University of Alberta

AN SVM RANKING APPROACH TO STRESS ASSIGNMENT

by

Qing Dou

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

Fall 2009
Edmonton, Alberta

Examining Committee

Greg Kondrak, Computing Science

Benjamin V. Tucker, Linguistics

Davood Rafiei, Computing Science

To my Mom,

It is Your wisdom that gives birth to this thesis

Abstract

The problem of stress assignment is to identify which syllables are phonetically more prominent than the others in a word. It is not only of theoretical interests to linguists but also very important to Text-to-Speech systems in terms of both accuracy and naturalness of pronunciation. Besides providing an in-depth survey of existing stress assignment algorithms in the fields of linguistics and speech generation, this thesis presents a ranking approach to stress assignment for both letters and phonemes. The final system is language independent and clearly outperforms all previous systems. When combined with a current state of art Letter-to-Phoneme system, error rate in stress assignment is reduced by up to 40%.

Acknowledgements

It is my luck to grow in such a wonderful NLP group. Especial thanks to Sittichai Jiampojamarn for all his contributions to the experiments, and to Shane Bergsma for his enthusiasm and resourceful ideas. And, of course, many thanks to Dr. Greg Kondrak for his fruitful supervision and suggestions. Also many thanks to my great friends: Abhishek Srivastava and Matthew Kyle Jorgensen for their hard work on reviewing the thesis.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Problem Description

Lexical stress, which is also called **accent**, refers to special emphasis, usually a higher tone and/or a longer duration, given to certain syllables in words. The problem of stress assignment is to identify the degree of stress for the syllables. Languages differ from each other in their stress systems. In some languages, the location of stress is entirely predictable. However, this is not the case for the others. Some languages have only two levels of stress, while others have more than two levels. There is a long history of research into this problem and many linguists have stated rules that explain the location of stress for either specific or a broad family of languages.

The problem has recently started to draw the attention of the computer speech community. The reasons are two fold. First of all, producing correct stress patterns is not a luxury for computer generated speech. Rather, it is very important in terms of both accuracy and naturalness of the speech. Secondly, it is well known that lexical stress affects word recognition rate of human (Joanne and Linda, 2006; Tagliapietra and Tabossi, 2005; Colombo, 1991). This agrees with research work (Rogier C. van Dalen and Rothkrantz, 2006; Wang and Seneff, 2001) in speech recognition, where increase can be found in recognition rate with help from a stress detection model.

Given the importance of stress patterns to pronunciation, researchers have begun to look for good stress assignment algorithms, which gives arise to the problem of

automatic stress assignment in **letter-to-phoneme(L2P)** systems. An L2P system, which converts an input word from its surface form to phonetic representation, is a part of **Text-to-Speech(TTS)** system. The phonetic transcription serves as input of a synthesis module, which generates actual sound signals. A complete phonetic transcription of a word does not contain only phonetic symbols but also prosodic information such as the location of syllabic accent.

Early L2P systems usually contained an independent word prosodic model for the task of stress assignment. Stress location was predicted in a pipeline process based on previously generated phonemes. However, this approach was proven to have worse performance than a joint approach (van den Bosch, 1997; Black et al., 1998). In the joint approach, output phoneme set was expanded in a way that stressed and unstressed phonemes are viewed as different symbols. Following this convention, some most recent works (Jiampojamarn et al., 2007; Bisani and Ney, 2002; Marchand and Damper, 2006) have ignored the problem of stress assignment.

Besides notating stress on phonemes, it is helpful and sometimes even necessary to notate it on letters. First and foremost, although native speakers usually have a good grip on stress locations, second-language learners often have difficulties in memorizing them. For instance, location of stress is often explicitly marked in textbooks for students of Russian. Moreover, in some languages such as Spanish, orthographic markers are obligatory for words with irregular stress.

Moreover, L2P systems can also benefit from orthographic stress markers. First of all, pronunciation of vowels is known to correlate closely with stress . Taking "accurate" and "acute" as an example, the first vowel letter "a" is pronounced as **[æ]** when stressed, while **[ə]** when not stressed. Secondly, the orthographic stress markers also serve as a global constraint, avoiding problems such as assigning multiple primary stress or no primary stress for a single word.

In a word, lexical stress is not only an interesting topic for linguists but also of great importance to the speech synthesis and recognition community. However, current algorithms for stress assignment are either language dependent or have rather poor performance. In the demand of existing data driven and language independent L2P systems, a better and more general stress assignment algorithm is highly

desirable.

1.2 Approaches to the Problem

To work in tandem with recent L2P systems, which normally employ a data driven approach, I took a data driven approach to the problem. Existing approaches to automatic stress assignment can be roughly divided into two categories: The first category includes approaches that employ rule based methods, where the level of stress depends on some contextual rules. Instead of being compiled manually, those rules are learned and stored automatically. The approaches in the second category try to capture some statistics of stress patterns using a generative model. For example, the stress level of a specific phoneme/syllable can be determined by frequency of stressed/unstressed phonemes (Pearson et al., 2000) or the probability of transitions between stressed and unstressed syllables (Demberg et al., 2007). Unfortunately, either category has its own drawbacks. The approaches that focus on local context usually ignore some global constraints, while those that look at global context are not able to utilize rich local context features.

To address the first issue, I formulated the task of stress assignment as a sequence prediction problem. It has been proven by previous research that there is a structure within the stress patterns. For example, each word only has one primary stress, and in metrical phonology, lexical stress is studied in a tree based framework.

To cope with the second issue, I took a discriminative approach to sequence prediction. Taking advantages of a limited number of possible stress patterns for each word, I trained an SVM ranker to select the most likely pattern from a small pool of alternatives. The resulting system takes words as input and produces stressed form of the words in three main steps: At the first step, an input word is divided into a sequence of substrings. The substring sequence is then coupled with all possible stress patterns in order to generate features that can be used by an SVM ranker to produce a score. After the pattern with the highest score has been chosen, a mapping process is introduced to map stress markers back to the input words. By using an SVM ranking approach, I can include arbitrary features over the entire input and

output sequences. Overall, the SVM ranker achieves exceptional performance.

To improve stress assignment in L2P conversion, I took two different approaches to incorporate the stress model with an existing L2P system. Firstly, following Webster (2004), I put stress markers on letters, which will provide the L2P system with additional input information. Alternatively, I took phonemes generated by the L2P system as input and placed stress markers on phonemes directly. Both resulting systems set a new standard for the state of art of predicting both phonemes and stress.

1.3 Contributions

This thesis makes a number of substantial contributions. First of all, it presents a language stress assignment algorithm that achieves exceptional performance in Germanic languages. Unlike previous systems that only work for a specific language, the proposed algorithm does not rely on any specific linguistic knowledge, and performs well on English, German, and Dutch. Most importantly, the results are much higher than those reported previously.

Secondly, when combined with a state of the art L2P system (Jiampojamarn et al., 2008), word accuracy considering stress and phonemes is improved from 78% to 86% for English, 81% to 88% for dutch, and 86% to 90% for German, which is up to a 40% reduction in error rate. Moreover, experiment results also show that the combined system clearly beats a publicly available TTS system.

Thirdly, the substrings based approach to stress assignment can work on both letters and phonemes. It also sets a norm for predicting stress on letters when perfect syllabification is not available.

Finally, the thesis conducts a thorough review of the task of stress assignment in the field of linguistics and computer speech, giving a clear image of various stress assignment algorithms and current sate of the art of automatic stress assignment for L2P systems.

1.4 Outline

The thesis is organized as follows. Chapter 2 provides some linguistic background of lexical stress. Chapter 3 introduces SVM and its related extensions, which finally leads to the SVM ranking algorithm. In Chapter 4, a thorough literature review of previous approaches to stress assignment is presented. Following that, I discuss how the SVM ranking algorithm is applied to the problem of stress assignment for letters and phonemes in chapter 5 and chapter 6 respectively. Then, in chapter 7, two different approaches are compared to combine the stress assignment model with a state of the art L2P system. In the end, chapter 8 concludes this thesis.

Chapter 2

Linguistic Background

In many languages, certain syllables in words are phonetically more prominent in terms of duration, pitch, and loudness. This phenomenon is referred to as **stress**. Unless otherwise specified, stress is discussed at the word level in this thesis.

Before I go deeply into some important linguistic theories of stress, I would like to first clarify some terminologies in this thesis. First and foremost, **word** is used to express a conceptual linguistic unit, which includes both orthographic (grapheme) and phonetic forms (phoneme).

Stress is a phonetic concept and usually notated in the phonetic form of a word. For example, in **IPA** (International Phonetic Alphabet), a raised mark ˈ shows the points of primary stress, while a lower mark ˌ is used for secondary stress. The stress markers are placed immediate before stressed syllables: [ˌpriˈsid]. In this thesis, the task of notating stress on phonemes is referred to as **phonetic stress assignment**.

Sometimes, it is also desirable to mark stress on letters. There are several good reasons for doing this. First of all, most people are not familiar with phonetic symbols in the IPA. Secondly, it is beneficial to beginners of a language in terms of correcting their pronunciation. And last but not least, some languages explicitly notate stress for words with irregular stress patterns. For orthographic forms, vowel letters bearing the stress are marked with an acute accent indicating primary stress and a grave accent for secondary stress (e.g. prècéde). I call the task of notating stress on letters **orthographic stress assignment**.

The objective of putting stress markers on letters in this thesis, which is to improve the performance of stress assignment in letter to phoneme conversion, is

merely engineering driven. Since it is not a trivial task to decide syllable boundaries (Bartlett et al., 2008), I put stress markers on vowel letters and phonemes that actually bear the stressed sound whenever a perfect syllabification is not available.

2.1 The Lexical Stress

Linguists often discuss lexical stress on the syllable level. Syllables are viewed as basic constructing units of streams of speech. For instance, the word "water" can be divided into two syllables "wa" and "ter." Here, "wa" is merely the written form of the first syllable, as a syllable also refers exclusively to a phonetic entity. The notion of stress is used to express the prominence of a syllable. It is important to bear in mind that stress is a relational concept. Whether a syllable is stressed or not is evaluated on the basis of its neighbors. Therefore, when a syllable is stressed, it is phonetically more prominent than its neighbors—generally louder and/or longer.

Stress can have different degrees. Some languages only have a single degree of stress, while others may have multiple degrees. In the case that a syllable is stressed but still weaker than the primary stressed one, a secondary stress is assigned to that syllable. English is commonly known to have two degrees of stress. For instance, word "precede" has second stress on the first syllable and primary stress on the second syllable. Regardless of the difference in degrees of stress, it is generally agreed that each word can have one and only one primary stress.

Similar to other variations, placement of stress is different across languages. Some languages have fixed stress—stress position can be predicted by fairly simple rules. For example, in Finnish and Hungarian stress is always on the first syllable, while in Polish stress always falls on the penult (the syllable before the last). However, in other languages such as English, Russian, and Dutch, stress position is variable. Rules that try to predict the stress position are much more complicated, and there are always exceptions that can not be covered.

As each word consists of one or more syllables, with some of them being stressed at different levels, we can substitute each syllable for an integer that indicates the syllable's stress level. The resulting sequence is called stress pattern.

In this paper, I use 1, 2, and 0 to represent primary stress, secondary stress, and no stress respectively. For instance, the stress pattern for "water" is 1-0. There are several interesting phenomena regarding the stress pattern. First of all, Clopper (2002) showed that people have strong preferences for a small subset of stress patterns. Furthermore, the number of unique stress patterns of each length is small; for example, in my training data there are only 71 unique stress patterns in English, 80 in German and 88 in Dutch.

2.2 Linguistic Theories of Lexical Stress

In linguistic theories, each language has its own stress system. However, they are divided into two main categories. The most obvious difference between the two categories lies in those systems which can be described by phonological rules and those whose placement of stress purely depends on the infrastructure of words. For languages which fall into the former category, linguists can find rules that govern stress positions for most words. Learners do not need to memorize stress positions separately except for a small number of words. People usually call systems of this kind **fixed stress systems**. In the other category, placement of stress is believed to interact closely with morphological changes, syllable positions, and even part of speech. There are no general stress assignment rules for those systems, and people have to remember the stress pattern for each word. Those systems are usually referred to as **free stress systems**. Nonetheless, those systems are of great interest to linguists and many theories on stress assignment, which are usually called **stress assignment algorithms**, have been developed.

2.2.1 Transformational Cycle

Previously, linguists believed that, in the same way as grammar, which can be used to parse sentences, there is also a set of rules for lexical stress. Chomsky and Halle (1968) introduced the concept of **Transformational Cycle** for English phonology. The operation of a transformational cycle consists of two major processes: the grammatical transformation of a string and cyclical application of phonology rules

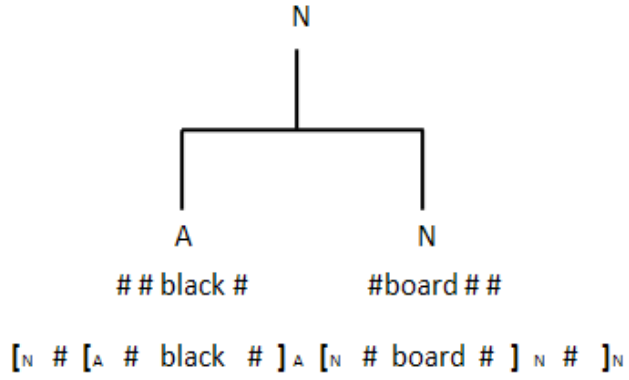

Figure 2.1: Formatives of compound word blackboard with surface structure marked, adapted from (Chomsky and Halle, 1968).

depending on the structure of the transformed string.

In the grammatical transformation process, a syntactic component converts a string into structured word affixes (formatives) with surface structure marked by. An example is shown in Figure 2.1, where the compound word $blackboard$ is divided into two formatives: $black$ with mark A (an adjective) and $board$ with mark N (a noun).

