A Ranking Approach to Stress Prediction for Letter-to-Phoneme Conversion

Qing Dou, Shane Bergsma, Sittichai Jiampojamarn and Grzegorz Kondrak
Department of Computing Science
University of Alberta
Edmonton, AB, T6G 2E8, Canada
{qdou,bergsma,sj,kondrak}@cs.ualberta.ca

Abstract

Correct stress placement is important in text-to-speech systems, in terms of both the overall accuracy and the naturalness of pronunciation. In this paper, we formulate stress assignment as a sequence prediction problem. We represent words as sequences of substrings, and use the substrings as features in a Support Vector Machine (SVM) ranker, which is trained to rank possible stress patterns. The ranking approach facilitates inclusion of arbitrary features over both the input sequence and output stress pattern. Our system advances the current state-of-the-art, predicting primary stress in English, German, and Dutch with up to 98% word accuracy on phonemes, and 96% on letters. The system is also highly accurate in predicting secondary stress. Finally, when applied in tandem with an L2P system, it substantially reduces the word error rate when predicting both phonemes and stress.

1 Introduction

In many languages, certain syllables in words are phonetically more prominent in terms of duration, pitch, and loudness. This phenomenon is referred to as *lexical stress*. In some languages, the location of stress is entirely predictable. For example, lexical stress regularly falls on the initial syllable in Hungarian, and on the penultimate syllable in Polish. In other languages, such as English and Russian, any syllable in the word can be stressed.

Correct stress placement is important in text-to-speech systems because it affects the accuracy of human word recognition (Tagliapietra and Tabossi, 2005; Arciuli and Cupples, 2006). However, the issue has often been ignored in previous letter-to-phoneme (L2P) systems. The systems that do generate stress markers often do not report separate figures on stress prediction accuracy, or they only provide results on a single language. Some only predict primary stress markers (Black et al., 1998; Webster, 2004; Demberg et al., 2007), while those that predict both primary and secondary stress generally achieve lower accuracy (Bagshaw, 1998; Coleman, 2000; Pearson et al., 2000).

In this paper, we formulate stress assignment as a sequence prediction problem. We divide each word into a sequence of substrings, and use these substrings as features for a Support Vector Machine (SVM) ranker. For a given sequence length, there is typically only a small number of stress patterns in use. The task of the SVM is to rank the true stress pattern above the small number of acceptable alternatives. This is the first system to predict stress within a powerful discriminative learning framework. By using a ranking approach, we enable the use of arbitrary features over the entire (input) sequence and (output) stress pattern. We show that the addition of a feature for the entire output sequence improves prediction accuracy.

Our experiments on English, German, and Dutch demonstrate that our ranking approach substantially outperforms previous systems. The SVM ranker achieves exceptional 96.2% word accuracy on the challenging task of predicting the full stress pattern in English. Moreover, when combining our stress predictions with a state-of-the-art L2P system (Jiampojamarn et al., 2008), we set a new standard for the combined prediction of phonemes and stress.

The paper is organized as follows. Section 2 provides background on lexical stress and a task definition. Section 3 presents our automatic stress prediction algorithm. In Section 4, we confirm the power of the discriminative approach with experiments on three languages. Section 5 describes how stress is integrated into L2P conversion.

Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pages 118–126,
Suntec, Singapore, 2-7 August 2009.

2 Background and Task Definition

There is a long history of research into the principles governing lexical stress placement. Zipf (1929) showed that stressed syllables are often those with low frequency in speech, while unstressed syllables are usually very common. Chomsky and Halle (1968) proposed a set of context-sensitive rules for producing English stress from underlying word forms. Due to its importance in text-to-speech, there is also a long history of computational stress prediction systems (Fudge, 1984; Church, 1985; Williams, 1987). While these early approaches depend on human definitions of vowel tensity, syllable weight, word etymology, etc., our work follows a recent trend of purely data-driven approaches to stress prediction (Black et al., 1998; Pearson et al., 2000; Webster, 2004; Demberg et al., 2007).

In many languages, only two levels of stress are distinguished: stressed and unstressed. However, some languages exhibit more than two levels of stress. For example, in the English word *economic*, the first and the third syllable are stressed, with the former receiving weaker emphasis than the latter. In this case, the initial syllable is said to carry a secondary stress. Although each word has only one primary stress, it may have any number of secondary stresses. Predicting the full stress pattern is therefore inherently more difficult than predicting the location of primary stress only.

