Predicting Prosodic Stress Using Recurrent Phoneme and Grapheme Embeddings

Peter Plantinga
The Ohio State University
plantinga.1@osu.edu

Abstract

Predicting prosodic stress is useful for tasks such as poetry detection, scansion, and pronunciation, as well as text-to-speech and speech-to-text. We use recurrent embeddings to represent the graphemes and phonemes of each syllable, as well as additional embeddings for the lexical stress and syllable identities. We achieve a 14-point absolute improvement over the previous state-of-the-art.

1 Introduction

Every spoken language uses patterns of pitch, duration, and loudness, that are recognizable even to someone not familiar with that language. These patterns are known as *prosody*. In some languages, certain parts of a word or phrase will be made more prominent phonetically, which is known as *lexical stress* in the case of words, and *prosodic stress* in the case of phrases or sentences. Stress is one of three important ingredients to prosody, the other two being rhythm and intonation.

While prosody and prosodic stress cannot be fully explained with a symbolic representation, one system that is often used in English is simply to mark each syllable in a phrase as either stressed or unstressed. While prosodic stress is much more nuanced than what this simple system can represent, the system is still often used for such tasks as metrical analysis of poetry.

In the field of natural language processing, little attention has been paid to the task of prosodic stress prediction. This task deserves more attention, as it has many applications, such as searching for documents that follow a poetic meter. There exist large databases of ancient texts which contain poetry, but it is cost-prohibitive to find the poetry in the database [1]. Another application could be finding poetic tweets on Twitter.

Correct stress placement is also very important for human comprehension of text-to-speech systems, as noted by Tagliapietra and Tabossi in [2]. In addition to comprehension, listeners have noted that text-to-speech systems can sound quite flat, or expressionless. Correct prediction of prosodic stress could significantly improve the quality of speech production systems.

2 Background

There are several approaches to prosodic stress prediction. Researchers in the vast field of text-to-speech have approached the problem from the angle of speech production. Hu et al. [3] use a hearer model to evaluate a stress prediction system for Chinese text-to-speech. Similarly, Sef reports on a stress prediction model using decision trees for Slovenian text-to-speech [4].

Other researchers have focused on predicting the lexical stress of individual words. Dou et al. [5] shorten each syllable of a word to just a single consonant before and after each vowel, and predict the lexical stress using an SVM ranker.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

Perhaps most similar to our work is research on automatic metrical analysis of poetry. Navarro-Colorado et al. [6] have used rule-based methods to analyze Spanish poetry. Agirrezabal et al. have used FST-based methods [7] and HMM and CRF algorithms [8] to predict the prosodic stress of a database of poetry written in English.

Since we compare against Agirrezabal et al. [8] it is worth exploring their work in more detail. In their best system, they use a CRF model, and as features they use a combination of syllable identities and 64 hand-crafted features. These features are too numerous to list but they include characters and lexical stress, as we do, plus others such as syllable phonological weight and the part-of-speech tags.

Our contributions include (i) using a neural network model to allow for deeper representations of the input before CRF decoding (ii) extracting features from syllables, lexical stress, graphemes and phonemes automatically (without the use of hand-crafted features) using embeddings, and (iii) evaluation using the same data as in Agirrezabal et al. [8] for direct comparison.

3 Task Definition

Since prosodic stress is highly correlated with lexical stress, it is worth noting the similarities and differences. The primary similarity is that prosodic stress tends to line up with lexical stress for multi-syllable words, though there are exceptions when the speaker intends to give special emphasis to certain syllables and therefore de-emphasizes others.

On the other hand, nearly every one-syllable word in CMU's pronunciation dictionary is marked as stressed, but in context, these words may or may not be stressed. Another difference is that some words in English, called *heteronyms*, have multiple pronunciations. For example, the words *project* and *affect* have different stress patterns depending on the part of speech that the word occupies.

We found that lexical stress and prosodic stress are the same for 77% of the syllables in our data. For comparison, our best model achieves 95% accuracy.