In the second process, phonological rules are applied sequentially. The order and selection of rules are determined by the structure and the syntactic tags of the transformed string. After each rule is applied, the innermost brackets will be removed. For instance, after applying the rule "In monosyllables, the vowel receives primary stress" we have a new representation:

$$[_N \#\# bla^1 ck \#\# bo^1 ard \#\#]_N$$

finally, by applying the rule "Assign primary stress to a primary-stressed vowel in the following context[1]",

$$- - \ldots V^1 \ldots]_N$$

the string will become

$$\#\# bla^1 ck \#\# bo^2 ard \#\#$$

[1]V stands for vowel and the dash indicates the position of the segment to which the rule applies. In this case, the position followed by a primary stressed vowel.

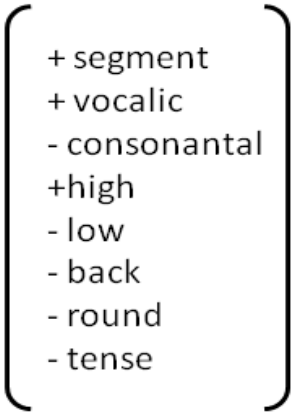

Figure 2.2: Distinctive features of letter i, adapted from (Chomsky and Halle, 1968). A plus sign indicates a particular feature is fired, and vice versa

In more complex contexts, a simple feature such as "V" is not enough. Rules are augmented by more features: **distinctive features**. Those features are viewed as characteristics that differentiate one utterance from another. For instance a feature vector of the sound "i" is shown in figure 2.2

2.2.2 Metric Tree Algorithm

Inspired by the alternation of stress used in poetries to form rhythmic patterns, phonologists began to study lexical stress in the framework of metrics and developed a new theory called metrical phonology (Prince, 1977). Later, Hayes (1981) described some metrical parameters that can be used to categorize languages according to their word-level stress patterns. Metrical phonology can be applied to analyze stress patterns at word, phrase, and even sentence level. The most important tool in metrical phonology is metrical tree, whose leaves are individual syllables.

A metrical tree is constructed in a bottom-up manner. At each level of construction, rules are applied to determine the relative prominence of all related branches. At the bottom level, syllables are first grouped together according to their weights[2].

[2]The weight of syllable is language dependent. Usually a syllable with long vowel or coda is viewed as heavy, while a syllable with short vowel or without coda is light

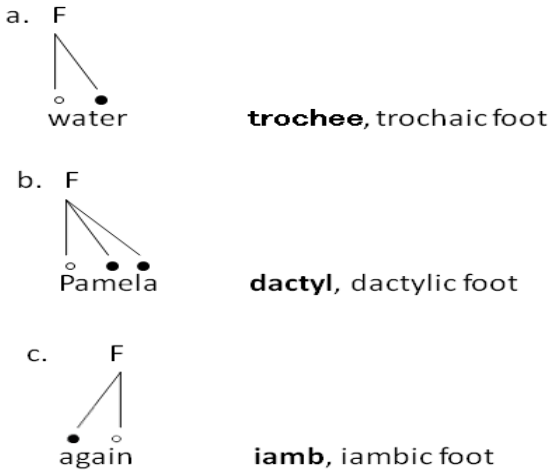

Figure 2.3: Common stress feet with their names in Greek, from (Spencer, 1996). The empty circle ∘ indicates a stressed syllable, while the dot · indicates an unstressed one.

Such a group of syllables is called **stress foot**. It is viewed as an important rhythmic unit in the study of metrical phonology. By definition, foot refers to a group of syllables consisting of exactly one syllable that is phonetically more prominent than the others. The more prominent syllable is called the stressed syllable in the foot, and the others are called unstressed syllables. One way to represent foot follows the tree diagram as shown in figure 2.3. The unstressed syllables are optional in a stress foot. A foot where unstressed syllables are absent is called **degenerate** foot.

The concept of stress foot can be applied to both fixed and free stress system. For instance, in those languages where syllables are always stressed alternately, every two syllables starting from the right or left hand side can form a binary stress foot. In the end, either the rightmost foot or the leftmost foot is more prominent than the others, and the stressed syllable in it receives the primary stress. In a more complicated case, syllable weight comes into play. The most prominent foot consists of either the rightmost or the leftmost syllables that are heavy.

In free stress systems, there is no single rule that can explain stress patterns across all words. Rather, a metrical tree needs to be constructed in a bottom up manner. At each level, a set of rules are applied to determine which branch is

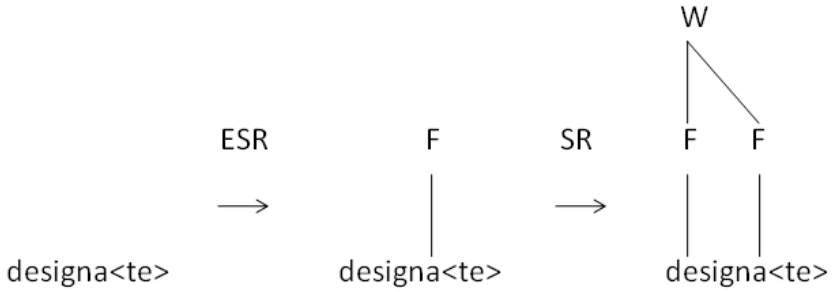

Figure 2.4: Formation of a simple metric tree, from (Spencer, 1996)

more prominent. For example, given two sibling nodes, the second is strong if, and only if, it branches. Another example is one of the most familiar rules for English called ESR (English Stress Rule): The final syllable is stressed when it is heavy, otherwise the penult is stressed. Figure 2.4 gives an example of how rules are applied at different levels of a tree to determine the stress location.

According to the illustration, stress is first assigned to the syllable “na” according ESR, and then by applying SR(strong retraction), stress is shifted to “de”.

2.2.3 Relative Frequency as a Determinant of Stress

It is well known that language is alive. Like any other living species, language changes all the time. It takes no more than a thousand years for a language to become unintelligible for our new generations. A language can change in all its aspects, including stress patterns. Therefore, any fixed rules for predicting stress positions will have no chance to survive the rapid changes. A better solution would be to find out some underlying mechanics that govern all those changes. Far from being a stress assignment algorithm, the following theory I am going to present sheds some light on such mechanics.

Zipf (1929) argued that frequency of use can serve as a determining factor of phonetic changes. To be more specific, he maintained that the degree of stress of any words or syllables is inversely proportional to their use in the streams of speech. In other words, if a word or a syllable is very frequently used, it will not be given any special attention, and thus will not be stressed.

Prefix	Frequency
ge-	443,639
be-	226,827
ver-	195,412
er-	122,662
an-	85,473
zu-	75,218
vor-	59,132
aus-	52,778
un-	49,831
ent-	48,456
da-	48,252
ein-	45,645

Table 2.1: Most frequent German prefixes together with their frequency of use, adapted from (Zipf, 1929)

Think about a simple example : “Mr Smith”. Which word will be stressed? Normally, the answer is “Smith”. The reason, according to Zipf’s theory, lies in that “Mr” is used much more frequently than “Smith” in the streams of speech. The theory also applies when it comes to syllabic accent. Taking an English word “worker” as an example, the frequency of the stressed syllable “work” is much lower than that of the second syllable “er” in spoken language. Interestingly, this phenomenon is not only found in English, but also in many other languages. In table 2.1, a list of German word prefixes together with their frequency in the streams of speech are presented.

As shown in the table, the first four prefixes with very high frequency are never stressed. As the frequency of the prefix decreases, some begin to attract stress. However, there are always some exceptions. For example, “ent”, with relatively low frequency, is never stressed. This phenomenon might be explained by the fact that shifts of stress take place over a long time. At present, the shift has not happened for that syllable. Compound words are good examples for such an explanation. Zipf stated that at the beginning of compounding, a prefix is only attached to a small number of words and is always stressed, making itself more noticeable. However, as the process goes, a prefix might be attached to more and more words,

which significantly increases its use in the streams of speech and loses its power of attracting stress.

Although this theory is not about stress assignment, it gives a deep insight into underlying mechanics that govern the shifts of stress positions.

2.3 Summary

In this chapter, I briefly introduced some important linguistic concepts of lexical stress: the syllable, the relational concept of stress, and different stress systems. Based on that, I discussed three linguistic theories of lexical stress. From the feature based algorithm to the theory of relative frequency, the underlying rules governing the stress positions in free stress systems have been explored. Given the fact that stress exhibits some regularities in terms of relative frequency and contextual features, I am convinced that modern machine-learning techniques can be applied to predict stress positions based on some statistics of such features.

Chapter 3

SVM Ranker

One contribution of this thesis is to formulate stress assignment as a sequence prediction problem and use Support Vector Machine(SVM) to train a model discriminatively in a ranking formulation. In this chapter, I will first give a short introduction to SVM as well as some of its variations. Then I will show how SVM can be applied to ranking problems.

3.1 The Support Vector Machine

Support Vector Machine refers to a class of max margin classifiers that are trained discriminatively. SVM views training data as two types of examples: positive and negative. Each of the examples is represented by a vector of features. Theoretically, there is no limit on the number of features, which allows rich feature representation of the training examples. The task of an SVM is to find a weight vector that separates the positive examples from the negative ones as far as possible. Before I go deeply into the detailed mathematic theories that support the above idea, it is easier to start with a simple example in the paradigm of linear classifiers.

Diagnosis is a good example of classification. In health services, diagnosis happens when a doctor comes to a conclusion on which disease a specific patient suffers from based on a set of syndromes. Suppose that a doctor needs to identify whether a patient suffers from strep throat based on only two syndromes: tonsil size and body temperature. The doctor decides to make a judgement on the basis of some historical data as shown in Figure 3.1, where black points represent positive

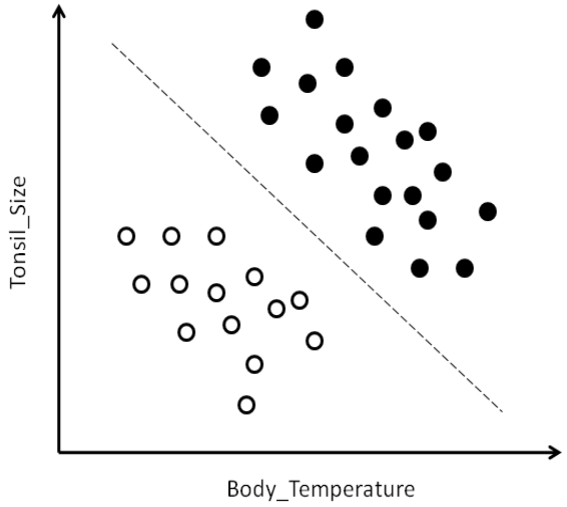

Figure 3.1: An example showing two sets of examples separated by a hyperplane

examples, and white points represent negative ones. In this figure, each point is described by only two variables (Body_Temperature,Tonsil_Size), which are called as features. If we draw a dash line (which is usually called hyperplane) in between, the two set of examples can be separated perfectly. Therefore, when making a decision, the doctor can simply classify any new case that falls below the dash line as a negative case, otherwise a positive case.

Maximize the Margin In Figure 3.1, it is easy to find many other hyperplanes to perfectly separate the two types of examples. The question then comes down to which hyperplane should be chosen. The most obvious criterion is that, the chosen hyperplane should perform better classification on unobserved data. From empirical experiences, intuition tells us to favor the plane that is as far as possible from both types of examples. The reason lies in that we never know what true distributions of the two types of examples are. The data used to plot Figure 3.1 is merely a limited amount of examples sampled from the true distributions. A plane

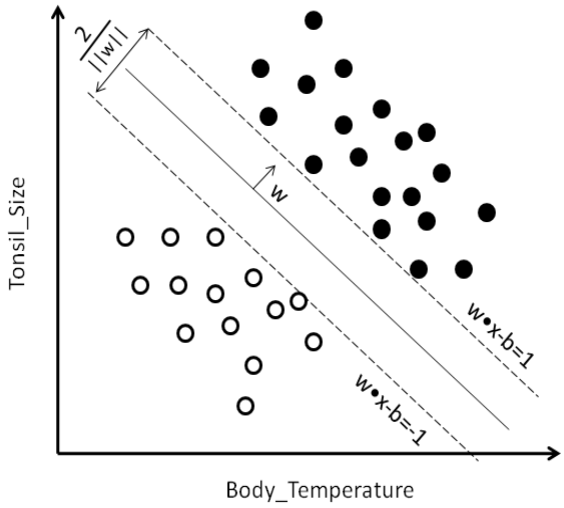

Figure 3.2: An example showing margin between the two sets of examples

that can perfectly separate the sampled examples might not be able to separate the examples in the true distributions well. By choosing a hyperplane that is far from each class of the observed examples, we hope that the plane chosen can separate the true distributions better.

The distance between the two types of examples is called margin. As shown in Figure 3.2, the margin is identified by the two parallel dashed lines, which represent support vectors. the solid line between the two dashed lines is the desired hyperplane. The position of the hyperplane is determined by the weight vector w, which is perpendicular to the plane. The distance between the two dash lines is inversed proportional to the norm of the weight vector $||w||$. Thus, to maximize the margin is equivalent to minimizing the norm of the weight vector $||w||$. SVM is designed to find out such a weight vector under some constraints. In this case, constraints are that the two types of examples fall below and above the two dashed lines respectively. Mathematically, this concept can be described using the following formula 3.1.

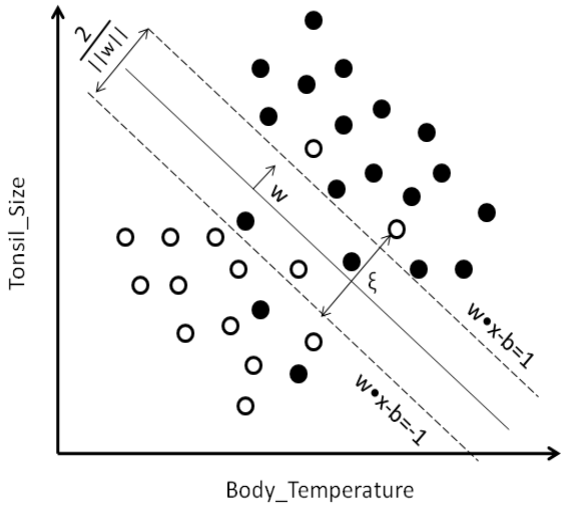

Figure 3.3: A none linear separable example

$$Min\frac{1}{2}||w||^2, \quad s.t.\ y_i(w \cdot x_i - b) \geq 1, \ \ 1 \leq i \leq n, y_i \in \{1, -1\} \tag{3.1}$$

The 1/2 is for mathematical convenience in solving the problem. The problem is convex and can be solved by standard quadratic programming techniques.