Our objective is to automatically assign primary and, where possible, secondary stress to out-of-vocabulary words. Stress is an attribute of syllables, but syllabification is a non-trivial task in itself (Bartlett et al., 2008). Rather than assuming correct syllabification of the input word, we instead follow Webster (2004) in placing the stress on the vowel which constitutes the nucleus of the stressed syllable. If the syllable boundaries are known, the mapping from the vowel to the corresponding syllable is straightforward.

We investigate the assignment of stress to two related but different entities: the spoken word (represented by its phonetic transcription), and the written word (represented by its orthographic form). Although stress is a prosodic feature, assigning stress to written words ("stressed orthography") has been utilized as a preprocessing stage for the L2P task (Webster, 2004). This preprocessing is motivated by two factors. First, stress greatly influences the pronunciation of vowels in English (c.f., *allow* vs. *alloy*). Second, since phoneme predictors typically utilize only local context around a letter, they do not incorporate the global, long-range information that is especially predictive of stress, such as penultimate syllable emphasis associated with the suffix *-ation*. By taking stressed orthography as input, the L2P system is able to implicitly leverage morphological information beyond the local context.

Indicating stress on letters can also be helpful to humans, especially second-language learners. In some languages, such as Spanish, orthographic markers are obligatory in words with irregular stress. The location of stress is often explicitly marked in textbooks for students of Russian. In both languages, the standard method of indicating stress is to place an acute accent above the vowel bearing primary stress, e.g., *adiós*. The secondary stress in English can be indicated with a grave accent (Coleman, 2000), e.g., *prècéde*.

In summary, our task is to assign primary and secondary stress markers to stress-bearing vowels in an input word. The input word may be either phonemes or letters. If a stressed vowel is represented by more than one letter, we adopt the convention of marking the first vowel of the vowel sequence, e.g., *méeting*. In this way, we are able to focus on the task of stress prediction, without having to determine at the same time the exact syllable boundaries, or whether a vowel letter sequence represents one or more spoken vowels (e.g., *beating* vs. *be-at-i-fy*).

3 Automatic Stress Prediction

Our stress assignment system maps a word, $\mathbf{w}$, to a stressed-form of the word, $\bar{\mathbf{w}}$. We formulate stress assignment as a sequence prediction problem. The assignment is made in three stages:

(1) First, we map words to substrings ($\mathbf{s}$), the basic units in our sequence (Section 3.1).

(2) Then, a particular stress pattern ($\mathbf{t}$) is chosen for each substring sequence. We use a support vector machine (SVM) to rank the possible patterns for each sequence (Section 3.2).

(3) Finally, the stress pattern is used to produce the stressed-form of the word (Section 3.3).

Table 1 gives examples of words at each stage of the algorithm. We discuss each step in more detail.

Word		Substrings		Pattern		Word'
$\mathbf{w}$	$\rightarrow$	$\mathbf{s}$	$\rightarrow$	$\mathbf{t}$	$\rightarrow$	$\bar{\mathbf{w}}$
worker	$\rightarrow$	*wor-ker*	$\rightarrow$	1-0	$\rightarrow$	*wórker*
overdo	$\rightarrow$	*ov-ver-do*	$\rightarrow$	2-0-1	$\rightarrow$	*òverdó*
react	$\rightarrow$	*re-ac*	$\rightarrow$	0-1	$\rightarrow$	*reáct*
æbstrækt	$\rightarrow$	æb-træk	$\rightarrow$	0-1	$\rightarrow$	æbstrækt
prisid	$\rightarrow$	ri-sid	$\rightarrow$	2-1	$\rightarrow$	prìsíd

Table 1: The steps in our stress prediction system (with orthographic and phonetic prediction examples): (1) word splitting, (2) support vector ranking of stress patterns, and (3) pattern-to-vowel mapping.

3.1 Word Splitting

The first step in our approach is to represent the word as a sequence of N individual units: $\mathbf{w} \rightarrow \mathbf{s} = \{s_1\text{-}s_2\text{-...-}s_N\}$. These units are used to define the features and outputs used by the SVM ranker. Although we are ultimately interested in assigning stress to individual vowels in the phoneme and letter sequence, it is beneficial to represent the task in units larger than individual letters.

Our substrings are similar to syllables; they have a vowel as their nucleus and include consonant context. By approximating syllables, our substring patterns will allow us to learn recurrent stress regularities, as well as dependencies between neighboring substrings. Since determining syllable breaks is a non-trivial task, we instead adopt the following simple splitting technique. Each vowel in the word forms the nucleus of a substring. Any single preceding or following consonant is added to the substring unit. Thus, each substring consists of at most three symbols (Table 1).

