text databases [19, 20] and in combination with acoustic parameters [21].

In this work, we investigate a simple approach inspired by the traditional class of models based on co-occurrence statistics. Such statistics are extracted over a parallel corpus of text and speech and common transformations are applied to the raw count matrices. In a real-world scenario, these representations could be easily included in the front-end of a text-to-speech system as simple look-up tables. Section 2 reviews the methodology proposed for learning count-based word vector representations. Section 3 defines the dataset used, while Section 4 details a set of experiments investigating the effect of the learned representations on a text-to-speech acoustic model. A perceptual evaluation is described in Section 5, followed by a discussion of the results in Section 6.

2. Count-based representations

Given a fixed vocabulary $V$ and a fixed set of acoustic classes $A$, we define a count matrix $M \in \mathbb{R}^{|V| \times |A|}$. $M_{ij}$ denotes the number of times the $i^{th}$ class is observed occurring with the $j^{th}$ vocabulary unit. The vocabulary can be defined over textual objects (e.g. words, syllables). The classes can be defined by discretizing an acoustic signal, such as f0 or energy. Sections 2.1 and 2.2 provide details on how these classes are determined.
Because occurrences can be context-dependent, we can extend the set of classes over a unit type to account for neighboring occurrences of the acoustic class. If we set a window of size \( w \), then \( M \in \mathbb{R}^{|V| \times |w| |A|} \). For example, consider an utterance for which \( U \) is a sequence of linguistic units and \( C \) is the corresponding sequence of acoustic classes. If \( w = 3 \), then at timestep \( t \) we count the occurrence of \( C_{t-1} \), \( C_t \), and \( C_{t+1} \) in the \( i \)th row of \( M \), for which \( i \) is the vocabulary index of the unit \( U_t \). Note that, in this case, the tokens \( U_{t-1} \) and the \( U_{t+1} \) are not used for the counts of \( U_t \). Each row of the raw count matrix \( M \) is then normalized by the total number of counts within each sub-vector of occurrences. Therefore, each sub-vector of the \( i \)th token’s row is a probability distribution over the acoustic classes \( A \). Each token’s row consists of the concatenation of \( w \) probability distributions.

Finally, we reduce the dimensionality of the normalized count matrix \( M \) by finding the Singular Value Decomposition (SVD) of the matrix, such that: \( M = U \Sigma V^T \). We take \( k \) left singular vectors of \( M \), such that the sum of squares of the retained singular values is at least 90% of the sum of squares of all singular values. The result of this operation is a matrix \( \hat{U} \in \mathbb{R}^{|V| \times k} \). Each row of this matrix corresponds to a entry in the vocabulary \( V \), and we let that be the representation for that linguistic unit.

### 2.1. Cluster-based class definition

This section describes a possible approach to the quantization of an acoustic signal into a set of classes \( A \). We assume we are given a set of linguistic units, corresponding to entries in a vocabulary \( V \), and its acoustic signal, such as \( f_0 \), with known unit boundaries.

Within each utterance, the signal is normalized to zero mean and unit variance. For each unit, the Discrete Cosine Transform (DCT) is applied to the samples associated with the linguistic unit. The DCT stylizes a signal of length \( N \), as a large variance in terms of speaking styles. Therefore, utterance selection using an active learning approach [23, 24] was performed and a subset of utterances corresponding mostly to narrated speech were selected.

At this point, each unit is assigned a vector with \( d \) components representing its \( f_0 \) signal. We then use k-means clustering to group the acoustic vectors into classes. For clustering, we exclude the zeroth DCT coefficient, as that is approximately the mean energy of the signal and can heavily bias the clustering step. We can regard the clustered vector as a representation of the shape of the signal for a given linguistic unit. The acoustic classes \( A \) are defined to be the clusters identified by k-means. An additional class is added to represent silences such as pauses or hesitations.

2.2. Mean-based class definition

The cluster-based representations ignores the mean value of the unit when defining the acoustic classes. Therefore, a simpler approach quantizes the mean value of the signal over the entire linguistic unit. If we consider the \( f_0 \) signal, we might observe that a speaker’s range is mostly within 100-300Hz, as shown in Figure 1. We then define 100 classes over this interval, each spanning a range of 2Hz. Two additional classes are added to include occurrences below and above the interval. An additional class is added to represent silences. Note that there are several hyperparameters required by the two proposed class definitions, such as number of clusters, number of retained DCT coefficients, and bin size. Details of hyperparameter choices are given in Section 4.