When it comes to classification, one can simple apply the trained w and b to a new instance x according to 3.2.

$$sign[w \cdot x - b] \tag{3.2}$$

3.1.1 SVM Extensions

However, in real scenarios, problems are often much more complicated than binary classification. The simple SVM presented in the previous section has to be modified to cope with more challenging problems.

Soft-Margin SVM In reality, a clear cut between two classes as shown in Figure 3.1 rarely exists. For instance, it is common for an experienced doctor to make a false judgement at the first glance of any new cases. This is because features are usually informative but not specific(determining). That is, they can help a doctor make a judgement but can not be used as basis of final decision. The reasons are two fold: First of all, diseases share a lot of common features such as fever and swollen tonsils; Secondly, due to individual differences, syndromes exhibits various levels of severity. Given the above complications, points in Figure 3.1 are rearranged as shown in Figure 3.3, where no straight lines can be found to separate the two set of examples. The problem thus becomes non-linear separable.

If the same constraints were still to be used as shown in formula 3.1, there would never be a solution when the space is non-linear separable. To deal with this problem, SVM introduces a slack variable ξ_i for each training instance. The variable can be viewed as a measure of tolerance to misclassification. With the slack variables, an SVM looks for a linear separation within an acceptable range of errors. To accommodate the change, not only the norm of weight vector $||w||$, but also the total error from misclassification will be minimized during training. The accordingly mathematic representation of the soft-margin SVM is given in formula 3.3

$$Min\frac{1}{2}||w||^2 + C \cdot \sum_i \xi_i, \quad s.t.\ y_i(w \cdot x_i - b) \geq 1 - \xi_i,\ \ 1 \leq i \leq n, y_i \in \{1, -1\} \tag{3.3}$$

The parameter C, which controls the total amount of acceptable error, can be tuned using a development set.

Multi-class SVM Another challenge that complicates the application of SVM is the multi-class problem. Think about the diagnosis problem again. Since most syndromes are not specific, a doctor usually needs to make a decision among several possible candidates when finalizing a diagnosis. Rather than making a binary judgement on each individual candidate, comparisons are made among all candidate. In the end, the most likely one is chosen as the final answer. Similarly, when

applying SVM to multi-class classification, an instance will not only be considered as positive or negative. Instead, all possible classes are evaluated simultaneously. To implement the above idea, constraints during training are changed as shown in formula 3.4,

$$Min\frac{1}{2}||w||^2+C\cdot\sum_i \xi_i, \quad s.t.\ \forall y_k w(\Psi(x_i,y')-\Psi(x_i,y_k)) \geq 1(y' \neq y_k)-\xi_i,\ \ 1 \leq i \leq n, \tag{3.4}$$

where y' is the only true label for each instance, and Ψ is a mapping of an instance to its feature vector through a possible label y_k. The new constraints ensure that the true labels receive higher scores than the false ones within the tolerance of ξ.

When it comes to classification, instead of using the sign to determine the class, the label with the highest score is chosen as the final prediction. In contrast to 3.2, the rule for multi-class classification is given in 3.5

$$y = argmax_{y_k} w\Psi(x,y_k) \tag{3.5}$$

3.2 Ranking SVM

Multi-class SVM assumes that the set of class labels is fixed. That is, the set of possible class labels is the same for each training instance. Furthermore, it assumes that only one class label is chosen as a final prediction. However, real world problems might not always satisfy these two assumptions. Often, the set of possible class labels is undefined. Moreover, when the selection pool is large, there is usually more than one ideal answer.

Modern search engines constantly face the above two problems. For a particular user query, a search engine is expected to return a set of candidates ordered by their relevance to key words in the query. However, limited by general heuristics that can be used to retrieve documents from a large number of web pages, a less relevant document might receive a high rank, while a more relevant one is overlooked. Therefore, search engine users usually click through top **k** pages until they finally find what they want. This suggests that, a search engine can be improved by

re-ranking the top k pages by some smarter knowledge. This is not the first time that the concept of re-ranking is introduced. In machine translation community, this technique has been widely used. The task of ranking is usually accomplished by a separate ranker.

A ranker is different from a multi-class classifier mainly in two ways. First of all, each input of a ranker consists of a set of examples while that of a classifier is a set of class labels. Secondly, the output of a ranker is an ordered set of the input examples. In contrast, the output of a multi-class classifier is a class label. Joachims (2002) first applied SVM to the ranking problem. Similarly, during training, a SVM is still going to minimize the norm of weight vector and the training errors. The only change made is in constraints. In ranking formulation, each training instance x consists of a set of examples (or class labels) $e_1, e_2 \ldots, e_n$ together with their rank relations r^* ($\forall(e_i, e_j) \in r^*$, e_i is ranked higher than e_j). Each relation poses a constraint, resulting multiple constraints for one training instance. The constraints for n training instances are shown in formula 3.6.

$$\begin{array}{l} \forall(e_i, e_j) \in r^* : w(\Psi(x_1, e_i) - \Psi(x_1, e_j)) \geq 1 - \xi_{i,j,1} \\ \forall(e_i, e_j) \in r^* : w(\Psi(x_2, e_i) - \Psi(x_2, e_j)) \geq 1 - \xi_{i,j,2} \\ \ldots \\ \forall(e_i, e_j) \in r^* : w(\Psi(x_n, e_i) - \Psi(x_n, e_j)) \geq 1 - \xi_{i,j,n} \end{array} \tag{3.6}$$

To rank a set of examples $e_1, e_2 \ldots, e_n$ for a new instance x_{new} in testing, scores can be calculated by formation 3.7.

$$w\Psi(x_{new}, e_i) \;\; i \in 1 \ldots n \tag{3.7}$$

In this thesis, I will show that the SVM ranker can also be applied to the stress assignment problem. If we view each word as an instance, all of its possible stress patterns will serve as input and output of the ranker. Then, an SVM is trained so that the correct stress pattern for a particular word receives a higher rank than all the other alternatives.

3.3 Summary

The chapter introduced SVM and several of its related extensions, which finally lead to the discussion of SVM ranker. The benefits of using SVM ranker in the task of stress assignment are threefold: First of all, taking each stress pattern as a single unit avoids producing nonsensical patterns, such as multiple or none primary stress in a word; Secondly, any helpful linguistic features can be easily added to the model; Lastly, the SVM maximizes the classification objective, which usually leads to better performance.

Chapter 4

Previous Work

The problem of stress assignment has remained as an interesting problem since the first time it was introduced. Some famous theories about lexical stress include Chomsky's transformational cycle (Chomsky and Halle, 1968) and Liberman's metrical theory (Prince, 1977), which have been introduced in chapter 2. The objective of this chapter is to give an overview of existing automatic approaches to stress assignment and their performance.

A direct comparison with works from linguistics is nonsensical, as researches in linguistics do not focus on how stress patterns of out of vocabulary(**OOV**) words can be predicted, but try to develop a theory that gives explanations to most in vocabulary words. Even in the field of computational linguistics there are also several factors that impede fair comparisons. First of all, stress assignment model is viewed as a part of L2P systems. When it comes to evaluations of those systems, some only look at phoneme accuracy, while others focus on overall performance (phoneme and stress). Occasionally, results of stress assignment are evaluated independently. Secondly, different types of corpus in use further complicate a fair comparisons across various works. The main reason lies in that the degree of challenge varies from one corpus to another. Thirdly, even using the same corpus, a different split of data for training and testing can result in a significant change in performance. Therefore, it is necessary to be aware that the numbers reported from different works are not always directly comparable.

Existing approaches to stress assignment can be roughly divided into two categories: rule based approaches and sequence based approaches. In the first category,

the level of stress is predicted by specific phonological rules for each syllable, letter, or phoneme independently. In the second category, assignment of stress is viewed as a sequence prediction problem. In the following sections, I will first introduce related works belong to the first category. However, a preference is given to works from the second category as they are closely related to my approach and sometimes even serve as important inspirations. Moreover, they also appear to have much better performance. In the end, I will present some other works that appear to be less significant, yet still provide some interesting insights into the problem of stress assignment.

4.1 Rule Based Approaches

4.1.1 Stress Assignment by Weight Patterns

Early rule based approaches date back to 1980s, when Church (1985) applied a table lookup technique to stress assignment. Church (1985) argued that stress could not be predicted by just looking at a five or six character long context window. Therefore, the work sought to invent a framework for dealing with long distance dependencies across syllables, which turned out to be a table loop-up approach. The resulting system took 4 linguistic features into consideration: syllable weight, part of speech, morphology, and etymology, among which, syllable weight was viewed as a key determinant. Syllable weight was used as an intermediate level connecting orthography and stress patterns as shown below.

- Orthography $\rightarrow$ Weight
- Weight $\rightarrow$ Stress

Originally, syllable weight was a binary feature ($Heavy(\mathbf{H})or Light(\mathbf{L})$), which was determined by some linguistic rules (e.g. a syllable is heavy if it is close or has long vowel). Each word then could be associated with a **weight** pattern, for instance, $obey \rightarrow LH$. The process of mapping from weight pattern to stress pattern was accomplished by looking up in a weight table. Table 4.1.1 gives an example.

Weight	Part of Speech	
	Verb	Noun
H	1	1
L	1	1
HH	31	10
HL	10	10
LH	01	10
LL	10	10
HHH	103	310
HHL	310	310
HLH	103	100
HLL	310	100
LHH	103	010
LHL	010	010
LLH	103	100
LLL	010	100
etc.		

Table 4.1: Weight table showing weight patterns and corresponding stress patterns, adapted from (Church, 1985)

The number in the stress patterns indicated different stress levels: 1 for primary stress, 2 for secondary, and 3 for tertiary. Normally, people agree that there are two levels of stress in English, the concepts of tertiary and quaternary usually disagree with each other.

In order to resolve some conflicts introduced by the original table, weight features were changed to multi valued. Besides syllable weight, morphology and etymology were also considered. For morphology, a morphological parser was used to parse word affix into different levels and associate each of them with a weight. In the end, word etymology knowledge was used to treat native and loan words differently.

4.1.2 Decision Trees

Black et al. (1998) discussed issues in building general letter to sound rules. The rules that convert letters to phonemes were constructed automatically through classification and regression trees **CRT**, which was also called decision trees **DT**. Given an input vector X with m attributes $a_1, \ldots, a_m$, a trained decision tree outputs a

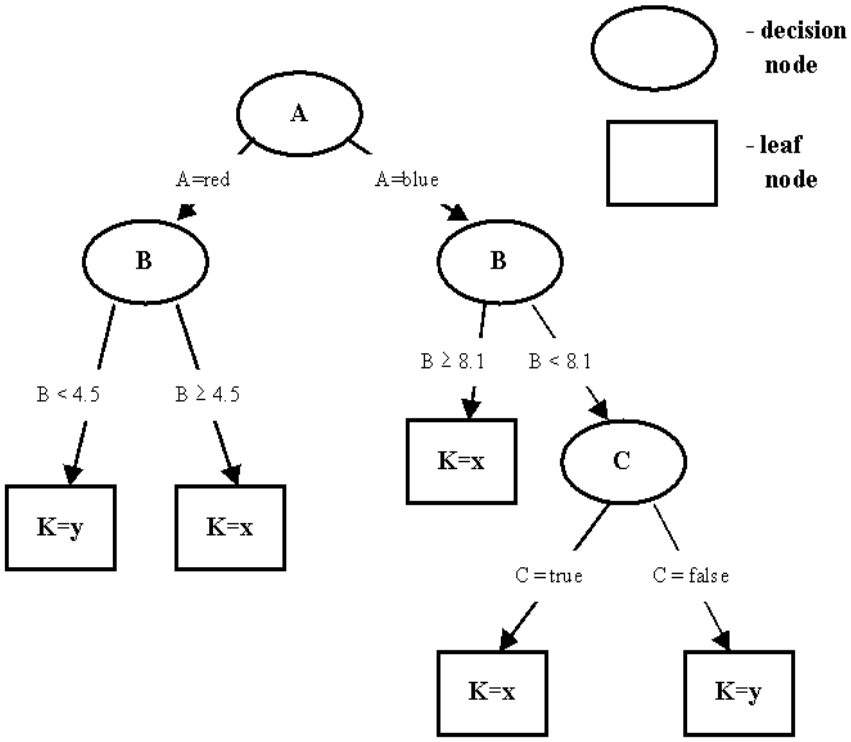

Figure 4.1: A simple example of decision tree

class label for X. For L2P problems, each input vector X represents a single letter. The attributes of the vector are usually surrounding letters. As shown in Figure 4.1, a decision tree takes an input vector and selectively asks questions regarding a specific attribute at decision nodes until a leaf node(also a class lable) is reached. The order of which attribute is examined is determined by how informative an attribute is. There are a number of advantages for using decision trees. First of all, the final representations are intuitive and interpretable. Secondly, they do not depend on assumptions of data distribution, which are suitable for exploring various less studied problems. Lastly, they scale well and can be trained efficiently.

The issue of stress assignment was addressed and compared in two different ways. In a joint approach, stress was predicted with phonemes jointly in the same tree. That is, stressed and unstressed phonemes were treated as different class labels. Whether an output phoneme is stressed or not relies on letters within a context window (4 in the left and 4 in the right) and part of speech.

On the other hand, a separate stress prediction model was also constructed, us-

Actual	Predicted		
	unstressed	stressed	%
unstressed	7390	378	95.1%
stressed	512	8207	94.1%

total correct 15597/16487 (94.6%)

Table 4.2: Accuracy of stress prediction evaluated per syllable on Oxford advanced learner dictionary, Adapted from (Black et al., 1998)

ing features such as syllable position, vowel height, vowel length. The results of stress prediction using the second approach are shown in Table 4.2

Although the stress assignment algorithm alone can achieve a very high accuracy per syllable, results on the same data set from Table 4.3 suggest that it is better to predict stress jointly with phonemes.