Using shorter substrings reduces the sparsity of our training data; words like *cryer*, *dryer* and *fryer* are all mapped to the same form: *ry-er*. The SVM can thus generalize from observed words to similarly-spelled, unseen examples.

Since the number of vowels equals the number of syllables in the phonetic form of the word, applying this approach to phonemes will always generate the correct number of syllables. For letters, splitting may result in a different number of units than the true syllabification, e.g., *pronounce* $\rightarrow$ *ron-no-un-ce*. This does not prevent the system from producing the correct stress assignment after the pattern-to-vowel mapping stage (Section 3.3) is complete.

3.2 Stress Prediction with SVM Ranking

After creating a sequence of substring units, $\mathbf{s} = \{s_1\text{-}s_2\text{-...-}s_N\}$, the next step is to choose an output sequence, $\mathbf{t} = \{t_1\text{-}t_2\text{-...-}t_N\}$, that encodes whether each unit is stressed or unstressed. We use the number '1' to indicate that a substring receives primary stress, '2' for secondary stress, and '0' to indicate no stress. We call this output sequence the *stress pattern* for a word. Table 1 gives examples of words, substrings, and stress patterns.

We use supervised learning to train a system to predict the stress pattern. We generate training $(\mathbf{s}, \mathbf{t})$ pairs in the obvious way from our stress-marked training words, $\bar{\mathbf{w}}$. That is, we first extract the letter/phoneme portion, $\mathbf{w}$, and use it to create the substrings, $\mathbf{s}$. We then create the stress pattern, $\mathbf{t}$, using $\bar{\mathbf{w}}$'s stress markers. Given the training pairs, any sequence predictor can be used, for example a Conditional Random Field (CRF) (Lafferty et al., 2001) or a structured perceptron (Collins, 2002). However, we can take advantage of a unique property of our problem to use a more expressive framework than is typically used in sequence prediction.

The key observation is that the output space of possible stress patterns is actually fairly limited. Clopper (2002) shows that people have strong preferences for particular sequences of stress, and this is confirmed by our training data (Section 4.1). In English, for example, we find that for each set of spoken words with the same number of syllables, there are no more than fifteen different stress patterns. In total, among 55K English training examples, there are only 70 different stress patterns. In both German and Dutch there are only about 50 patterns in 250K examples.[1] Therefore, for a particular input sequence, we can safely limit our consideration to only the small set of output patterns of the same length.

Thus, unlike typical sequence predictors, we do not have to search for the highest-scoring output according to our model. We can enumerate the full set of outputs and simply choose the highest-scoring one. This enables a more expressive representation. We can define arbitrary features over the entire output sequence. In a typical CRF or structured perceptron approach, only output features that can be computed incrementally during search are used (e.g. Markov transition features that permit Viterbi search). Since search is not

[1] See (Dou, 2009) for more details.

needed here, we can exploit longer-range features.

Choosing the highest-scoring output from a fixed set is a ranking problem, and we provide the full ranking formulation below. Unlike previous ranking approaches (e.g. Collins and Koo (2005)), we do not rely on a generative model to produce a list of candidates. Candidates are chosen in advance from observed training patterns.

3.2.1 Ranking Formulation

For a substring sequence, $\mathbf{s}$, of length N, our task is to select the correct output pattern from the set of all length-N patterns observed in our training data, a set we denote as $\mathbf{T}_N$. We score each possible input-output combination using a linear model. Each substring sequence and possible output pattern, $(\mathbf{s}, \mathbf{t})$, is represented with a set of features, $\mathbf{\Phi}(\mathbf{s}, \mathbf{t})$. The score for a particular $(\mathbf{s}, \mathbf{t})$ combination is a weighted sum of these features, $\boldsymbol{\lambda} \cdot \mathbf{\Phi}(\mathbf{s}, \mathbf{t})$. The specific features we use are described in Section 3.2.2.

Let $\mathbf{t}^j$ be the stress pattern for the jth training sequence $\mathbf{s}^j$, both of length N. At training time, the weights, $\boldsymbol{\lambda}$, are chosen such that for each $\mathbf{s}^j$, the correct output pattern receives a higher score than other patterns of the same length: $\forall \mathbf{u} \in \mathbf{T}_N, \mathbf{u} \neq \mathbf{t}^j$,

$$\boldsymbol{\lambda} \cdot \mathbf{\Phi}(\mathbf{s}^j, \mathbf{t}^j) > \boldsymbol{\lambda} \cdot \mathbf{\Phi}(\mathbf{s}^j, \mathbf{u}) \qquad (1)$$

The set of constraints generated by Equation 1 are called *rank constraints*. They are created separately for every $(\mathbf{s}^j, \mathbf{t}^j)$ training pair. Essentially, each training pair is matched with a set of automatically-created negative examples. Each negative has an incorrect, but plausible, stress pattern, $\mathbf{u}$.