3. Data

We use the data made available to the Blizzard Challenge 2013 [22], provided by Lessac Technologies Inc. and originally available from Voice Factory International Inc. The data consists of a single female speaker reading the text of classic novels. This database is of particular interest as the speaker is a professional narrator and actress, which suggests the prosodic variation correlates meaningfully with the text being read. It is also a large dataset, which is interesting for this type of study. However, as the speaker mimics character voices over several books, there is a large variance in terms of speaking styles. Therefore, utterance selection using an active learning approach [23, 24] was performed and a subset of utterances corresponding mostly to narrated speech were selected.

Given the utterance-level segmentation already available from the Blizzard challenge, state-level forced alignment was obtained using context independent HMMs with Festival1 and HTK2 via the Merlin toolkit [25]. Pauses were inserted motivated by acoustic evidence, using Festvox’s elmu[26]. The training set consists of approximately 18 hours of speech over 13000 utterances, with approximately 220k word and 300k syllable tokens. We set aside an additional 300 and 100 utterances for validation and test purposes.

---

1http://www.cstr.ed.ac.uk/projects/festival
2http://htk.eng.cam.ac.uk
4. Experiments

4.1. Baseline

The baseline system is a simple feedforward multilayer perceptron. A network of 6 hidden layers, each with 1024 nodes, is used. The hidden layers use \( \tanh \) as the activation function and the output layer uses a linear activation function. For training, mini-batch size is set to 256 and we set a maximum number of epochs to 25 with 5 warmup epochs. Learning rate is initially set to 0.002 for warmup epochs and after that reduced by 50% with each epoch. Momentum is set to 0.3 for warmup epochs and 0.9 for all others. L2 regularization weight is set to \( 10^{-5} \). Training is done with the Merlin Toolkit [25].

For output features, we use \( \log f_0 \), 60-dimensional mel cepstral coefficients (MCCs), and 25 band aperiodicities (BAPs) at 5 ms intervals, extracted using STRAIGHT [27]. To these features, we append their respective dynamic features (deltas and delta-deltas). The \( \log f_0 \) signal is linearly interpolated through unvoiced regions and a binary voiced/unvoiced decision is appended to the acoustic feature vector. The complete output vector has a total of 259 dimensions, which are then normalized to zero mean and unit variance.

Input features to the network are derived from the labels extracted with Festival and they correspond to a set of 592 binary questions defined at phone, syllable, and word levels. These are quinphone identity, syllable stress, and guessed part-of-speech, as well as all positional features. To these questions, we append 2 features indicating phone count and state number. We let this standard feature set be the input to the baseline system. Additional models are trained under identical conditions, but append the learned representations to the baseline feature set, using a window of 3 units. That is, if using word-level features, we append the representations for previous, current, and next word. All input features are normalized to the range \([0.01, 0.99]\).

4.2. Fundamental Frequency

In this set of experiments, we learn representations at word-level using the \( f_0 \) signal. The two approaches described in Sections 2.1 and 2.2 are used. The training data contains approximately 220k word tokens. The vocabulary is defined by taking all words types that occur at least 5 times. All other words are mapped to a token \( \text{UNK} \) symbolizing out-of-vocabulary entries. This generates a set of units \( V \) with 4468 word types. With this vocabulary, we map 8.7% of the total tokens to \( \text{UNK} \).

For cluster-based representations, we set the number of DCT coefficients to 8, excluding the zeroth coefficient. Then use k-means to map the DCT coefficients at word-level into 20 clusters. An additional class accounts for silence or pause tokens. This gives us the set of acoustic classes \( A \). For mean-based representations, we set the \( f_0 \) range to be between 100Hz and 300Hz. With a bin size of 2Hz, this gives us 100 acoustic classes. To these we append 2 additional classes for any \( f_0 \)-means occurring above or below the range and 1 additional class for silence or pause tokens. Figure 1 shows an example of 4 clusters and the histogram of word means for the \( f_0 \) signal.

We set the count window \( w \) to be 3, which results in a count vector of size \( w|A| \) for each word. Note that silence or pause tokens are not in the vocabulary \( V \), but they are taken into account because \( w > 1 \). Their counts participate in the neighboring acoustic classes of in-vocabulary units. We also consider representations using both cluster and mean-based class definitions. In this case, counts are extracted and normalized separately and concatenated to form the matrix \( M \). We then find the SVD of this matrix to get the reduced matrix \( \tilde{U} \).

Table 1 details objective results using the \( f_0 \) signal and both count methods. We observe improvements with respect to the baseline with all representations based on the \( f_0 \) signal. In terms of the two proposed methods, the mean-based approach appears to outperform the cluster-based representations. Although surprisingly their interaction improves MCD, it does not perform as well as the mean-based approach on \( f_0 \) RMSE and correlation.