	L2P+S	LTPS
LNS	96.36%	96.27%
Letter	–	95.80%
WNS	76.92%	74.69%
Word	63.68%	74.56%

(LNS=letter/phone ignoring stress, WNS=word ignoring stress)

Table 4.3: Comparisons of stress predicton between a pipeline and a joint system, Adapted from (Black et al., 1998). L2P+S is a pipeline system, while L2PS is a joint system

4.1.3 Stress Assignment by Context Free Grammar

Coleman (2000) used probabilistic context free grammar to model the theory of metrical phonology. Words were analyzed at different levels. For instance, a word could be simple, containing only one prosodic word; or compound, with multiple prosodic words. A prosodic word could be further decomposed into stress feet. Each stress foot compromised one or more syllables, which were headed by a stressed syllable. A syllable could be further divided into onset and rime to determine its weight. Analysis was done in a bottom up manner using context free grammar. Part of its productions used are presented in Figure 4.2

Given such a set of grammars, any standard parsing algorithms could be applied to construct a metric tree. The grammar was designed to ensure that all the words in

$$\text{Word} \rightarrow \text{PrWd}$$

$$\text{Word} \rightarrow \text{PrWd PrWd}$$

$$\text{PrWd} \rightarrow \text{Foot}_{\text{w}} \text{Foot}_{\text{w}} \text{Foot}_{\text{s}}$$

$$\text{PrWd} \rightarrow \text{Syllable}_{\text{w}} \text{Foot}_{\text{s}}$$

$$\text{Foot} \rightarrow \text{Syllable}_{\text{s}} \text{Syllable}_{\text{w}}$$

$$\text{Foot} \rightarrow \text{Syllable}_{\text{s}} \text{Syllable}_{\text{w}} \text{Syllable}_{\text{w}}$$

$$\text{Syllable} \rightarrow \text{Onset Rime}$$

Figure 4.2: A subset of rules in the context free grammar for stress assignment

a machine-readable dictionary could be parsed. Then a development cycle was introduced to perfect the grammar, including adding and modifying them to increase prediction accuracy. However, the induced deterministic grammar had rather poor performance, with only 67% of words correct.

Most of mistakes arose from structural ambiguity—a word could be parsed into several trees. To disambiguate possible parses, probabilistic context free grammar was used. Rule probabilities were estimated from the training data using Maximum Likelihood Estimation **MLE**. Then quality of parses were then evaluated by the likelihood of the parses. According to the report, the probabilistic grammar had a significant improvement (from 67% to 75%) over the deterministic one.

4.2 Stress Assignment as Sequence Prediction

4.2.1 Joint n-gram Approach

Demberg et al. (2007) applied a joint n-gram model to the L2P problem, including stress assignment and syllabification. As shown in Equation 4.1, a joint n-gram model uses Viterbi algorithm to seek the most likely phoneme sequence $p_1^n : p_1, p_2, \ldots p_n$ given a sequence of letters $l_1^n : l_1, l_2, \ldots l_n$. Each letter-phoneme pair $< l; p >$ is viewed as a state, whose probability is dependant on previous k states $< l; p >_{i-k}^{i-1}$ (previously generated letter-phoneme pairs).

$$p_1^n = argmax \prod_{i=1}^{n} P(< l; p >_i \mid < l; p >_{i-k}^{i-1}) \tag{4.1}$$

When syllabification and stress assignment were treated as a joint process in L2P conversion, a state pair was changed into a tuple containing four tags $< l; p; b; a >$, where b indicated a syllable boundary and a stood for a stressed phoneme. The probability of each state was estimated using MLE. Due to major data sparse problems, probabilities of unobserved states were estimated by a variant of Modified Kneser-Ney Smoothing (Chen and Goodman, 1996)

The most significant contribution of the work was the introduction of phonological constraints. Particularly, in stress assignment, they found out that 15-20% of words were incorrectly stressed, among which 37% had more than one primary stress, about 27% were not assigned any stress, and 36% were stressed in a wrong position. This meant that almost 2/3 of the errors in stress assignment were caused by violation of an important phonological constraint—each word has only one primary stress. To incorporate this constraint into the model, an additional flag A was added. The conditional probability of each state then became:

$$P(< l; p; b; a >_i \mid < l; p; b; a >_{i-k}^{i-1}, A) \tag{4.2}$$

The transition probability is 0 if a constraint is violated, e.g. when the A flag indicates that a stress has been assigned and a_i indicates another accent.

They tested their model on CELEX German (Baayen et al., 1996). The training data contained 240k words, and the testing data contained 12,326 words. Different from traditional random splitting stratagem, the test data was carefully designed so that the inflections of a word were either all in the training set or all in the test set. Stem overlap between training and test set only occurred in compounds and derivations. Experiment results showed that the constraint was very effective, reducing error rate of stress assignment on letters from 30% to 9.9%.

4.2.2 Global Statistics

Although the one word one primary stress constraint works very well for German, it is not applicable in some other languages such as English. It is well known that words in English have multiple levels of stress, and there seems to be no constraints on secondary stress: that is, a word may have none, one, or more than one secondary stress.

Pearson et al. (2000) compared a few approaches to stress assignment and introduced a new method called global statistics. Instead of predicting stress level for each vowel phoneme sequentially, the method selected the most likely stress pattern for each input word as a whole. The idea of stress prediction as sequence selection was based on a limited number of possible stress patterns. The authors found out that, in the 19,000 word Cybertalk dictionary, there were only 118 stress patterns. For the 95,000 word Comlex dictionary, the number was 151. Moreover, 95% of the stress patterns had less than 5 syllables and some of them appeared much more frequently than the others.

There were two main components in the system. The first one was a frequency table of all stress patterns, containing counts of each stress pattern observed in the training data. The second one was a frequency table for vowels. Each line contained a vowel phoneme and its counts for each stress level. When it came to stress assignment, all possible stress patterns for a word were enumerated. Then, for each possible pattern, a product of pattern frequency and counts of each vowel at a specific stress level indicated by the pattern, was computed. In the end, the pattern with the highest product was chosen as the final prediction.

The Global Statistics reported a much higher accuracy than decision trees. The former achieved 81.0% word accuracy on STLs Cybertalk dictionary, while the latter one only got 67.3% accuracy. To take the local context ignored by the Global Statistics into account, a combined approach was introduced. In the combined approach, product of counts from Global Statistics was used to optimize phoneme sequences generated by a decision tree. Experiment results showed that the combined method was even superior, increasing word accuracy from 81.0% to 87.1%.

4.3 Other Related Work

Early implementations of the metrical phonology theory for word stress assignment date back to 1980s. Williams (1987) described a pipeline system based on the metrical phonology for British English. At initial stage, a syllabification unit borrowed an algorithm described in (O'Connor and Trim, 1953) to divide phoneme strings into syllables in comply with the Maximum Onset Principle (Selkirk, 1982). A mapping process then mapped syllables from phonetic representations to class label, such as consonant(C),long vowel(V). Afterwords, a pattern matching approach was taken to parse the syllables into stress foot(Section 2.2.2) according to their weights. The next step was to choose the foot that bears the primary stress or secondary stress, which was accomplished according to a set of rules(e.g. given two sister nodes, the second is strong if, and only if, it branches). Finally, the syllable bearing the stress was determined, which is always the first syllable within a foot. Test was taken using 1055 unique words, among which 706 were regular according to Fudge's criteria (Fudge, 1984). On average, 80% of words were assigned with a correct stress pattern by the system. The accuracy for regular words was 92%, while only 54% for irregular words.

Bagshaw (1998) took a table lookup approach that was originally proposed by Church (1985) to the problem of stress assignment. The method worked on syllabified phoneme sequences. A table stored mappings between sequence of syllable weight and stress pattern. Therefore, the first stage of the algorithm was to decide the syllable weight. The weight of a syllable was distinguished at a three level granularity :reduced, light and heavy, according to phonemic structure of the syllable and the type of its nucleus. As defined by (Bagshaw, 1998), reduced syllables contain a reduced vowel (or syllabic consonant). Light syllables are those containing a short vowel that is not followed by any consonants within the same syllable. Heavy syllables consist of either a long vowel or diphthong, or short vowel plus at least one following consonant. After each syllable's weight was decided, a lookup table was used to choose a corresponding stress pattern. Moreover, part-of-speech tag of a certain word also influenced the mapping. For example, a word

tagged as a verb (past tense) with a weight sequence LRHH (light, reduced, heavy, heavy) was assigned the stress sequence PUSU (primary stress, unstressed, secondary stress, unstressed), whereas a common noun (plural) with the same weight sequence LRHH was assigned SUPU. The method reported a 64.97% - 83.26% per syllable accuracy for English.

It is well known that stress patterns usually correlate with morphological process, that is, a morphological change often results in shift of stress positions. For instance, the word $phótograph$ has its primary stress on the first syllable, while in its corresponding noun $photógraphy$, the primary stress falls on the second syllable. Inspired by this phenomenon, Webster (2004) proposed an automatic morphologically-based stress prediction algorithm to improve L2P accuracy. Different from many other works, the algorithm predicted which letter receives primary stress rather than phoneme. However, secondary stress was not considered. In training phase, a word was first decomposed into pseudo morphemes(orthographic prefix and suffix), which was claimed as a patent, and therefore was not explained in details. A greedy algorithm was then used to iteratively find the prefix or suffix that correlates strongest with a particular stress location. For prefix, location indicated the number of vowel letters from beginning of a word. For suffix, location was counted from the end of a word. For example, in German, the prefix ver correlates strongly with location 2 and in English, suffix $ation$ highly correlates with location 3. The algorithm itself achieved an accuracy of 80.3% for English and 81.0% for German. In combination with a decision tree, the stress location was added as an additional feature. In the combined approach, the accuracy for English and German was increased to 87.1% and 92.2% respectively.

The affect of orthographic affix on stress positions is also noticed in (Arciuli and Thompson, 2006). This work first investigated the limitation of current Text-To-Speech systems(TTS) in terms of stress assignment. They ran a stimuli through a TTS system and examined the phoneme output. Results showed that the system correctly assigned stress for only 64% of words. In another test, which involved human participants, results suggested that the TTS system did not assign stress adequately. Based on findings from research in psychology (Joanne and Linda, 2006), they de-

signed a neural network to predict stress locations depending on words' beginnings and endings. They only chose words with 2 syllables from CELEX (Baayen et al., 1996) for training and testing. The selected words were then all represented by their beginnings and endings. Beginnings were defined as all the letters up to and including the first vowel or vowel cluster. For example, the beginning of word $saucy$ is sau. Endings were defined reversely from the end of a word. For instance, the ending of word $saucy$ is y. Words with long beginnings or endings were excluded. Then a neural network with 216 inputs, 28 hidden nodes, and 4 outputs was used to predict stress. A word accuracy of 86% was reported. However, the limitation of this approach was also obvious. First of all, words selected in the experiment have only less than 3 syllables. Secondly, only primary stress was considered.

4.4 Summary

Early approaches to automatic stress assignment directly borrowed language specific rules, which were normally complicated and not well structured. More recent data driven approaches tended to learn pronunciation rules automatically but ignored many important linguistic constraints. Later, people began to view stress assignment as a sequence prediction problem that took care of some global constraints and achieved significantly better results. However, they are still far from perfect and can be greatly improved by the method to be presented in the following chapter.

Chapter 5

Orthographic Stress Assignment

Marking vowel letters bearing stress is one of the objectives of this thesis. To accomplish the task, I designed a system that takes words in their written form as input and outputs words with markers on specific vowels bearing the stress. Instead of classifying stress level for each individual letter, the challenge is formulated as a sequence prediction problem within a powerful discriminative ranking framework.

This chapter will present the system and experimental results in details. As the approach is data driven, I will begin with explaining how the dataset for training, developing, and testing is prepared. Then, I will outline the three main components of the system in Section 5.2. Afterward, each sub-system: word splitting, SVM-ranking, and pattern to vowel mapping is introduced. Finally experimental results and analysis are provided.

5.1 Dataset

The Dataset used for this thesis is prepared from CELEX (Baayen et al., 1996). CELEX is a product of the Max Planck Institute for Psycholinguistics in the Netherlands. The dictionary consists of three languages (English, Dutch, and German), and includes both orthographic and phonetic forms of words. The English section of CELEX is compiled from the Oxford Advanced Learner's Dictionary (1974) and the Longman Dictionary of Contemporary English (1978). It contains 160,595 English words of British English. The German portion of CELEX is based on the Bonnlex and Molex computer dictionaries, as well as a German spelling lexicon. It

contains 365,530 words. The Dutch CELEX is derived from Van Dale's Comprehensive Dictionary of Contemporary Dutch (1984) and the Groene Boekje Word List of the Dutch Language (1954), and most of the dictionary entries from the Institute for Dutch Lexicology's 42.4 million-token corpus. It contains 381,292 entries.

Not all the entries in CELEX are suitable for the task of stress assignment. Therefore, some preprocessing is required to cleanup the unsuitable items. First of all, only words with at least three letters are kept. Then entries with the same written forms are removed. This is to ensure that there are no overlapping between training and testing set. However, sometimes words with same written forms can have different stress patterns due to their part of speech. For instance, $pr\acute{e}sent$ is stressed on the first syllable when tagged as a noun, while stressed on second syllable $pres\acute{e}nt$ when classified as a verb. In this case, only one of the two is selected for either training or testing. Moreover, since the focus is on stress assignment for single words, phrases and words with hyphens are removed. Finally, any unpronounceable abbreviations, such as lbw or BBC are also discarded. The final English dataset contains approximately 65K words. When similar preprocessing is performed for Dutch and German, the datasets have about 299K and 296K words respectively.