We adopt a Support Vector Machine (SVM) solution to these ranking constraints as described by Joachims (2002). The learner finds the weights that ensure a maximum (soft) margin separation between the correct scores and the competitors. We use an SVM because it has been successful in similar settings (learning with thousands of sparse features) for both ranking and classification tasks, and because an efficient implementation is available (Joachims, 1999).

At test time we simply score each possible output pattern using the learned weights. That is, for an input sequence $\mathbf{s}$ of length N, we compute $\boldsymbol{\lambda} \cdot \mathbf{\Phi}(\mathbf{s}, \mathbf{t})$ for all $\mathbf{t} \in \mathbf{T}_N$, and we take the highest scoring $\mathbf{t}$ as our output. Note that because we only consider previously-observed output patterns, it is impossible for our system to produce a nonsensical result, such as having two primary stresses in one word. Standard search-based sequence predictors need to be specially augmented with hard constraints in order to prevent such output (Roth and Yih, 2005).

Substring	s_i, t_i
	s_i, i, t_i
Context	s_{i-1}, t_i
	$s_{i-1}s_i, t_i$
	s_{i+1}, t_i
	$s_i s_{i+1}, t_i$
	$s_{i-1}s_i s_{i+1}, t_i$
Stress Pattern	$t_1 t_2 \ldots t_N$

Table 2: Feature Template

3.2.2 Features

The power of our ranker to identify the correct stress pattern depends on how expressive our features are. Table 2 shows the feature templates used to create the features $\mathbf{\Phi}(\mathbf{s}, \mathbf{t})$ for our ranker. We use binary features to indicate whether each combination occurs in the current $(\mathbf{s},\mathbf{t})$ pair.

For example, if a substring *tion* is unstressed in a $(\mathbf{s}, \mathbf{t})$ pair, the *Substring* feature $\{s_i, t_i = \textit{tion}, 0\}$ will be true.[2] In English, often the penultimate syllable is stressed if the final syllable is *tion*. We can capture such a regularity with the *Context* feature s_{i+1}, t_i. If the following syllable is *tion* and the current syllable is stressed, the feature $\{s_{i+1}, t_i = \textit{tion}, 1\}$ will be true. This feature will likely receive a positive weight, so that output sequences with a stress before *tion* receive a higher rank.

Finally, the full *Stress Pattern* serves as an important feature. Note that such a feature would not be possible in standard sequence predictors, where such information must be decomposed into Markov transition features like $t_{i-1}t_i$. In a ranking framework, we can score output sequences using their full output pattern. Thus we can easily learn the rules in languages with regular stress rules. For languages that do not have a fixed stress rule, preferences for particular patterns can be learned using this feature.

[2] *tion* is a substring composed of three phonemes but we use its orthographic representation here for clarity.

3.3 Pattern-to-Vowel Mapping

The final stage of our system uses the predicted pattern $\mathbf{t}$ to create the stress-marked form of the word, $\bar{\mathbf{w}}$. Note the number of substrings created by our splitting method always equals the number of vowels in the word. We can thus simply map the indicator numbers in $\mathbf{t}$ to markers on their corresponding vowels to produce the stressed word.

For our example, *pronounce* → *ron-no-un-ce*, if the SVM chooses the stress pattern, 0-1-0-0, we produce the correct stress-marked word, *pronóunce*. If we instead stress the third vowel, 0-0-1-0, we produce an incorrect output, *pronoúnce*.

4 Stress Prediction Experiments

In this section, we evaluate our ranking approach to stress prediction by assigning stress to spoken and written words in three languages: English, German, and Dutch. We first describe the data and the various systems we evaluate, and then provide the results.

4.1 Data

The data is extracted from CELEX (Baayen et al., 1996). Following previous work on stress prediction, we randomly partition the data into 85% for training, 5% for development, and 10% for testing. To make results on German and Dutch comparable with English, we reduce the training, development, and testing set by 80% for each. After removing all duplicated items as well as abbreviations, phrases, and diacritics, each training set contains around 55K words.

In CELEX, stress is labeled on syllables in the phonetic form of the words. Since our objective is to assign stress markers to *vowels* (as described in Section 2) we automatically map the stress markers from the stressed syllables in the phonetic forms onto phonemes and letters representing vowels. For phonemes, the process is straightforward: we move the stress marker from the beginning of a syllable to the phoneme which constitutes the nucleus of the syllable. For letters, we map the stress from the vowel phoneme onto the orthographic forms using the ALINE algorithm (Dwyer and Kondrak, 2009). The stress marker is placed on the first letter within the syllable that represents a vowel sound.[3]

[3]Our stand-off stress annotations for English, German, and Dutch CELEX orthographic data can be downloaded at: http://www.cs.ualberta.ca/~kondrak/celex.html.