Taking a closer look at our task, here are two lines from Yeats' poem titled "He Wishes for the Cloths of Heaven" [9]:

I have spread my dreams under your feet;
Tread softly because you tread on my dreams.

While this poem loosely conforms to a poetic meter called *iambic tetrameter* where every second syllable is stressed and each line is eight syllables long, these lines deviate from the overall form of the poem.

In the first line, there are several irregularities – an extra syllable at the beginning of the line, and a reversal of the stress pattern at the word *under*:

-	-	+	-	+	+	-	-	+
I	have	spread	my	dreams	un	-der	your	feet;

In addition to other deviations from the poetic meter, the second line is ambiguous in how it might be pronounced. Some readers might choose not to emphasize the second syllable of *because* for the purpose of adding extra emphasis to the word *tread*.

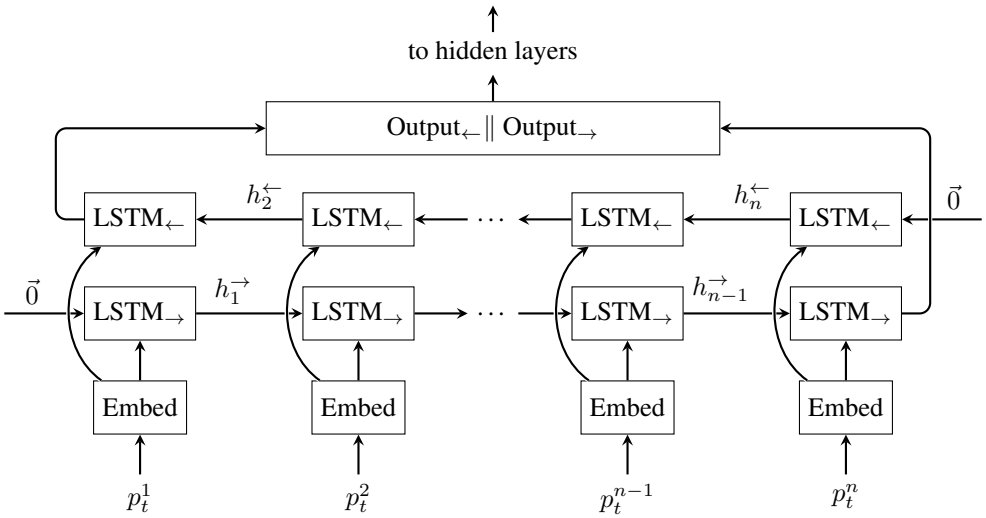
+	+	-	-	+/-	-	+	-	-	+
Tread	soft	-ly	be	-cause	you	tread	on	my	dreams.

It is quite common in poetry for the stress of a line to deviate from the poetic meter. While this makes the task of predicting prosodic stress more difficult, it also means that a system that performs well on this task may perform well more generally on text that is not poetry.

4 Neural Network Architecture

Our model for this task has three main parts described in the following sections: the input layer (including embeddings), the hidden layers, and the output layer. The full architecture can be seen in Figure 2.

Figure 1: Embedding of phoneme input (same as embedding for grapheme input) for syllable t of the input line. $p_t^1, \ldots, p_t^n$ are one-hot vectors representing all of the phonemes in the syllable. The final output of the forward and backward layers is concatenated to form the full embedding. This embedding is jointly trained with the full model.



4.1 Input Layer

As input to the network, we use 16-dimensional embedding vectors for each of four separate inputs: syllable identities, graphemes, phonemes, and lexical stress. These embeddings are jointly trained with the rest of the network.

For the syllable identities we find that using the whole case-normalized syllable is most effective, unlike Dou et al. [5] who use just the single consonant before and after the vowel. We also convert all syllable tokens that only appear once in the training data to an UNK token.