4.3. Energy

The type of representation proposed in this paper can exploit any type of acoustic signal. We experiment with the zeroth mel-cepstral coefficient, which may be regarded as a measure of the energy of a speech frame. Table 1 details objective results for systems appending representations learned with the zeroth mel-

Table 1: Objective results for count-based representations at word and syllable levels for \( f_0 \) and energy signals, counting over classes defined over means or clustered vectors. Dimension indicates the dimensionality of the representation on the decomposed count matrices. Input dimension denotes the dimensionality of the input features to the network, which includes a window of 3 units. MCD is mel-cepstral distortion, BAP is band aperiodicity error, and RMSE and CORR are the root-mean-squared error and correlation between predicted and original \( f_0 \) signals on voiced frames only. Systems marked with an asterisk (*) are used for a listening test.

<table>
<thead>
<tr>
<th>Unit</th>
<th>Signal</th>
<th>Classes</th>
<th>Dimension</th>
<th>Input dimension</th>
<th>MCD</th>
<th>BAP</th>
<th>F0-RMSE</th>
<th>F0-CORR</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline*</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>word</td>
<td>( f_0 )</td>
<td>clusters</td>
<td>50</td>
<td>744</td>
<td>5.688</td>
<td>2.531</td>
<td>37.629</td>
<td>0.472</td>
</tr>
<tr>
<td>word</td>
<td>( f_0 )</td>
<td>mean</td>
<td>150</td>
<td>1044</td>
<td>5.673</td>
<td>2.520</td>
<td>37.095</td>
<td>0.483</td>
</tr>
<tr>
<td>word</td>
<td>( f_0 )</td>
<td>cluster+mean</td>
<td>200</td>
<td>1194</td>
<td>5.566</td>
<td>2.515</td>
<td>37.263</td>
<td>0.473</td>
</tr>
<tr>
<td>word</td>
<td>energy</td>
<td>clusters</td>
<td>20</td>
<td>25</td>
<td>5.600</td>
<td>2.521</td>
<td>38.217</td>
<td>0.452</td>
</tr>
<tr>
<td>word</td>
<td>energy</td>
<td>mean</td>
<td>100</td>
<td>994</td>
<td>5.590</td>
<td>2.529</td>
<td>38.211</td>
<td>0.473</td>
</tr>
<tr>
<td>word</td>
<td>energy</td>
<td>cluster+mean</td>
<td>150</td>
<td>1044</td>
<td>5.680</td>
<td>2.527</td>
<td>38.245</td>
<td>0.462</td>
</tr>
<tr>
<td>word</td>
<td>( f_0 )+energy</td>
<td>cluster+mean</td>
<td>350</td>
<td>1644</td>
<td>5.637</td>
<td>2.517</td>
<td>37.194</td>
<td>0.479</td>
</tr>
<tr>
<td>syllable</td>
<td>( f_0 )</td>
<td>cluster+mean</td>
<td>180</td>
<td>1134</td>
<td>5.686</td>
<td>2.527</td>
<td>38.018</td>
<td>0.476</td>
</tr>
<tr>
<td>syllable</td>
<td>energy</td>
<td>cluster+mean</td>
<td>150</td>
<td>1044</td>
<td>5.673</td>
<td>2.525</td>
<td>38.029</td>
<td>0.474</td>
</tr>
<tr>
<td>syllable</td>
<td>( f_0 )+energy</td>
<td>cluster+mean</td>
<td>330</td>
<td>1584</td>
<td>5.645</td>
<td>2.505</td>
<td>37.264</td>
<td>0.483</td>
</tr>
<tr>
<td>word+syllable*</td>
<td>( f_0 )+energy</td>
<td>cluster+mean</td>
<td>680</td>
<td>2634</td>
<td>5.612</td>
<td>2.501</td>
<td>36.927</td>
<td>0.498</td>
</tr>
</tbody>
</table>
cepstral signal. The same details described in Section 4.2 are used for these representations. Cluster-based representations use the same hyperparameters. Mean-based representations use 80 classes over the range [3, 7] with a bin size of 0.05, and we include 3 additional classes. As before, the system combining both count approaches concatenates the normalized counts before applying SVD. For the system combining both signals (fzero+energy), we use SVD to produce two separate matrices \( \hat{U}_{\text{fzero}} \) and \( \hat{U}_{\text{energy}} \) and we simply concatenate the learned representations to the baseline system’s features.