Another preprocessing step is to move stress markers from phonemes to letters. In CELEX, stress markers are provided in the form of phonetic representations of each word. If a certain syllable is stressed, there will be a stress marker at the beginning of that stressed syllable. To map the stress markers to letters, each stress marker is first moved to the vowel phoneme of each syllable and then mapped to the letters using an automatically generated letter-to-phoneme alignment. Whenever a stressed phoneme is aligned with two consecutive vowel letters (e.g. ee in $meet$), the stress marker is always placed on the first vowel letter.

Finally, some attentions need to be paid to diacritics. A diacritic is a small sign added to a letter to alter its pronunciation. It is well known that there are 26 alphabetic letters in English. However, some languages have more basic letters. For example, in German, some vowel letters have two dots above them: $\ddot{a}, \ddot{u}, \ddot{o}$.

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*

Table 5.1: The steps in our stress prediction system (with orthographic and phonetic prediction examples): (1) word splitting, (2) support vector selection of stress patterns, and (3) pattern-to-vowel mapping.

In CELEX, those letters are represented with original English letters followed by a special symbol such as $\#$. Traditionally, when writing German with English letters, people tend to replace diacritics with some other letters: $\ddot{a} \rightarrow ae, \ddot{u} \rightarrow ue, \ddot{o} \rightarrow ue$. However, doing so is not advantageous to my system. Therefore, I take another convention. That is, stripping off all those special symbols that stand for diacritics.

5.2 System Outline

The stress assignment system takes any input word in written form $\mathbf{w}$ and maps it to the stressed form $\bar{\mathbf{w}}$ with stress markers on any letter bearing the stress. The process is divided into three steps:

(1) First, a word $\mathbf{w}$ is mapped to a sequence of substrings ($\mathbf{s}$).

(2) Then, a particular stress pattern ($\mathbf{t}$) is chosen for each substring sequence, which will be used together with the substring sequence to construct a feature vector. A support vector machine (SVM) is applied to rank the possible patterns for the substring sequence .

(3) Finally, the predicted stress pattern is used to produce the stressed-form of the word $\bar{\mathbf{w}}$. .

Table 5.1 gives examples of words and stress patterns at each stage of the process. Details of the three steps will be presented in the following sections.

5.3 Word Splitting

At the first step of the process, an input word is transformed into its substring representation s :s_1-s_2-...-s_N. The substrings are crucial as they are used to define the feature vector of each input word. While the ultimate goal is to assign stress markers to specific letters, taking substrings instead of individual letters as basic units produce more informative features. Admittedly, there are many ways to split a word into substrings. However, a good splitting scheme is not easy to choose. I will first introduce the methods used in different systems and then compare them with another alternative that is not taken.

SUBSTRING Since stress markers are assigned to individual vowels bearing the stress, it is natural to come up with an algorithm that extracts substrings centered by each vowel letter in a word. In this system, each substring consists of exactly one vowel letter and its neighboring consonants. Given the above description, some issues might come up. First of all, the choice of vowel letters might be arguable. In this thesis, a, e, i, o, u, y are viewed as vowel letters. Although y is not always viewed as a vowel, I do not want to state rules for those special cases in each language, which will make the system less language independent. Another issue that is questionable is why only neighboring consonants are taken. The answer is using short substrings helps with the data sparse problem. For instance, by looking at only neighboring consonants, words $fryer$, and $dryer$ are all transformed to $ry - er$. The learning algorithm then can generalize better from training data. However, there is always a dilemma between generalization and specification.

The substrings are good approximations of real syllables. First of all, each of the substrings has a nucleus, which is the only vowel letter. Secondly, most of them also have pseudo onset or coda, which is represented by the neighboring consonants.

The number of substrings generated by this splitting technique always equals the number of vowel letters in the original input word. However, this number is not always the same as the number of syllables in the word, as in the case $ron - no - un - ce$, which should be divided into three substrings $pro - noun - ce$ according to a perfect syllabification. Intuitively, this has some negative effects on

the performance of the model as the number of substrings produced is always equal to or greater than the true number of syllables and hence increases the length of sequence to be predicted. But in general, it performs well.

ORACLESYL ORACLESYL splits a word according to gold orthographic syllabifications. For instance, the word $pronounce$ is spit as $pro - noun - ce$. The ORACLESYL is regarded as a gold standard. The purpose of using ORACLESYL is to find out the maximum performance of my stress assignment algorithm and to evaluate how much the SUBSTRING splitting method hurts the prediction.

There is also another variation of ORACLESYL, which is to discard the consonants that are not next to any vowels in a syllable. This stratagem shows significant improvement in SUBSTRING. However, the gain for ORACLESYL is not obvious and sometimes even negative.

VOWEL-GROUP VOWEL-GROUP is a splitting scheme used at the initial stage of research that lead to this thesis. In VOWEL-GROUP splitting system, it is assumed that all the syllables have onset, nucleus, and coda. Therefore, each substring consists of one or more vowel letters and their neighboring consonants. Taking the word $pronounce$ as an example, the VOWEL-GROUP will split the word into substring sequence $ron - noun - ce$. At the very beginning, not only the neighboring consonants were taken. Instead, a greedy method was used to include as many consonants as possible till the first vowel was met. For instance, $pronounce$ was spit as $pron - nounc - ce$.

The most significant difference between SUBSTRING and VOWEL-GROUP lies in the number of resulting substrings. While SUBSTRING always produces equal or more substrings than the actual number of syllables, VOWEL-GROUP constantly generates equal or less substrings. This is because VOWEL-GROUP always puts contingent vowels into one substring. For instance the word $reality$ has 4 syllables $re - a - li - ty$. However, it is converted to only three substrings $real - li - ty$ by VOWEL-GROUP. Table Table 5.2 gives three examples of substring sequence generated by three different systems.

Word	System	Substrings
pronounce	SUBSTRING	ron-no-un-ce
	ORACLESYL	pro-noun-ce
	VOWEL-GROUP	ron-noun-ce

Table 5.2: Examples of substring sequence generated by different systems).

Although fewer substrings makes the prediction easier, the issues in choosing a fair evaluation standard finally forced this approach to be abandoned. First of all, it is problematic to evaluate based on the output patterns. For instance, if $reality$ is split into three substrings $real - li - ty$, the stress will be mapped to the first substring $real$. But suppose the gold standard syllabification is available, the stress actually falls on the second substring. Is it justifiable to say that a system makes a correct prediction if the output pattern is $1-0-0$ in the former case? The answer is no. If that was positive, a system can achieve 100% accuracy by just splitting each word into only one substring and always outputs 1 as final prediction, which is obviously self-cheating. A fair evaluation should be independent of splitting method, which is achieved by mapping stress markers to vowels according to output patterns. However, the mapping process introduces a lot of errors in VOWEL-GROUP as it is hard to decide which vowel in a vowel groups receives stress such as ea in $real$ and $reality$.

5.4 Stress Prediction with SVM Ranking

After creating a substring sequence $\mathbf{s} = s_1$-s_2-...-s_N, the next step is to choose an output sequence $\mathbf{t} = \{t_1$-t_2-...-$t_N\}$ that encodes the level of stress for each substring in $\mathbf{s}$. The level of stress is represented by different numbers: '0' for no stress, '1' for primary stress, and '2' for secondary stress.

Normally, such problems can be solved in a traditional sequence prediction framework using HMM , CRF (Lafferty et al., 2001). Those sequence prediction algorithms conduct a search in output space. The output sequence is predicted step by step from the very beginning depending on some input features and transition probabilities. The goal is to find a most likely output sequence for a given input

Word Length		1	2	3	4	5	6	7	8	9	10	11
Number of Patterns	English	1	4	9	14	13	13	11	6			
	German	1	2	3	4	5	6	7	8	8	5	2
	Dutch	1	2	3	4	5	6	7	8	9	7	2

Table 5.3: Number of possible stress patterns for English, German, and Dutch words at different length(measured by the number of syllables).

sequence. However, when it comes to stress prediction, One can take advantages of some important observations of stress patterns and choose the best one as a whole.

The Key observation is: unlike normal sequence prediction problem, where the size of output space is almost unbounded, the number of possible stress patterns is rather limited. In 55k English words, the number of unique patterns observed is only about 70. Within 250k German and Dutch words, the number is only about 50. This observation implies that, after seeing enough stress patterns in training, one hardly encounters a new stress pattern that has never been seen before. Therefore, by considering only the stress patterns that have been observed in training, the system can safely limit the output space to a rather small number of candidates. Given this fact, any input substring sequence only has a small set of possible stress patterns. Table 5.3 presents the number of possible patterns for words at different lengths. The length is measured by the number of syllables.

To choose the correct pattern from a set of possible candidates, we just need to find a function that is able to assign scores to each of the sequence and pick the one with the highest score as the final answer. This can be fit into a ranking framework. While previous works have used various methods to generate a list of candidates (Collins and Koo, 2005), my system simply chooses stress patterns observed in training data as candidates for any substring sequence.

5.4.1 Ranking Formulation

SVM is a supervised machine learning algorithm which requires tagged positive and negative training examples. Given such examples, an SVM will learn a weight vector to separate the positive examples from the negative ones. In this task, an SVM is expected to learn to separate the correct stress pattern from a set of alterna-

tives. Given an input substring sequence $\mathbf{s} = s_1$-s_2-...-s_N of length N, there is only one positive stress pattern $\mathbf{t}$, which is extracted during the word splitting process according to the stress markers provided in the training set. The positive training pair $(\mathbf{s}, \mathbf{t})$ is then augmented with a set of negative training pairs, which consist of all other possible patterns with length N seen in training phase.

After creating those training examples, an SVM is trained to choose the correct pattern $\mathbf{t}$ from a set of candidates $\mathbf{T}$. A linear scoring scheme is adopted, in which the score for each pair is a linear combination of feature-weight products. To calculate the score, each training pair is first mapped to a feature vector $\mathbf{\Phi}(\mathbf{s}, \mathbf{t})$ (see Section 5.4.2). The score for a particular $(\mathbf{s}, \mathbf{t})$ combination is a weighted sum of these features, $\boldsymbol{\lambda} \cdot \mathbf{\Phi}(\mathbf{s}, \mathbf{t})$, where $\boldsymbol{\lambda}$ is a weight vector to be learned by the SVM.

The core idea of SVM is to minimize the norm of weight vecotr $||\boldsymbol{\lambda}||$ subjects to a set of constraints. 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 the other patterns of the same length: $\forall \mathbf{u} \in \mathbf{T}_N, \mathbf{u} \neq \mathbf{t}^j$,

$$\boldsymbol{\lambda} \cdot \mathbf{\Phi}(\mathbf{s}^j, \mathbf{t}^j) > \boldsymbol{\lambda} \cdot \mathbf{\Phi}(\mathbf{s}^j, \mathbf{u}) \tag{5.1}$$

The set of constraints generated by Equation 5.1 are called **rank constraints**(see Section 3.2). 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, $\mathbf{u}$.

At test time, each input word will first be split into a sequence of substrings of length N. Then for each possible stress pattern $\mathbf{t} \in \mathbf{T}_N$, a score is computed by $\boldsymbol{\lambda} \cdot \mathbf{\Phi}(\mathbf{s}, \mathbf{t})$. In the end, the pattern with the highest score will be chosen.

5.4.2 Feature Engineering

One of the most important advantages of using SVM is the ability to use an arbitrary number of features. It has been successful in similar settings learning with thousands of sparse features. Intuitively, the choice of features have a deep impact

on the performance of the ranker. Table Table 5.4 shows the feature templates used to create the features $\mathbf{\Phi}(\mathbf{s}, \mathbf{t})$ for the SVM ranker.

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	$t_1 t_2 \dots t_N$

Table 5.4: Feature Template

The features selected are based on some linguistic theories on syllabic accent. First of all, the $Substring$ features are inspired by the idea that stress is a property of syllables. That is, some syllables are stressed more frequently than others. This has been studied extensively in Zipf (1929), which claimed that stress usually falls on syllables with relative low frequency in the streams of speech and vice versa. Obvious examples include "tion" in English, and "ge" in German. Therefore, if a substring is often stressed in the training data, it is also very likely to be stressed in new contexts. However, the $Substring$ feature alone is not enough to capture the relationships between syllables. As emphasized in Section 2.1, stress is a relative concept. That is, a stressed syllable is phonetically more prominent than its neighbors. To capture this regularity, context features are included.

The most interesting feature might be the pattern feature $t_1 t_2 \dots t_N$. Humans usually prefer certain sequences, which has been demonstrated by research in sequence tagging, machine-translation, and so on. This also applies when it comes to stress prediction. Clopper (2002) showed that people have strong preference to certain stress patterns. I confirm this by extracting statistics of stress patterns from the training data. Table 5.5 is an excerpt from the extracted statistics, showing two most frequently used stress patterns in English for any length shorter than 6 syllables. The percentage indicates the relative frequency in a group with the same length.

From the table it is easy to find that the two most frequently used patterns ac-

Pattern	Frequency	Percentage
1	6256	100.00
1-0	17566	83.74
0-1	2662	12.69
1-0-0	8856	53.90
0-1-0	5171	31.47
0-1-0-0	3075	37.95
2-0-1-0	2246	27.72
2-0-1-0-0	983	34.48
0-1-0-0-0	999	33.93

Table 5.5: Stress patterns and their relative frequency.

count for at least over 60% of the frequency within each group. This phenomenon indicates that it might be a good idea to take the whole pattern as a feature. Note that, in traditional sequence prediction problems, the sequence features are approximated by concatenation of n-gram subsequences. This is because it is usually impossible to enumerate all possible output sequences in structure prediction. However, thanks to the limited number of possible stress patterns, my system can take the whole sequence as a feature, which is more precise and informative.

All the features are binary. That is, if a feature is found in the sequence-pattern pair $(\mathbf{s}, \mathbf{t})$, it will be 1, otherwise 0. For example, if a substring *tion* is unstressed in a $(\mathbf{s}, \mathbf{t})$ pair, the *Substring* feature $\{t_i = 0, s_i = \textit{tion}\}$ will be true. It is the same for the *Context* feature s_{i+1}, t_i. For instance, in English, the penultimate syllable is always stressed if the final syllable is *tion*. Whenever this happens, the context feature $\{t_i = 1, s_{i+1} = \textit{tion}\}$ will be true.