In addition to syllable identities, we embed the (un-normalized) graphemes of each syllable with a recurrent neural network, since the syllables are of different lengths. Each grapheme is represented with a 16-dimensional embedding vector, and the final output of the RNN is concatenated with the syllable identities as input to the network. For this task we use a single layer of 8 bi-directional LSTM nodes (for a final embedding size of 16), with a dropout rate of 0.4 for both the embedding vector and the recurrent layer. The architecture of the embedding is shown in Figure 1.

We also embed the phonemes of each syllable with the same architecture. The phonemes are looked up in CMU pronunciation dictionary.

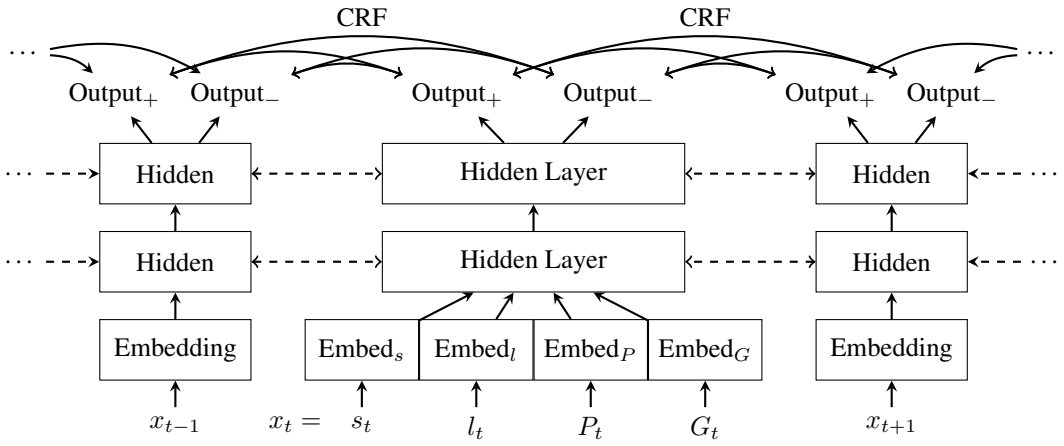
In order to predict the phonemes of out-of-vocabulary words, we use a separate model, following the model given by Toshniwal et al. in [10]. The architecture is an encoder-decoder model [11] with global attention [12]. We use three layers of 512 bi-directional LSTM nodes for the encoder model, and concatenate the last hidden state of the topmost forward and backward layers to a vector which is copied to all decoder layers as the initial hidden state.

Using this model to predict only the phonemes, we achieve a phoneme error rate of 4.9 and word accuracy of 79.8% on the development set, which is comparable to Toshniwal et al. However, the CMU pronunciation dictionary also provides lexical stress, which is clearly beneficial for this task. In order to align lexical stress with syllables, we predict a sequence of both phones and lexical stress, of the same format as found in CMU dict. For example:

$$\text{speculating} \Rightarrow \text{s p eh 1 k y ah 0 l ey 2 t ih 0 ng}$$

In this example, "1" represents primary stress, "2" represents secondary stress, and "0" represents an unstressed syllable. Our prediction model for this joint prediction task achieves an error rate of

Figure 2: Neural network architecture for syllable t. x_t is the input to the hidden layers, and is composed of embeddings e_t for syllables, lexical stress, phonemes, and graphemes. The inputs are one-hot vectors, represented by s_t, l_t, $p_t^1, \ldots, p_t^n$, $g_t^1, \ldots, g_t^m$. In the case of feed-forward hidden layers, the only time syllable context is considered is in CRF decoding. In the case of recurrent hidden layers, however, the hidden state is passed to neighboring syllables.



5.9 and a word accuracy of 74.1% on this combined data. Given that the task is more difficult than simply predicting phones, this is a reasonable result.

The phoneme and stress sequence can be broken down further into syllables using the grapheme syllable representations. We count the number of consonants after the first vowel in each grapheme syllable, and append the same number of consonants to the end of the phoneme syllable, taking into account a few common grapheme pairs that only have one phoneme, such as *ng* and *ck*.