System	Eng		Ger	Dut
	P+*S*	*P*	*P*	*P*
SUBSTRING	96.2	98.0	97.1	93.1
ORACLESYL	95.4	96.4	97.1	93.2
TOPPATTERN	66.8	68.9	64.1	60.8

Table 3: Stress prediction word accuracy (%) on **phonemes** for English, German, and Dutch. *P*: predicting primary stress only. *P*+*S*: primary and secondary.

CELEX also provides secondary stress annotation for English. We therefore evaluate on both primary and secondary stress (*P*+*S*) in English and on primary stress assignment alone (*P*) for English, German, and Dutch.

4.2 Comparison Approaches

We evaluate three different systems on the letter and phoneme sequences in the experimental data:

1) SUBSTRING is the system presented in Section 3. It uses the vowel-based splitting method, followed by SVM ranking.

2) ORACLESYL splits the input word into syllables according to the CELEX gold-standard, before applying SVM ranking. The output pattern is evaluated directly against the gold-standard, without pattern-to-vowel mapping.

3) TOPPATTERN is our baseline system. It uses the vowel-based splitting method to produce a substring sequence of length N. Then it simply chooses the most common stress pattern among all the stress patterns of length N.

SUBSTRING and ORACLESYL use scores produced by an SVM ranker trained on the training data. We employ the ranking mode of the popular learning package SVM^{light} (Joachims, 1999). In each case, we learn a linear kernel ranker on the training set stress patterns and tune the parameter that trades-off training error and margin on the development set.

We evaluate the systems using *word accuracy*: the percent of words for which the output form of the word, $\bar{\mathbf{w}}$, matches the gold standard.

4.3 Results

Table 3 provides results on English, German, and Dutch phonemes. Overall, the performance of our automatic stress predictor, SUBSTRING, is excellent. It achieves 98.0% accuracy for predicting

System	Eng		Ger	Dut
	P+S	*P*	*P*	*P*
SUBSTRING	93.5	95.1	95.9	91.0
ORACLESYL	94.6	96.0	96.6	92.8
TOPPATTERN	65.5	67.6	64.1	60.8

Table 4: Stress prediction word accuracy (%) on **letters** for English, German, and Dutch. *P*: predicting primary stress only. *P+S*: primary and secondary.

primary stress in English, 97.1% in German, and 93.1% in Dutch. It also predicts both primary and secondary stress in English with high accuracy, 96.2%. Performance is much higher than our baseline accuracy, which is between 60% and 70%. ORACLESYL, with longer substrings and hence sparser data, does not generally improve performance. This indicates that perfect syllabification is unnecessary for phonetic stress assignment.

Our system is a major advance over the previous state-of-the-art in phonetic stress assignment. For predicting stressed/unstressed syllables in English, Black et al. (1998) obtained a per-syllable accuracy of 94.6%. We achieve 96.2% *per-word* accuracy for predicting both primary and secondary stress. Others report lower numbers on English phonemes. Bagshaw (1998) obtained 65%-83.3% per-syllable accuracy using Church (1985)'s rule-based system. For predicting both primary and secondary stress, Coleman (2000) and Pearson et al. (2000) report 69.8% and 81.0% word accuracy, respectively.

The performance on letters (Table 4) is also quite encouraging. SUBSTRING predicts primary stress with accuracy above 95% for English and German, and equal to 91% in Dutch. Performance is 1-3% lower on letters than on phonemes. On the other hand, the performance of ORACLESYL drops much less on letters. This indicates that most of SUBSTRING's errors are caused by the splitting method. Letter vowels may or may not represent spoken vowels. By creating a substring for every vowel letter we may produce an incorrect number of syllables. Our pattern feature is therefore less effective.

Nevertheless, SUBSTRING's accuracy on letters also represents a clear improvement over previous work. Webster (2004) reports 80.3% word accuracy on letters in English and 81.2% in German. The most comparable work is Demberg et al. (2007), which achieves 90.1% word accuracy on letters in German CELEX, assuming perfect letter syllabification. In order to reproduce their strict experimental setup, we re-partition the full set of German CELEX data to ensure that no overlap of word stems exists between the training and test sets. Using the new data sets, our system achieves a word accuracy of 92.3%, a 2.2% improvement over Demberg et al. (2007)'s result. Moreover, if we also assume perfect syllabification, the accuracy is 94.3%, a 40% reduction in error rate.