A simple way to verify whether the selected features are informative or not is to look at the learned weights of those features. Figure 5.1 presents the top 20 features with highest weight values. To make it easier to interpret, those features are extracted based on ORACLESYL splitting. The weights were learnt using 55k English words.

The 20 features can give good explanations to some very common examples. For instance, the feature with the highest weight is $t_i = 1, s_i =, s_{i+1} = tion$. This indicates that when the following syllable is $tion$, it is highly possible that its

Feature	Weights
t=1,S(i)=,S(i+1)=tion	33.784
t=0,S(i)=pre,S(i+1)=par	33.462
t=1,S(i)=,S(i+1)=tions	32.157
1-0	32.052
t=2,S(i)=re,S(i+1)=ed	30.672
t=2,S(i)=re,S(i+1)=or	30.337
t=0,S(i)=e,S(i+1)=u	30.015
t=2,S(i)=de,S(i+1)=sen	29.828
t=2,S(i)=trans,S(i+1)=o	29.400
t=0,S(i)=pu,S(i+1)=er	28.246
t=2,S(i)=co,S(i+1)=ter	28.210
2-0-1-0	27.568
0-2-0-1-0-0	27.427
t=0,S(i)=pi,S(i+1)=a	27.115
t=2,S(i)=de,S(i+1)=na	26.482
0-2-0-0-0-1-0	25.969
0-1	25.770
2-0-1-0-0	25.738
t=2,S(i)=de,S(i+1)=car	25.695
t=2,S(i)=de,S(i+1)=sa	25.215

Figure 5.1: Top 20 features with their weights. Learned from 55k training examples based on $oracle$ splitting

previous syllable s_i receives the primary stress regardless what the syllable s_i is. It seems that, in English, the penult syllable is always stressed when the last syllable is $tion$. In these features, one can also find very common prefix such as re, de, and co, which usually receives secondary stress. Moreover, among the top 20 features, there are also very frequently used stress patterns. Each of them is among the top 2 frequently used patterns in the group of the same length. It is also interesting to find that not a single $Substring$ feature appears in the top 20 list. This might support the claim that stress is a relational concept.

5.5 Mapping

In order to make evaluation independent from splitting methods, the final step is to map stress markers back to input words according to predicted stress pattern. For SUBSTRING, stress markers are mapped back to individual vowels. The fact that the number of substrings always equals the number of vowels in the word makes the mapping process straight-forward. Each vowel in original word is associated with exactly one substring and one digit in output stress patterns. For example, if the SVM ranker chooses the stress pattern—0-1-0-0 for substring sequence *ron-no-un-ce*, a correct stress-marked word—*pronóunce*, will be produced. In contrast, if a wrong stress pattern (e.g. 0-0-1-0) is chosen, the final output (*pronoúnce*) will be wrong.

The mapping process for ORACLESYL is slightly different. If stress markers were still mapped to vowel letters, extra errors would be introduced as the system has no way of knowing which vowel letter bears the stress. Instead, with perfect syllabification boundaries, the system simply puts a stress marker in front of a stressed syllable. If a pattern is correctly predicted, there is no problem mapping stress markers back to the right syllable.

5.6 Experiments and Results

In this section, I will evaluate the performance of my system on orthographic stress assignment for three free stress languages: English, German, and Dutch. I will first describe the experiment set up, then introduce different systems to be compared, and provide results and analysis in the end.

5.6.1 Experiment Setup

The English, German, and Dutch data is extracted from CELEX. After the data is cleaned by the methods described in Section 5.1, a random splitting stratagem is taken to split the corpora randomly: 85% for training, 5% for development, and 10% for testing. The development set is used to tune the C parameter for the SVM. Due to much richer morphological changes, the German and Dutch corpora contain

4 times more words than English. To make results on German and Dutch comparable with English, the training, development, and testing set is reduced by 80% for each. The resulting training, development, and testing set for each of the three languages contains around 55K, 3k, and 6k words respectively.

The core component of the system is an SVM ranker. Fortunately, a free and efficient implementation (Joachims, 1999) is available. The version supports learning ranking functions, which is originally introduced in (Joachims, 2002) for applications like search engines and recommender systems.

5.6.2 Comparison Approaches

4 different systems are evaluated on the same data set:

1) SUBSTRING uses the vowel-based splitting method to split words into substrings. Then it trains an SVM as described in Section 5.4.1 and performs predictions based on the learned weights.

2) ORACLESYL splits the input word into syllables according to the CELEX gold-standard. The SVM training and prediction are the same with SUBSTRING. However, the final mapping process is slightly different from SUBSTRING.

3) ORACLEVOWELSPLIT uses gold-standard syllables, but discards consonants that are not adjacent to the vowel. It is very similar to SUBSTRING, except its substrings only include consonants that are within the corresponding syllable, whereas SUBSTRING's substrings may include a consonant from a neighboring syllable.

4) TOPPATTERN is a naive 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. The mapping process is the same as that in the SUBSTRING

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

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
ORACLEVOWELSPLIT	95.8	97.2	96.3	91.8
TOPPATTERN	65.5	67.6	64.1	60.8

Table 5.6: Stress prediction word accuracy (%) on **letters** for English, German, and Dutch. *P*: predicting primary stress only. *P*+*S*: primary and secondary.

The accuracy is evaluated at word level. A word is viewed as correct only if the whole output stressed word $\bar{\mathbf{w}}$ matches the gold standard. The accuracy indicates what percentage of predicted patterns match the gold standard patterns.

5.6.3 Results

The performance on letters (Table 5.6) is quite excellent. SUBSTRING predicts primary stress with accuracy above 95% for English and German, and equal to 91% in Dutch. It is a bit surprising that Dutch has the lowest score as it appears to be the most predictable language in terms of L2P conversion when stress is not considered (Jiampojamarn et al., 2007).

The performance of ORACLESYL is clearly better than the SUBSTRING: constantly around 1% higher for each language. Note that even 1% at this level means substantial reduction in error rate. There are two main factors that might contribute to this: first of all, perfect syllables provide more informative features. For instance, while one of the most common postfixes $tion$ is always treated as one single substring in ORACLESYL, it is always split into two substrings ti and on in SUBSTRING; Secondly, the length of substrings generated by ORACLESYL is shorter, which results in even fewer stress patterns. This can be verified by looking at the number of unique of patterns observed in training set. By examining the number of stress patterns generated from the English training set, I found that there were only 76 unique patterns for ORACLESYL, while the number for SUBSTRING was 103.

I also examined how much an incorrect split of input words can influence prediction accuracy. The phrasing "incorrect" here means that the number of substrings is

not equal to the number of syllables. To investigate this, I conducted a comparison of accuracies between two different cases. In the first case, the number of substring produced by SUBSTRING equals the number of syllables, while in the second case, it is not. Table 5.7 gives an overview of the comparison considering both primary and secondary stress for English.

	$N_{sub} = N_{syll}$	$N_{sub} \neq N_{syll}$	$Total$
$Correct$	3269	2857	6126
$Wrong$	193	230	423
$Accuracy(\%)$	95.4	92.5	93.5

Table 5.7: Comparisons of accuracies between two cases: Number of substrings equals number of syllables; Number of substrings does not match the number of syllables.

When looking at the accuracy when $N_{sub} = N_{syll}$ only, the SUBSTRING even outperforms the ORACLESYL. However, when $N_{sub} \neq N_{syll}$, the accuracy drops significantly. Based on the results from Table 5.7, it can be concluded that majority of loss in SUBSTRING against ORACLESYL results from its inability to split input words in accordance with the true syllable boundaries.

In Section 5.4.2, I talked about the importance of pattern features. Different from how this feature is used in traditional sequence prediction problem, each pattern is viewed as a binary feature in the ranking problem. Similarly, they also have a substantial impact on the performance as shown in Figure 5.2

Another interesting point is to compare performance of ORACLESYL with ORACLEVOWELSPLIT. In using ORACLEVOWELSPLIT, one might expect that shorter syllables help the model generalize better. While ignoring consonants that are not next to any vowels in a syllable helps reduce the error rate by almost 22% for English, the effect is counterproductive for both German and Dutch.

The reported primary stress accuracy in Table 5.6 was calculated by ignoring secondary stress. However, the system was trained for both. An alternative way is to train a system for primary stress only. One can expect the accuracy on primary stress to be improved as predicting both is inherently more difficult than predicting only primary stress. Interestingly, results from Table 5.8 suggest that focusing on

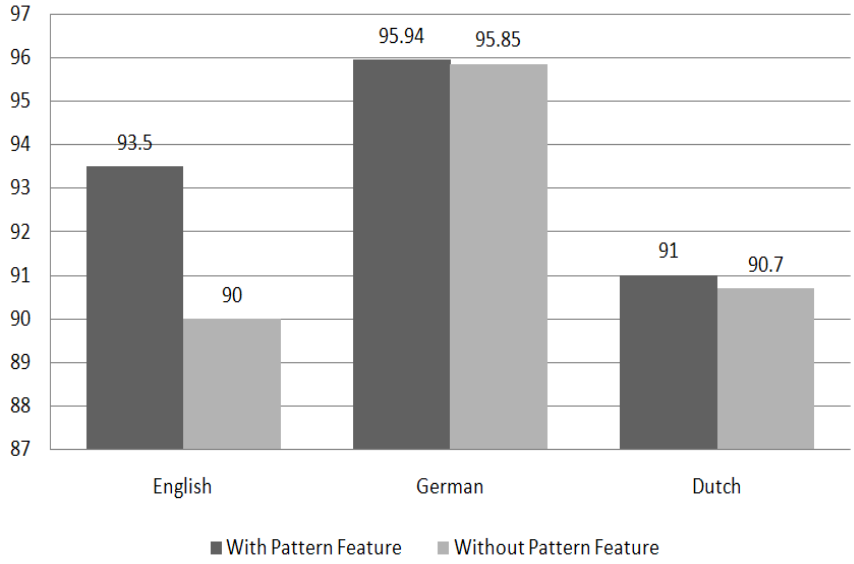

Figure 5.2: Pattern features show benefits for all three languages on letters

primary stress only does not really help improve the accuracy.

System	Predict *P*	Predict *P*+*S*
SUBSTRING	95.1	95.1
ORACLESYL	95.8	96.0

Table 5.8: Primary stress only accuracy (%) on **letters** for English

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. However, the only difference lies in that Demberg et al. (2007) take a different data splitting method. Demberg et al. (2007) found that the word accuracy is much higher if the data is split just randomly. This is caused by overlapping of word stems. A word can have a number of inflections in morphological rich languages such as German. For instance, the verb *arbeiten* can have many inflections such as *arbeite*, *arbeitet*, *arbeitete*, and so on, depending on the personals and tenses. However, all those inflections share the same word stem *arbeiten*. Most of time, such inflections don't change the stress pattern. If an algorithm has seen *arbeiten*

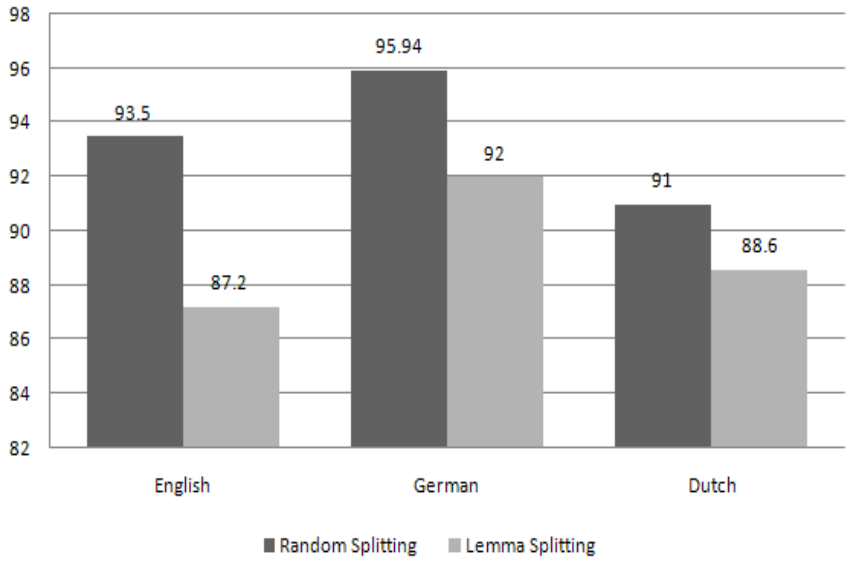

Figure 5.3: Comparisons of random splitting and lemma splitting on three languages for letters. System compared: SUBSTRING, Training Size: 55k.

in training, it is very likely to make a correct prediction for $arbeite$. Demberg et al. (2007) took a more challenging splitting method to make sure that there is no stem overlapping between training and testing. In CELEX, **lemma** is used to refer to word stem. Words sharing the same stem all have the same lemma ID. To make a more convincing comparison, I split the data so that lemma IDs in training and testing are disjoint. With 250k training instances, the SUBSTRING obtains 92.3% accuracy. Furthermore, assuming perfect syllabification, the ORACLESYL has a 94.3% word accuracy, a 40% reduction in error rate.

To find out how much the lemma splitting could influence the performance, I compared results on two datasets for three languages. Results are shown in Figure 5.3. As expected, the word accuracy for each language drops substantially when the training and testing set is split disjointly by lemma ID.

5.6.4 Learning Curves

Finally, I examined the relationship between training data size and performance by plotting learning curves for letter stress accuracy (Figure 5.4). I kept all the developing and testing data as the same size as in Section 5.6.1 but gradually increased the training size for German and Dutch up until 250k.