Lexical stress:	1	0	2	0
Phonemes:	s p eh k	y ah	l ey	t ih ng
Graphemes:	spec	-u	-la	-ting

Since there are only three symbols for lexical stress, it is not obvious that an embedding layer would perform better than a one-hot encoding, as our experiments have shown. We hypothesize that this boost is due to two facts. First, in a one-hot encoding, dropout will turn off an entire input instead of just a portion of the input. Second, a sixteen dimensional vector puts the input on more even footing with the other inputs (which is why sixteen dimensions does better than ten dimensions).

4.2 Hidden Layers

We experiment with both feed-forward and recurrent layers for the hidden layers. In both models, we use a dropout rate of 0.4, which performs the best empirically. While this value is higher than most reported dropout rates, we hypothesize that the large rate performs well due to a small dataset (and thus a need for heavy regularization).

In both models, we use two hidden layers with 100 nodes each. In the feed-forward model, we use the ReLU activation function. Our recurrent model uses bi-directional LSTM nodes, and the output of the topmost forward and backward layers is concatenated at every timestep as input to the output layer, for a total of 200 dimensions.

4.3 Output Layer

For output, we use a single fully connected layer with two outputs for every syllable, with 0.2 as a dropout rate. Then a CRF decoder is applied which picks the optimal sequence of outputs. Using only the output layer is a similar model to the CRF model used in Agirrezabal et al. in [8], with the addition of dropout and conversion of tokens that appear only once to UNK tokens.

Table 1: Accuracy of systems using just syllables as input. Each range indicates one standard deviatiation over three trials.

Model	Stress Error Rate	Line Accuracy
Agirrezabal et al. HMM [8]	9.57	49.9
Agirrezabal et al. CRF [8]	11.87	43.9
Output layer + CRF		
One-hot syllables	10.49 ± 0.10	48.1 ± 0.3
Embedded syllables	9.78 ± 0.04	49.1 ± 0.9
Feed-forward hidden layers		
One-hot syllables	9.62 ± 0.08	50.9 ± 0.3
Embedded syllables	9.72 ± 0.18	51.6 ± 0.6
Recurrent hidden layers		
One-hot syllables	9.45 ± 0.34	51.6 ± 1.4
Embedded syllables	8.27 ± 0.18	56.9 ± 0.8

The neural network architecture found in this paper is implemented with TensorFlow [13]. We use a batch size of 32 and a learning rate of 0.002 (decreasing by 30% each time the line accuracy decreases). Also, we train for 50 epochs.

5 Experiments

In order to test the effectiveness of our system, we train using the same data as in Agirrezabal et al. [8] and directly compare accuracy as computed by using 10-fold cross-validation.

5.1 Data

The data used for these experiments was made publicly available by the authors of the "For Better For Verse" website, from the University of Virginia [14]. They have manually annotated a set of 80 poems containing about 1,100 lines of text for the purpose of training students in metrical analysis.

Each line is annotated with both a poetic meter (such as iambic pentameter) and a mark for each syllable whether it is stressed or unstressed. There are sometimes several possible analyses given, and as in previous work, we take the minimum Levenshtein distance between the prediction and any analysis to be the error rate for the line.

5.2 Input: Only Syllables

In our first experiment, we test the effectiveness of different hidden layer configurations. We use as input only the syllable identities.

First, by using only the output layer, we imitate the CRF model used in Agirrezabal et al. [8], and achieve a small improvement due to dropout in the output layer and converting all tokens that appear only once in the training data to UNK tokens.

Second, we add feed-forward and recurrent hidden layers to test their effectiveness at representing this data. The results of all systems can be seen in Table 1. The two models give similar results on the one-hot encoded data, but adding an embedding layer is much more helpful for the recurrent network than the feed-forward network or the output layer.

5.3 Additional Input: Graphemes, Phonemes, and Lexical Stress

For the purpose of exploring the effect of these three additional inputs, we provided each of the three inputs independently to the model, along with the syllable identites.