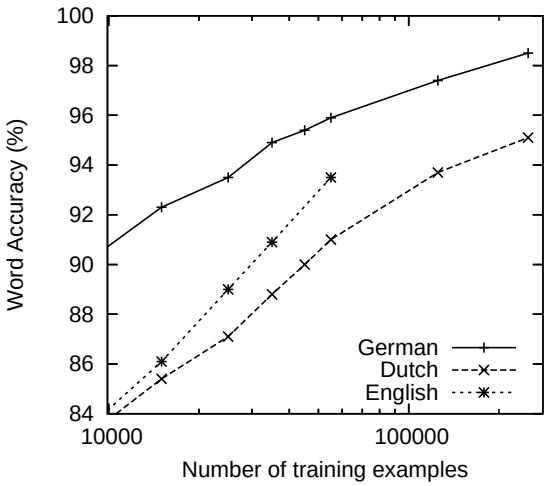

Figure 1: Stress prediction accuracy on letters.

We performed a detailed analysis to understand the strong performance of our system. First of all, note that an error could happen if a test-set stress pattern was not observed in the training data; its correct stress pattern would not be considered as an output. In fact, no more than two test errors in any test set were so caused. This strongly justifies the reduced set of outputs used in our ranking formulation.

We also tested all systems with the Stress Pattern feature removed. Results were worse in all cases. As expected, it is most valuable for predicting primary and secondary stress. On English phonemes, accuracy drops from 96.2% to 95.3% without it. On letters, it drops from 93.5% to 90.0%. The gain from this feature also validates our ranking framework, as such arbitrary features over the entire output sequence can not be used in standard search-based sequence prediction.

Finally, we examined the relationship between training data size and performance by plotting learning curves for letter stress accuracy (Figure 1). Unlike the tables above, here we use the

full set of data in Dutch and German CELEX to create the largest-possible training sets (255K examples). None of the curves are levelling off; performance grows log-linearly across the full range.

5 Lexical stress and L2P conversion

In this section, we evaluate various methods of combining stress prediction with phoneme generation. We first describe the specific system that we use for letter-to-phoneme (L2P) conversion. We then discuss the different ways stress prediction can be integrated with L2P, and define the systems used in our experiments. Finally, we provide the results.

5.1 The L2P system

We combine stress prediction with a state-of-the-art L2P system (Jiampojamarn et al., 2008). Like our stress ranker, their system is a data-driven sequence predictor that is trained with supervised learning. The score for each output sequence is a weighted combination of features. The feature weights are trained using the Margin Infused Relaxed Algorithm (MIRA) (Crammer and Singer, 2003), a powerful online discriminative training framework. Like other recent L2P systems (Bisani and Ney, 2002; Marchand and Damper, 2007; Jiampojamarn et al., 2007), this approach does not generate stress, nor does it consider stress when it generates phonemes.

For L2P experiments, we use the same training, testing, and development data as was used in Section 4. For all experiments, we use the development set to determine at which iteration to stop training in the online algorithm.

5.2 Combining stress and phoneme generation

Various methods have been used for combining stress and phoneme generation. Phonemes can be generated without regard to stress, with stress assigned as a post-process (Bagshaw, 1998; Coleman, 2000). Both van den Bosch (1997) and Black et al. (1998) argue that stress should be predicted at the same time as phonemes. They expand the output set to distinguish between stressed and unstressed phonemes. Similarly, Demberg et al. (2007) produce phonemes, stress, and syllable-boundaries within a single joint n-gram model. Pearson et al. (2000) generate phonemes and stress together by jointly optimizing a decision-tree phoneme-generator and a stress predictor based on stress pattern counts. In contrast, Webster (2004) first assigns stress to letters, creating an expanded input set, and then predicts both phonemes and stress jointly. The system marks stress on letter vowels by determining the correspondence between affixes and stress in written words.

Following the above approaches, we can expand the input or output symbols of our L2P system to include stress. However, since both decision tree systems and our L2P predictor utilize only local context, they may produce invalid global output. One option, used by Demberg et al. (2007), is to add a constraint to the output generation, requiring each output sequence to have exactly one primary stress.

We enhance this constraint, based on the observation that the number of valid output sequences is fairly limited (Section 3.2). The modified system produces the highest-scoring sequence such that the output's corresponding stress pattern has been observed in our training data. We call this the **stress pattern constraint**. This is a tighter constraint than having only one primary stress.[4] Another advantage is that it provides some guidance for the assignment of secondary stress.