As expected, using the zeroth mel-cepstral signal provides little improvement in terms of \( f0 \). However, quite surprisingly, it does not outperform representations based on the \( f0 \) in terms of mel-cepstral distortion. No clear difference is observed in terms of the mean and cluster-based methods, but we again observe slight improvements through their interaction. Combining both signals results in the best improvements of all representations defined at word-level.

### 4.4. Syllable-level representations

We can easily extend this approach to other types of linguistic units, such as the syllable. We represent syllable types textually as the concatenation of the phones present in a given syllable and we build \( V \) by mapping all units with fewer than 5 occurrences over the training data to the unknown token UNK. From the approximately 300k tokens, a vocabulary of 3447 unit types is defined. This maps 1.9% of the total tokens to UNK. The remaining parameters are similar to those of the word-level representations, except we vary the number of singular vectors kept after SVD. For brevity, we do not include all system combinations and we evaluate only representations using both cluster-based and mean-based approaches. In terms of combination of different counts, discretization methods were concatenated before matrix decomposition. All other levels of variation assume separate matrices, which were then added to the linguistic specification using a window of 3 textual units.

Although we observe some improvements with syllable-level representations, they do not outperform their equivalent system at word-level. As before, the interaction of both \( f0 \) and energy representations shows the best results. The system including representations at both syllable and word levels gives the best results of all configurations. Although there might be some correlation between representations, their interaction is still useful to the acoustic model.

### 5. Subjective evaluation

Given the large number of system configurations, we opt to conduct a perceptual evaluation on selected systems, based on the results shown in Table 1. Besides the baseline, we consider a system using the best combination of word-level features and a system using the best combination of word and syllable level features. The systems are marked with an asterisk on Table 1.

A preference test with a no preference option was conducted on the three selected systems. From the test set, 50 utterances were randomly selected and synthesized with the acoustic parameters generated from each system. 20 native speakers judged randomized utterance pairs for each pair of systems. Each utterance pair was judged 10 times and each condition received a total of 500 judgements. Percentage preferences are shown in Table 2, which includes the results of a 1-tailed-binomial test assuming an expected 50% split, with the no preference judgments distributed equally over the other two conditions.

The results are consistent with those in Table 1. Systems using the proposed additional features are preferred over the system using no features. Considering multiple linguistic levels (e.g., words, syllables) is preferred over using only representations learned at word level.

### 6. Discussion and conclusion

To produce the system configurations shown in Table 1, we varied three main main factors pertaining to how the representations were learned: discretization method (e.g., cluster, mean), acoustic signal (e.g. \( f0 \), energy), and linguistic level (e.g. words, syllables). In terms of the first factor, we observe that the mean-based approach outperforms the cluster-based approach, but combining both methods provides better results than either method separately. This pattern is also observed when varying the acoustic signal: \( f0 \) is shown to have a stronger impact on the objective results than energy, but the interaction of both signals gives the best results. In terms of linguistic units, we observe that word representations outperform syllable representations, but combining both levels appears more useful than either method in isolation. This was further validated with a listening tests, as shown in Table 2. In general, we observe that the more information is incorporated into the linguistic specification of a TTS system using the proposed representations, the better that TTS system performs.

With respect to the current methodology, it should be noted that no optimization of hyperparameters was attempted. No tuning was performed, for example, on the number of clusters or the number of bins for each discretization method. It was surprising to observe such improvements with fairly arbitrary initial choices of setting for these hyperparameters. Further improvements might be observed with careful optimization.

These results are encouraging and suggest further lines of research for this method. In terms of discretization methods, we might consider earlier work for the \( f0 \) signal, such as Tilt [28], MoMel [29], ProsoGram [30], or SLAM [31]. Acoustic signals such as jitter and shimmer might also be useful in the context of a TTS system. Learning representations at multiple levels might be useful, as recent work as shown that the phrase level is particularly relevant [20]. Other lines of research could focus on the speaker dependency of the proposed representations. Although there surely is some dependency, it is not known whether these representations would be useful for different speakers.

We have proposed a novel method to learn word and syllable vector representations by taking counts over discretized acoustic events. In general, we have observed that the more discretization approaches, acoustic signal, and levels of linguistic analysis are incorporated into a TTS system via these count-based representations, the better that TTS system performs.

**Acknowledgements:** This research was supported by the Swiss NSF under grant CRSII2 141903: Spoken Interaction with Interpretation in Switzerland (SIWIS), the EPSRC Standard Research Studentship (EP/K503034/1), and the EPSRC Standard Research Grant EP/P011586/1: Speech Synthesis for Spoken Content Production (SCRIPT).
7. References