The phonemes provide the least information to the model. Most of the information provided by phonemes is also included in the graphemes, but graphemes include additional information such as

Table 2: Accuracy of systems using each input independently. Each range indicates one standard deviation over five trials.

Model	Stress Error Rate	Line Accuracy
Feed-forward hidden layers		
syllable embeddings	9.72 ± 0.18	51.6 ± 0.6
syllables + phonemes	7.72 ± 0.11	58.8 ± 0.7
syllables + graphemes	7.42 ± 0.21	61.0 ± 0.8
syllables + lexical stress	6.64 ± 0.24	62.3 ± 1.0
Recurrent hidden layers		
syllable embeddings	8.27 ± 0.18	56.9 ± 0.8
syllables + phonemes	6.65 ± 0.13	62.9 ± 0.7
syllables + graphemes	6.11 ± 0.06	65.7 ± 0.4
syllables + lexical stress	5.44 ± 0.16	68.6 ± 1.0

capitalization and punctuation. Unsurprisingly, lexical stress is the most informative for the task of predicting prosadic stress.

The comparison between inputs can be seen in Table 2.

5.4 Full Model

For our final experiment, we add the feature sets one at a time to the model in order from most informative to least informative, to assess whether each part is contributing to the final score.

We see that the recurrent model is consistently 5-6 points higher than the fully-connected model, and all systems seem to gain per-line accuracy from both the phoneme and grapheme embeddings. However, there is no appreciable gain in the stress error rate from the phoneme embeddings. This is likely due to the fact that the phonemes hold very little information per-syllable, except for what can already be found in the graphemes. The results of the models with all inputs can be seen in Table 3.

6 Error Analysis

We manually evaluated lines with a large number of errors (more than 2) to find systematic biases. One bias that we found was that our system tends to overpredict alternating stressed and unstressed syllables, due to a majority of the training data following this pattern. This is especially a problem for lines in a non-alternating poetic meter. Here is an example:

Predicted:	+	-	-	+	+	-	+	-	+	-	-	+
Actual:	+	-	-	+	-	-	+	-	-	+	-	-
Syllables:	Ev	-er	to	come	up	with	Dac	-tyl	tri	-syl	-a	-ble

The system predicts three sets of alternating stress, whereas the label contains none.

Another error that is common in the data is overly long sequences of stressed or unstressed syllables. For example:

Predicted:	+	+	-	+	-	-	+	+	+	+	+	-
Actual:	-	+	-	-	-	+	+	-	+	-	+	-
Syllables:	No	won	-der	of	it:	sheer	plod	makes	plough	down	sill	-ion

While each individual syllable in the sequence of predicted stressed syllables is likely to be stressed in other contexts, since there is a long sequence of them, the speaker is unlikely to stress all of the syllables.

The last example is one where part-of-speech tagging with additional context may improve the model. Just looking at this line, the last word seems to be marked incorrectly.

Table 3: For this experiment, we add embeddings one at a time in order of effectiveness as judged by the results of the previous experiment, as seen in Table 2. Each range indicates one standard deviation over five trials.

Model	Stress Error Rate	Line Accuracy
Agirrezabal et al. CRF		
syllables	11.87	43.9
+ 64 features	8.59	55.3
Feed-forward hidden layers		
syllable embeddings	9.72 ± 0.18	51.6 ± 0.6
+ lexical stress	6.64 ± 0.24	62.3 ± 1.0
+ graphemes	6.19 ± 0.06	64.9 ± 0.2
+ phonemes	6.19 ± 0.06	65.1 ± 0.4
Recurrent hidden layers		
syllable embeddings	8.27 ± 0.18	56.9 ± 0.8
+ lexical stress	5.44 ± 0.16	68.6 ± 1.0
+ graphemes	5.21 ± 0.08	69.2 ± 0.6
+ phonemes	5.07 ± 0.12	69.8 ± 0.7