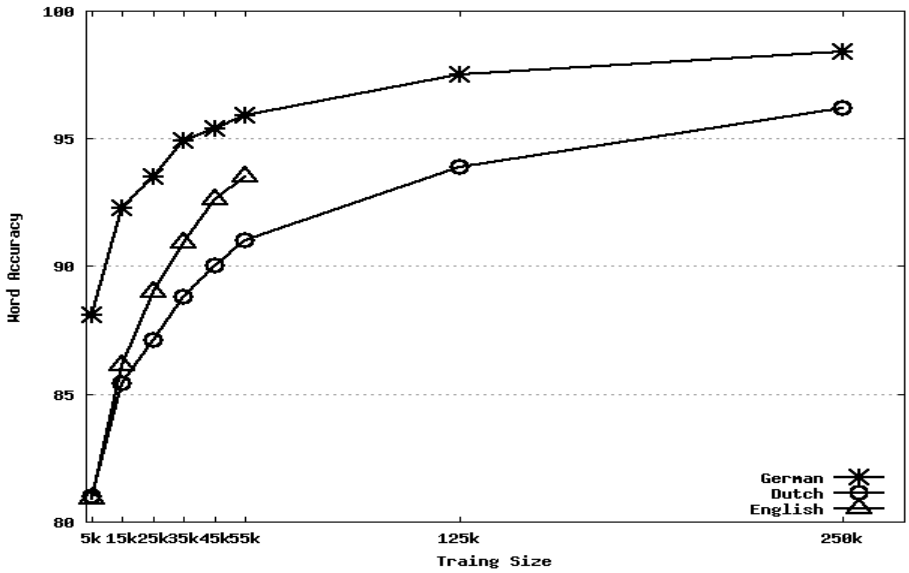

Figure 5.4: Letter stress assignment learning curves for English, German, and Dutch

5.7 Conclusions

Most previous work on orthographic stress assignment either only care about individual letters or have weakness in incorporating certain phonetic constraints. Inspired by Pearson et al. (2000), I introduced a discriminative ranking approach to select a correct stress pattern for each word from a set of alternatives. Instead of working with individual letters, I proposed a way to split input words into substrings that resemble syllables. The algorithm can be easily adapted to real syllables when a good syllabification algorithm or perfect syllabification is available.

To better understand the power of the discriminative ranking approach, this chapter has also conducted a detailed analysis on feature selection The substring and context features are very effective in representing the prominence of individual syllables depending on their neighbors. The pattern feature not only serves as a language independent constraint but also helps select the most likely sequence.

Overall, my system's performance is exceptional, with at least 40% reduction of error rate compared with earlier systems. Moreover, the language independent features allow it to excel in English, German, and Dutch.

Chapter 6

Phonetic Stress Assignment

Stress assignment on phonemes is another task as defined in the problem description. The system takes phonemes as input and assigns stress markers to vowel phonemes bearing the stress. With this task definition, the SVM ranking approach described in the previous chapter can be easily adapted to the phoneme domain.

In this chapter, I will first compare differences between orthographic stress assignment and phonetic stress assignment. Based on that, I will discuss changes made to adapt the system to the phoneme domain. In the end, experiment results and analysis are provided.

6.1 Differences in Phoneme Domain

The task of stress assignment actually becomes easier in phoneme domain. Same as stress assignment on letters, the task is defined as marking specific vowels that bear the stress. Fortunately, in phonetic representations, each syllable only has one vowel phoneme. For instance, the word $pronounce$ is phonetically transcribed as [prənaʊns] in IPA. The two vowel sounds are transcribed as [ə] and [aʊ] respectively. In contrast, the orthographic form has 4 vowel letters. Admittedly, there are more vowel phonemes than vowel letters(e.g. a,e,i,o,u,y) as shown in Figure 6.1. However, the SUBSTRING will still greatly benefit from the convention of the phonetic transcription. First of all, knowing the set of vowel phonemes, the number of substrings generated is always equal to the number of syllables in a word as there

IPA	example	DISC
ɪ	pit	I
ɛ	pet	E
æ	pat	{
ʌ	putt	V
ɒ	pot	Q
ʊ	put	U
ə	another	@
iː	bean	i
ɑː	barn	#
ɔː	born	$
uː	boon	u
ɜː	burn	3
eɪ	bay	1
aɪ	buy	2
ɔɪ	boy	4
əʊ	no	5
aʊ	brow	6
ɪə	peer	7
ɛə	pair	8
ʊə	poor	9
æ	timbre	c
ɑ̃ː	détente	q
æ̃ː	lingerie	O
ɔ̃ː	bouillon	~

Figure 6.1: 24 Englis vowels in DISC format together with example words and corresponding IPA transcription. Adapted from (Baayen et al., 1996)

is exactly one vowel phoneme in each syllable [1]. Secondly, the number of possible patterns also becomes smaller as the average length of the substring sequence becomes shorter. Therefore, I expect better performance on phonemes.

[1]There are also some exceptions. In English, a syllable might not have any vowel phoneme but syllabic consonants. However, those exceptions can be ruled out by referring to a table like Figure 6.2

IPA	example	DISC
ŋ	baco**n**	C
m̩	idealis**m**	F
n̩	burde**n**	H
l̩	dangl**e**	P

Figure 6.2: Englis syllabic consonants in DISC format together with example words and corresponding IPA transcription. Adapted from (Baayen et al., 1996)

Word		Substrings		Pattern		Word'
w	→	**s**	→	**t**	→	$\bar{\mathbf{w}}$
æbstrækt	→	æb-ræk	→	0-1	→	æbstrækt
prisid	→	ri-sid	→	2-1	→	prìsíd

Table 6.1: The steps in the stress prediction system (with examples in phonetic representations): (1) word splitting, (2) support vector ranking of stress patterns, and (3) pattern-to-vowel mapping.

6.2 Adapt SVM Ranker to The Phoneme Domain

The most significant change made to the system is the set of vowels for each of the three languages. In the letter domain, a, e, i, o, u, y are constantly treated as vowels for all the three languages. In the phoneme domain, each language has its own set of vowels. However, besides the changes made to vowel set, no additional changes are made to the system. The system still works in the three-step framework as shown in Table 6.1

Similarly, 4 different systems are compared:SUBSTRING, ORACLESYL, ORACLEVOWELSPLIT, and TOPPATTERN. The choice between different splitting methods becomes much easier as the number of substrings produced by all systems is always equal to the number of actual syllables.

6.3 Experiments and Results

The training, development, and testing data are the same as those used in the in previous chapter except that phonetic transcriptions are used. In CELEX, stress

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
ORACLEVOWELSPLIT	96.7	97.5	96.9	92.8
TOPPATTERN	66.8	68.9	64.1	60.8

Table 6.2: Stress prediction word accuracy (%) on **phonemes** for English, German, and Dutch. *P*: predicting primary stress only. *P+S*: primary and secondary.

markers are put right at the beginning of each syllable. The mapping of stress markers to specific phonemes is straight-forward, given there is only one vowel phoneme in each syllable. The accuracy is also evaluated at word level. Table 6.2 provides results for English, German, and Dutch.

The performance of SUBSTRING is excellent. It achieves 98.0% accuracy in predicting primary stress for English, 97.1% for German, and 93.1% for Dutch. It also predicts both primary and secondary stress in English with high accuracy, 96.2%. Performance is much higher than the baseline accuracy, which is between 60% and 70%.

As expected, the performance on phonemes is always better than that on letters: almost 25% reduction in error rate for each language. The ORACLESYL does not outperform in all three languages anymore as SUBSTRING can also split a word in accordance with the actual number of syllables. Therefore, perfect syllabification might not be needed for stress assignment on phonemes.

The system represents 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%. Bagshaw (1998) obtained 65%-83.3% per-syllable accuracy using Church (1985)'s rule-based system. In contrast, my systems achieves 96.2% *per-word* accuracy for predicting both primary and secondary stress.

Numbers of previously reported word accuracy are also much lower. For predicting both primary and secondary stress, Coleman (2000) reported 69.8% word accuracy. The most comparable work is Pearson et al. (2000), in which the global statistics achieved 81% word accuracy while the joint approach improved it to 87%.

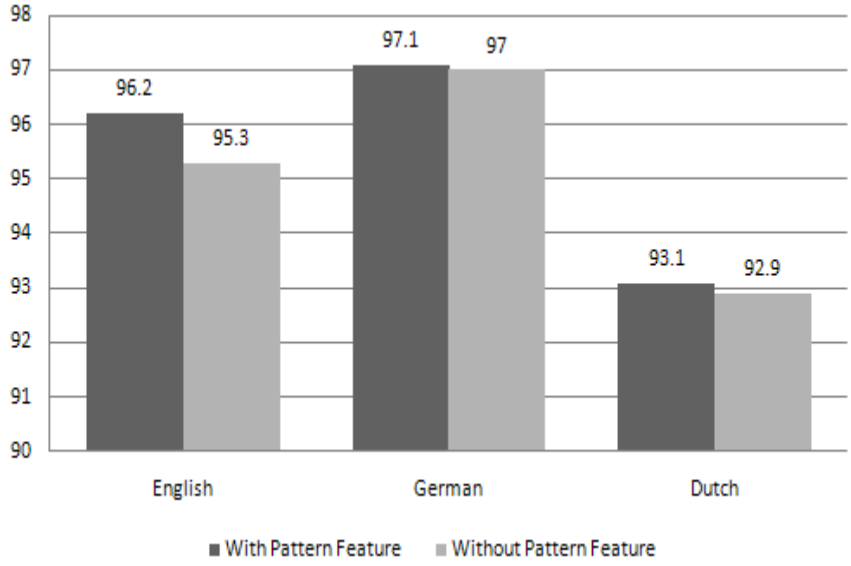

Figure 6.3: Pattern features show benefits for all three languages on phonemes

Same as letters, the stress pattern feature also has a significant impact on the system's performance. Figure 6.3 compares word accuracy on phoneme side with respect to the appearance of pattern feature.

To make all results comparable with those on letters. I also replicated the experiments on lemma splitting set, the results are shown in Figure 6.4. Without much surprise, the numbers on dataset split by lemma id are all lower than on those split randomly.

6.4 Conclusions

I successfully adapted SVM Ranking approach to the problem of stress assignment on phonemes. Except for the minor changes made to accommodate phoneme symbols for different languages, the rest of the system remains the same. The ranking approach to stress assignment clearly outperforms previously reported systems such as Black et al. (1998), Coleman (2000), and Pearson et al. (2000) on English. Moreover, it is again proved to be language independent by showing excellent performance on both German and Dutch.

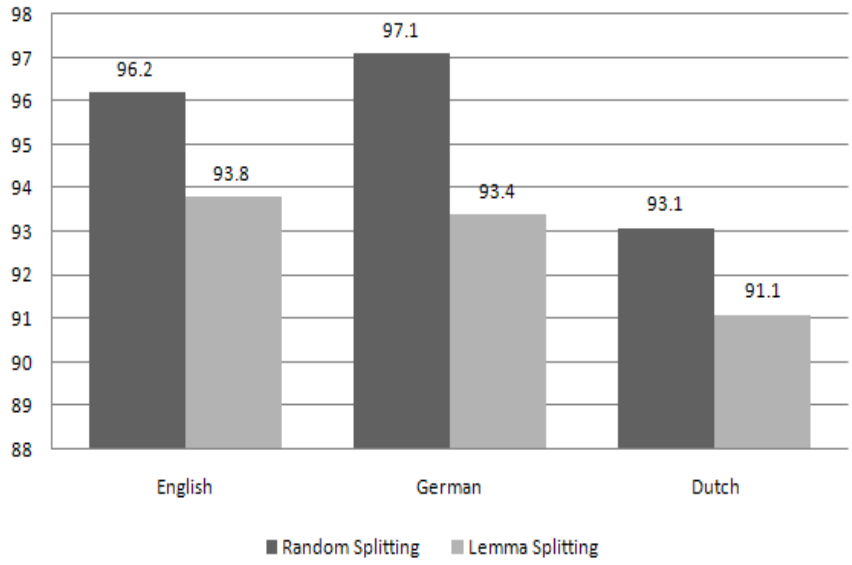

Figure 6.4: Comparisons of random splitting and lemma splitting for three languages on phonemes. System compared: SUBSTRING, Training Size: 55k.

Chapter 7

Combining with Letter to Phoneme Conversion

Improving the stress assignment of L2P systems is one of the main objectives of this thesis. Letter to phoneme conversion seeks to convert orthographic form of words to their phonetic forms, which are represented by a set of phonetic symbols. Lexical stress is important to L2P systems in terms of both naturalness and accuracy of pronunciation. However, I found the problem of stress assignment was overlooked in many existing L2P. 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 have poor performance (Bagshaw, 1998; Coleman, 2000; Pearson et al., 2000). This chapter will discuss how to incorporate my stress prediction model with a current state of art L2P system (Jiampojamarn et al., 2008). I will first briefly introduce the L2P system under exploration, then compare two different ways to incorporate my lexical stress model, and last, show the improvements made.

7.1 The L2P System

The 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 features include all letter n-grams that fit within a context window, HMM-like transition features that express the likelihood of the output phoneme sequence, and linear-chain features that bind the phoneme sequence with the letter n-grams. 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, 2006; Jiampojamarn et al., 2007), this approach does not consider modelling of stress patterns.

7.2 Combining Stress and Phoneme Generation

As discussed in Chapter 4, various approaches have been tried to generate stressed phonemes. First of all, the most direct way is to treat it as a post process. That is, phonemes are first generated regardless of stress. Then a separate stress prediction model will predict the stress based on the phonemes. Many earlier systems took this approach (Bagshaw, 1998; Coleman, 2000). Unfortunately, the performance was poor. Secondly, both van den Bosch (1997) and Black et al. (1998) argued that stress should be predicted at the same time as phonemes. This can be accomplished by simply expanding the output set to distinguish stressed and unstressed phonemes. The joint process appears to be superior to the post process and is adopted in several recent works. For instance, Demberg et al. (2007) produced phonemes, stress, and syllable-boundaries within a single joint n-gram model. Pearson et al. (2000) generated phonemes and stress together by jointly optimizing a decision-tree phoneme predictor and a stress predictor based on counts of stressed and unstressed phonemes. Thirdly, Webster (2004) first assigned stress to letters based on correlations between stress positions and word affixes. In this sense, the input set was expanded. Then the stress markers were used as extra features to predict both phonemes and stress jointly.