Inspired by the aforementioned strategies, we evaluate the following approaches:

1) JOINT: The L2P system's input sequence is letters, the output sequence is phonemes+stress.

2) JOINT+CONSTR: Same as JOINT, except it selects the highest scoring output that obeys the stress pattern constraint.

3) POSTPROCESS: The L2P system's input is letters, the output is phonemes. It then applies the SVM stress ranker (Section 3) to the phonemes to produce the full phoneme+stress output.

4) LETTERSTRESS: The L2P system's input is letters+stress, the output is phonemes+stress. It creates the stress-marked letters by applying the SVM ranker to the input letters as a preprocess.

5) ORACLESTRESS: The same input/output as LETTERSTRESS, except it uses the gold-standard stress on letters (Section 4.1).

[4] In practice, the L2P system generates a top-N list, and we take the highest-scoring output on the list that satisfies the constraint. If none satisfy the constraint, we take the top output that has only one primary stress.

System	Eng		Ger	Dut
	P+S	*P*	*P*	*P*
JOINT	78.9	80.0	86.0	81.1
JOINT+CONSTR	84.6	86.0	90.8	88.7
POSTPROCESS	86.2	87.6	90.9	88.8
LETTERSTRESS	86.5	87.2	90.1	86.6
ORACLESTRESS	91.4	91.4	92.6	94.5
Festival	61.2	62.5	71.8	65.1

Table 5: Combined phoneme *and* stress prediction word accuracy (%) for English, German, and Dutch. *P*: predicting primary stress only. *P+S*: primary and secondary.

Note that while the first approach uses only local information to make predictions (features within a context window around the current letter), systems 2 to 5 leverage global information in some manner: systems 3 and 4 use the predictions of our stress ranker, while 2 uses a global stress pattern constraint.[5]

We also generated stress and phonemes using the popular Festival Speech Synthesis System[6] (version 1.96, 2004) and report its accuracy.

5.3 Results

Word accuracy results for predicting both phonemes and stress are provided in Table 5. First of all, note that the JOINT approach, which simply expands the output set, is 4%-8% worse than all other comparison systems across the three languages. These results clearly indicate the drawbacks of predicting stress using only local information. In English, both LETTERSTRESS and POSTPROCESS perform best, while POSTPROCESS and the constrained system are highest on German and Dutch. Results using the oracle letter stress show that given perfect stress assignment on letters, phonemes and stress can be predicted very accurately, in all cases above 91%.

We also found that the phoneme prediction accuracy alone (i.e., without stress) is quite similar for all the systems. The gains over JOINT on combined stress and phoneme accuracy are almost entirely due to more accurate stress assignment. Utilizing the oracle stress on letters markedly improves phoneme prediction in English (from 88.8% to 91.4%). This can be explained by the fact that English vowels are often reduced to schwa when unstressed (Section 2).

Predicting both phonemes and stress is a challenging task, and each of our globally-informed systems represents a major improvement over previous work. The accuracy of Festival is much lower even than our JOINT approach, but the relative performance on the different languages is quite similar.

A few papers report accuracy on the combined stress and phoneme prediction task. The most directly comparable work is van den Bosch (1997), which also predicts primary and secondary stress using English CELEX data. However, the reported word accuracy is only 62.1%. Three other papers report word accuracy on phonemes and stress, using different data sets. Pearson et al. (2000) report 58.5% word accuracy for predicting phonemes and primary/secondary stress. Black et al. (1998) report 74.6% word accuracy in English, while Webster (2004) reports 68.2% on English and 82.9% in German (all primary stress only). Finally, Demberg et al. (2007) report word accuracy on predicting phonemes, stress, *and* syllabification on German CELEX data. They achieve 86.3% word accuracy.

6 Conclusion

We have presented a discriminative ranking approach to lexical stress prediction, which clearly outperforms previously developed systems. The approach is largely language-independent, applicable to both orthographic and phonetic representations, and flexible enough to handle multiple stress levels. When combined with an existing L2P system, it achieves impressive accuracy in generating pronunciations together with their stress patterns. In the future, we will investigate additional features to leverage syllabic and morphological information, when available. Kernel functions could also be used to automatically create a richer feature space; preliminary experiments have shown gains in performance using polynomial and RBF kernels with our stress ranker.

Acknowledgements

This research was supported by the Natural Sciences and Engineering Research Council of Canada, the Alberta Ingenuity Fund, and the Alberta Informatics Circle of Research Excellence.

[5] This constraint could also help the other systems. However, since they already use global information, it yields only marginal improvements.