Predicted:	-	+	+	-	-	+	-	-	+	-
Actual:	-	+	-	+	-	+	-	+	-	+
Syllables:	To	serve	there	with	my	Mak	-er	and	pre	-sent

But with the additional information from the next line, we gather that the word *present* is a verb, not a noun, and should be pronounced with the emphasis on the second syllable. Here is the line in context:

To serve there with my Maker and present
My true account, lest he returning chide;

7 Conclusion

We have presented a model for predicting the prosodic stress of text, achieving a 14-point gain over previous methods. Our approach has the added benefit of extracting all features automatically through embedding of the graphemes, phonemes, and lexical stress. The only inputs required for this model to work are lines of text with corresponding prosodic stress markings, and a pronunciation dictionary with phonemes and lexical stress markings.

One limitation of this study is the small amount of data available with which to train the model. A larger dataset would undoubtedly provide further gains in accuracy. In addition, data that is not poetry would prove that this method could be used in many contexts. It is possible that ToBI-labeled data could be used for this purpose [15].

Finally, given the lack of appropriate data, we intend to pursue a semi-supervised approach, to leverage larger corpora.

Acknowledgements

Thanks to Guillaume Genthial for his tutorial on how to make a sequence tagging model in TensorFlow, and to Matvey Ezhov for his tutorial and code for a sequence-to-sequence model in TensorFlow.

References

[1] Bamman, D. & Smith, D. (2012) Extracting Two Thousand Years of Latin from a Million Book Library. *Journal on Computing and Cultural Heritage (JOOCH)*, pp. 2.

[2] Tagliapietra, L. & Tabossi, P. (2005) Lexical stress effects in Italian spoken word recognition. In *The XXVII Annual Conference of the Cognitive Science Society*, pp. 2140–2144.

[3] Hu, G.P., Liu, Q.F., Hu, Y. & Wang, R.H. (2004) Hearer model based stress prediction for Chinese TTS system. *Proceedings of ISCSLP 2004, the International Symposium on Chinese Spoken Language Processing*, pp. 161–164.

[4] Sef, T. (2004) Lexical stress assignment model for the Slovenian text-to-speech synthesis system. *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video, and Speech Processing*, pp. 683–686.

[5] Dou, Q., Bergsma, S., Jiampojamarn, S. & Kondrak, G. (2009) A ranking approach to stress prediction for letter-to-phoneme conversion. *Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP*, pp. 118–126.

[6] Navarro-Colorado, B. (2015) A computational linguistic approach to Spanish Golden Age Sonnets: metrical and semantic aspects. *Proceedings of NAACL-HLT Fourth Workshop on Computational Linguistics for Literature*, pp. 105–113.

[7] Agirrezabal, M., Arrieta, B., Astigarraga, A. & Hulden, M. (2013) ZeuScansion: a tool for scansion of English poetry. *Proceedings of the 11th International Conference on Finite State Methods and Natural Language Processing*, pp. 18–24.

[8] Agirrezabal, M., Alegria, I. & Hulden, M. (2016) Machine learning for metrical analysis of English poetry. *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 772–781.

[9] Yeats, W.B. (1899) He wishes for the cloths of heaven.

[10] Toshniwal, S. & Livescu, K. (2016) Jointly learning to align and convert graphemes to phonemes with neural attention models. *arXiv preprint*, `http://arXiv.org/abs/1610.06540`

[11] Cho, K., Bart, M., Gulchere, C., Bahndanau, D., Bougares, F., Schwenk, H. & Bengio, Y. (2014) Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734.

[12] Cheng, J., Dong, L. & Lapata, M. (2016) Long Short-Term Memory-Networks for Machine Reading *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 551–561.

[13] Abadi, M. et al. (2015) TensorFlow: Large-scale machine learning on heterogeneous systems.

[14] Tucker, H. (2011) Poetic data and the news from poems: A For Better For Verse memoir. *Victorian Poetry*, pp. 267–281.

[15] Beckman, M.E. & Ayers, G. (1997) Guidelines for ToBI labelling. *The OSU Research Foundation*.