Inspired by the above approaches, I adopt two different ways to incorporate my stress prediction model with any L2P Systems. One way is to provide stressed word forms as described in Webster (2004); the other is to generate stress as a post process. In experiments, 4 different systems are compared [1].

1) JOINT: The L2P system's input sequence is letters, the output sequence includes

[1] Dou et al. (2009) described a constraint approach, which filtered the output of the JOINT system by removing phoneme sequences whose stress patterns have never been observed

System	Eng		Ger	Dut
	P+S	*P*	*P*	*P*
JOINT	78.9	80.0	86.0	81.1
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 7.1: Combined phoneme *and* stress prediction word accuracy (%) for English, German, and Dutch. *P*: predicting primary stress only. *P+S*: primary and secondary.

phonemes+stress.

2) POSTPROCESS: The L2P system's input is letters, the output is phonemes. It then applies the SVM stress ranker (Chapter 6) to the phonemes to produce the full phoneme+stress output.

3) 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(Chapter 5) to the input letters as a pre-process.

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

The first approach uses only local information to make predictions (features within a context window around the current letter), which is used to compare with the power of my ranking approach to stress assignment. To compare with previous approaches, I also generated stress and phonemes using the popular Festival Speech Synthesis System [2] (version 1.96, 2004) and reported its accuracy on phoneme+stress.

7.3 Experiments

The experiment data is the same as those used in Chapter 5 and Chapter 6. Word accuracy results for predicting both phonemes and stress are provided in Table 7.1.

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

First of all, 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. Most errors made by the JOINT approach include generating nonsensical stress patterns, such as multiple primary stress or no primary stress. In English, both LETTERSTRESS and POSTPROCESS perform best, while POSTPROCESS is the highest on German and Dutch. Given the above results, it is hard to claim which is the best way to incorporate stress into a L2P system. The answer could depend on the performance of L2P systems, stress assignment algorithms, and languages. 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%.

Unfortunately, I found that the accuracy of phoneme prediction alone is quite similar among all the systems. The gains over the JOINT on combined stress and phoneme accuracy are almost entirely due to more accurate stress assignment. However, 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 pronounced differently depending on their stress status.

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

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 was only 62.1%. Three other papers report word accuracy on phonemes and stress, using different data sets. Pearson et al. (2000) reported 58.5% word accuracy for predicting phonemes and primary/secondary stress. Black et al. (1998) reported 74.6% word accuracy in English, while Webster (2004) reported 68.2% on English and 82.9% in German (all primary stress only). Finally, Demberg et al. (2007) reported word accuracy on predicting phonemes, stress, *and* syllabification on German CELEX data. They achieved 86.3% word accuracy.

7.4 Conclusions

The problem of stress assignment has been overlooked in previous L2P systems. I investigated several different approaches to stress assignment in L2P systems and chose two different ways to incorporate my stress prediction model into a state of art L2P system. Experiment results show that my stress prediction model can greatly improve the performance of the existing L2P system. Moreover, the resulting system also achieves a major advance over all previous systems.

Chapter 8

Conclusions and Future Work

This thesis described a powerful data driven method for automatic stress assignment. Taking benefits from a discriminative ranking approach, my system outperforms all existing systems on English, German, and Dutch for both letters and phonemes.

Viewing stress assignment as a ranking problem, the system naturally takes linguistic constraints on lexical stress into account, avoiding generation of nonsensical stress patterns. When the problem is so formulated, an SVM ranker turns out to be an excellent formalism for choosing the correct stress patterns from a limited set of alternatives. The SVM ranker not only allows for an extensive feature set, but also searches for the optimal solution as long as adequate training data is available.

The substring features are language independent, which allows the system to work without syllabifications. However, I showed that the system could achieve even better result with perfect syllabification. The learnt feature weights also agreed with some previous theories on lexical stress. That is, each syllable has its own probability of being stressed. Moreover, whether it will receive primary or secondary stress depends on the stress probability of other syllables and its location in the word.

When working in tandem with a state of the art L2P system, the system reduced the total error of stress assignment by up to 40%. Furthermore, the combined system also achieved a major advance over all previously reported systems.

There are also several limitations of this work that need to be addressed in the future. First of all, part of speech, which is usually viewed as an important feature in

stress assignment, was not considered in this thesis. Further work needs to be done to include this feature when it plays an important role in a specific task. Secondly, the two approaches that integrated the stress model into a L2P system were simple solutions. From previous research experiences, a joint approach that maximizes probability of both phoneme and stress sequences is expected to give even better results than a pipeline system.

Bibliography

[Arciuli and Thompson2006] Joanne Arciuli and James Thompson. 2006. Improving the assignment of lexical stress in text-to-speech systems. In *Proceedings of the 11th Australian International Conference on Speech Science and Technology*, pages 296–300.

[Baayen et al.1996] Harald Baayen, Richard Piepenbrock, and Leon Gulikers. 1996. The CELEX2 lexical database. LDC96L14.

[Bagshaw1998] Paul C. Bagshaw. 1998. Phonemic transcription by analogy in text-to-speech synthesis: Novel word pronunciation and lexicon compression. *Computer Speech and Language*, pages 119–142.

[Bartlett et al.2008] 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.

[Bisani and Ney2002] Maximilian Bisani and Hermann Ney. 2002. Investigations on joint-multigram models for grapheme-tophoneme conversion. In *In Proceedings of the 7th International Conference on Spoken Language Processing*, pages 105–108.

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

[Chen and Goodman1996] Stanley F. Chen and Joshua Goodman. 1996. An empirical study of smoothing techniques for language modeling. *In Proceedings of Association of Computational Linguistics*.

[Chomsky and Halle1968] Noam Chomsky and Morris Halle. 1968. The sound pattern of english. *New York: Harper and Row*.

[Church1985] Kenneth Church. 1985. Stress sssignment in letter to sound rules for speech synthesis. In *Proceedings of the 23rd annual meeting on Association for Computational Linguistics*, pages 246–253.

[Clopper2002] Cynthia G. Clopper. 2002. Frequency of stress patterns in english: A computational analysis. *IULC Working Papers Online*.

[Coleman2000] John Coleman. 2000. Improved prediction of stress in out-of-vocabulary words. In *State of the Art in Speech Synthesis*, pages 2/1–2/6.

[Collins and Koo2005] Michael Collins and Terry Koo. 2005. Discriminative reranking for natural language parsing. *Computational Linguistics*, 31(1):25–70.

[Colombo1991] Lucia Colombo. 1991. The role of lexical stress in word recognition and pronunciation. *Psychological Research*, pages 71–79.

[Crammer and Singer2003] Koby Crammer and Yoram Singer. 2003. Ultraconservative online algorithms for multiclass problems. *Journal of Machine Learning Research*, 3:951–991.

[Demberg et al.2007] Vera Demberg, Helmut Schmid, and Gregor Moehler. 2007. Phonological constraints and morphological preprocessing for grapheme-to-phoneme conversion. *In Proc. of ACL-2007*, pages 96–103.

[Dou et al.2009] Qing Dou, Shane Bergsma, Sittichai Jiampojamarn, and Grzegorz Kondrak. 2009. A ranking approach to stress prediction for letter-to-phoneme conversion. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*.

[Fudge1984] Erik C. Fudge. 1984. English word-stress. *Allen and Unwin,London*.

[Hayes1981] Bruce Hayes. 1981. A metrical theory of stress rules. *PhD Thesis, MIT, Distributed by Indiana University Linguistics Club*.

[Jiampojamarn et al.2007] Sittichai Jiampojamarn, Grzegorz Kondrak, and Tarek Sherif. 2007. Applying many-to-many alignments and hidden markov models to letter-to-phoneme conversion. In *In Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 372–379.

[Jiampojamarn et al.2008] Sittichai Jiampojamarn, Colin Cherry, and Grzegorz Kondrak. 2008. Joint processing and discriminative training for letter-to-phoneme conversion. In *46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 905–913.

[Joachims1999] 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.

[Joachims2002] Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *KDD*, pages 133–142.

[Joanne and Linda2006] ARCIULI Joanne and CUPPLES Linda. 2006. The processing of lexical stress during visual word recognition: Typicality effects and orthographic correlates. *Quarterly Journal of Experimental Psychology*, pages 920–948.

[Lafferty et al.2001] 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.

[Marchand and Damper2006] Yannick Marchand and Robert I. Damper. 2006. Can syllabification improve pronunciation by analogy of english? *Natural Language Engineering*, pages 1–24.

[O'Connor and Trim1953] Joseph Desmond O'Connor and John L.M. Trim. 1953. Vowel, consonant and syllable- a phonological definition. *Word*, pages 103–122.

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

[Prince1977] Mark Liberman; Alan Prince. 1977. On stress and linguistic rhythm. *Linguistics Inquiry 8*.

[Rogier C. van Dalen and Rothkrantz2006] Pascal Wiggers Rogier C. van Dalen and Leon J. M. Rothkrantz. 2006. Lexical stress in continuous speech recognition. In *In Proceedings of Interspeech 2006*, pages 2382–2385.

[Selkirk1982] Elisabeth O. Selkirk. 1982. The syllable. *In H. van der Hulst and N. Smith (Eds.), The structure of phonological representations: Part II*, pages 337–383.

[Spencer1996] Andrew Spencer. 1996. Phonology theory and description. *Blackwell Publishers Inc.*

[Tagliapietra and Tabossi2005] Lara Tagliapietra and Patrizia Tabossi. 2005. Lexical stress effects in italian spoken word recognition. In *XXVII Annual Conference of the Cognitive Science Society*, pages 2140–2144.

[van den Bosch1997] Antal van den Bosch. 1997. *Learning to pronounce written words: A study in inductive language learning*. Ph.D. thesis, Universiteit Maastricht, The Netherlands.

[Wang and Seneff2001] Chao Wang and Stephanie Seneff. 2001. Lexical stress modeling for improved speech recognition of spontaneous telephone speech in the jupiter domain. In *in Proc. of EUROSPEECH*.

[Webster2004] Gabriel Webster. 2004. Improving letter-to-pronunciation accuracy with automatic morphologically-based stress prediction. In *In INTERSPEECH-2004*, pages 3573–2576.

[Williams1987] Briony Williams. 1987. Word stress assignment in a text-to-speech synthesis system for british english. *Computer Speech and Language v2*, pages 235–272.

[Zipf1929] George Kingsley Zipf. 1929. Relative frequency as a determinant of phoentic change. *Harvard Studies in Classical Philology*, pages 1–95.

Appendix A

Implementation Details

The implementation details and usage of the system are presented in three main steps as discussed in Chapter 5.

A.1 Word Splitting

A.1.1 Letters

Two different ways were presented to split words into substring sequences: SUBSTRING and ORACLESYL. They have been implemented by *letterexamples.pl* and *lettersybexamples.pl* respectively.

The *letterexamples.pl* does not only generate substring sequences but also stress patterns. It converts stressed word forms to their corresponding stress patterns and substring sequences as shown below.

$$pro'nounce \quad \rightarrow \quad 0-1-0-0 \quad ro-no-un-ce$$

The input contains stressed word forms, with a single quote sign ' for primary stress and double quote sign " for secondary stress. If the word does not contain any stress markers, the output stress patterns will be all 0s. The following is a sample command for running the script.

$$cat \quad inputfile \quad | \quad ./letterexamples.pl \quad > \quad outputfile$$

The *lettersybexamples.pl* generates both stress patterns and substring sequences on the basis of perfect orthographic syllabification. Therefore, a syllabification

	SUBSTRING	ORACLESYL
English	ephonexamples.pl	ephonsybexamples.pl
German	gphonexamples.pl	gphonesybxamples.pl
Dutch	dphonexamples.pl	dphonsybexamples.pl

Table A.1: Scripts for splitting words in their phonetic forms).

database needs to be prepared according to the format as shown in the following example.

$$pronounce \quad pro-noun-ce$$

The script takes maximum 3 arguments to run. Following is a sample command.

$$./lettersybexamples.pl \quad databasefile \quad inputfile \quad (chunk) \quad > \quad outputfile$$

The first argument is the name of the syllabification database file. The inputfile should contains words in their stressed forms. The last argument is optional. When it equals to "chunk", consonants not neighboring any vowel letters in a syllable will be discarded.

A.1.2 Phonemes

The above two scripts work for all three languages. However, since each language has its own set of phonetic symbols, different scripts have been written. They are presented in Table A.1 according to their purposes.

The scripts in the SUBSTRING column can be run as the following example:

$$cat \quad inputfile \quad | \quad ./ephonexamples.pl \quad > \quad outputfile$$

The inputfile should contain phoneme strings with stress markers.

The scripts in the ORACLESYL column can be run as:

$$./lettersybexamples.pl \quad inputfile > \quad outputfile$$

The inputfile contains syllabified phonetic strings with stress markers: $pr@-n6n-s$. The hyphen stands for the syllable boundary.

Arguments	Explanation
-r	Set 1 to re-create the feature vectors.
-c	Regularization parameter of SVM
-k	Kernel, 0 for now
-t	Amount of data used for training : but 0 means use all.
-s	Name of data sets to use. Ordered as: training development testing

Table A.2: Explanations for SVM ranker arguments

.

A.2 SVM Ranking

The example files generated by word splitting corpus are used to train an SVM ranker. First of all, features are extracted for training, developing, and testing by a shell program—$featureExtraction.new.sh$. The following command gives an example.

$$./featureExtraction.new.sh \quad trainfile \quad devfile \quad testfile$$

The $trainfile$, $devfile$, and $testfile$ files are outputs of word splitting scripts discussed in the previous section.

Then an SVM ranker is ready to be trained and tested by running *runSVM.sh*. The files passed to this script have to be the same as those passed to *featureExtraction.new.sh*. The arguments passed to an SVM can be modified by changing *runSVM.sh*. Details are explained in Table A.2.

A.3 Mapping

The mapping process transfers unstressed form of words into their stressed form according to output stress patterns. Based on the definition of SUBSTRING and ORACLESYL splitting method, mappings are straightforward. Each number in the output stress patterns corresponds uniquely to either a single vowel letter or a syllable of the input words depending on the splitting method used.