[6] http://www.cstr.ed.ac.uk/projects/festival/

References

Joanne Arciuli and Linda Cupples. 2006. The processing of lexical stress during visual word recognition: Typicality effects and orthographic correlates. *Quarterly Journal of Experimental Psychology*, 59(5):920–948.

Harald Baayen, Richard Piepenbrock, and Leon Gulikers. 1996. The CELEX2 lexical database. LDC96L14.

Paul C. Bagshaw. 1998. Phonemic transcription by analogy in text-to-speech synthesis: Novel word pronunciation and lexicon compression. *Computer Speech and Language*, 12(2):119–142.

Susan Bartlett, Grzegorz Kondrak, and Colin Cherry. 2008. Automatic syllabification with structured SVMs for letter-to-phoneme conversion. In *ACL-08: HLT*, pages 568–576.

Maximilian Bisani and Hermann Ney. 2002. Investigations on joint-multigram models for grapheme-to-phoneme conversion. In *ICSLP*, pages 105–108.

Alan W Black, Kevin Lenzo, and Vincent Pagel. 1998. Issues in building general letter to sound rules. In *The 3rd ESCA Workshop on Speech Synthesis*, pages 77–80.

Noam Chomsky and Morris Halle. 1968. The sound pattern of English. *New York: Harper and Row*.

Kenneth Church. 1985. Stress assignment in letter to sound rules for speech synthesis. In *ACL*, pages 246–253.

Cynthia G. Clopper. 2002. Frequency of stress patterns in English: A computational analysis. *IULC Working Papers Online*.

John Coleman. 2000. Improved prediction of stress in out-of-vocabulary words. In *IEEE Seminar on the State of the Art in Speech Synthesis*.

Michael Collins and Terry Koo. 2005. Discriminative reranking for natural language parsing. *Computational Linguistics*, 31(1):25–70.

Michael Collins. 2002. Discriminative training methods for Hidden Markov Models: Theory and experiments with perceptron algorithms. In *EMNLP*, pages 1–8.

Koby Crammer and Yoram Singer. 2003. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991.

Vera Demberg, Helmut Schmid, and Gregor Möhler. 2007. Phonological constraints and morphological preprocessing for grapheme-to-phoneme conversion. In *ACL*, pages 96–103.

Qing Dou. 2009. An SVM ranking approach to stress assignment. Master's thesis, University of Alberta.

Kenneth Dwyer and Grzegorz Kondrak. 2009. Reducing the annotation effort for letter-to-phoneme conversion. In *ACL-IJCNLP*.

Erik C. Fudge. 1984. English word-stress. *London: Allen and Unwin*.

Sittichai Jiampojamarn, Grzegorz Kondrak, and Tarek Sherif. 2007. Applying many-to-many alignments and Hidden Markov Models to letter-to-phoneme conversion. In *NAACL-HLT 2007*, pages 372–379.

Sittichai Jiampojamarn, Colin Cherry, and Grzegorz Kondrak. 2008. Joint processing and discriminative training for letter-to-phoneme conversion. In *ACL-08: HLT*, pages 905–913.

Thorsten Joachims. 1999. Making large-scale Support Vector Machine learning practical. In B. Schölkopf and C. Burges, editors, *Advances in Kernel Methods: Support Vector Machines*, pages 169–184. MIT-Press.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *KDD*, pages 133–142.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional Random Fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289.

Yannick Marchand and Robert I. Damper. 2007. Can syllabification improve pronunciation by analogy of English? *Natural Language Engineering*, 13(1):1–24.

Steve Pearson, Roland Kuhn, Steven Fincke, and Nick Kibre. 2000. Automatic methods for lexical stress assignment and syllabification. In *ICSLP*, pages 423–426.

Dan Roth and Wen-tau Yih. 2005. Integer linear programming inference for conditional random fields. In *ICML*, pages 736–743.

Lara Tagliapietra and Patrizia Tabossi. 2005. Lexical stress effects in Italian spoken word recognition. In *The XXVII Annual Conference of the Cognitive Science Society*, pages 2140–2144.

Antal van den Bosch. 1997. *Learning to pronounce written words: A study in inductive language learning*. Ph.D. thesis, Universiteit Maastricht.

Gabriel Webster. 2004. Improving letter-to-pronunciation accuracy with automatic morphologically-based stress prediction. In *ICSLP*, pages 2573–2576.

Briony Williams. 1987. Word stress assignment in a text-to-speech synthesis system for British English. *Computer Speech and Language*, 2:235–272.

George Kingsley Zipf. 1929. Relative frequency as a determinant of phonetic change. *Harvard Studies in Classical Philology*, 15:1–95.