Towards text-based prediction of phrasal prominence

Teemu M. Kuusisto

Master's thesis
UNIVERSITY OF HELSINKI
Department of Computer Science

Helsinki, April 23, 2015

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Laitos — Institution — Department
Faculty of Science	Department of Computer Science

Tekijä — Författare — Author
Teemu M. Kuusisto

Työn nimi — Arbetets titel — Title

Towards text-based prediction of phrasal prominence

Oppiaine — Läroämne — Subject
Computer Science

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Master's thesis	April 23, 2015	72

Tiivistelmä — Referat — Abstract

The objective of this thesis was text-based prediction of phrasal prominence. Improving natural sounding speech synthesis motivated the task, because phrasal prominence, which depicts the relative saliency of words within a phrase, is a natural part of spoken language. Following the majority of previous research, prominence is predicted on binary level derived from a symbolic representation of pitch movements. In practice, new classifiers and new models from different fields of natural language processing were explored. Applicability of spatial and graph-based language models was tested by proposing such features as word vectors, a high-dimensional vector-space representation, and DegExt, a keyword weighting method. Support vector machines (SVMs) were used due to their widespread suitability to supervised classification tasks with high-dimensional continuous-valued input. Linear inner product and non-linear radial basis function (RBF) were used as kernels. Furthermore, hidden Markov support vector machines (HM-SVMs) were evaluated to investigate benefits of sequential classification. The experiments on the widely used Boston University Radio News Corpus (BURNC) were succesful in two major ways: Firstly, the non-linear support vector machine along with the best performing features achieved similar performance than the previous state-of-the-art approach reported by Rangarajan et al. [RNB06]. Secondly, newly proposed features based on word vectors moderately outperformed part-of-speech tags, which has been inevitably the best performing feature throughout the research of text-based prominence prediction.

ACM Computing Classification System (2012 CCS):
Computing methodologies → Natural language processing
Computing methodologies → Supervised learning by classification
Computing methodologies → Support vector machines

Avainsanat — Nyckelord — Keywords
machine learning, data mining, feature extraction, phrasal prominence, spatial language models

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Contents

1 Introduction

Phrasal prominence is an aspect of spoken language which depicts the relative saliency of words within a phrase. Being part of prosody, prominence has a multitude of functions describing elements of language that are encoded by neither grammar nor choice of vocabulary, for instance, emphasizing important terms, introduction of new terms, expressing contrasts, resolving ambiguities and guide the dialogue structure. Phrasal prominence modeling benefits a wide range of applications. Constructing a fluent natural sounding speech synthesizer is just impossible without accurately modeling prominence. Moreover, analyzing prominence from speech signal benefits automatic speech recognition and speech-to-speech translation.

This thesis concentrates on predicting accurate instances of phrasal prominence based on written text. The objective specifically is to support applications of speech synthesis in increasing their subjective naturalness. This is an optimization problem, since instead of a single correct manifestation of prominence, there usually exist a bunch of suitable alternatives that arise from different interpretations of the text. Phrases in different contexts or considered from different point of view bear different prominence.

Supervised machine learning techniques are used to classify prominence. That is, models are trained from a set of text documents which are prelabeled with a discrete symbolic representation of prominence. Humans are said to be able to hear four distinct levels of prominence but there seems not to be consensus among researchers about finer-grained separation than the existence versus absence of prominence. Therefore, like most of the previous work, this thesis treats the problem as a binary classification task.

Minimal supervision and maximal generalizability of proposed techniques are the prevalent philosophies in the research of this thesis. Although the classification of prelabeled text is a supervised task, those labels still need to be generated either by human experts or by an unsupervised machine learning approach. Hand-labeling is time consuming work even for experts. Accordingly, the required amount of work to extract text-based features is multiplied by every language they are applied to. For instance, part-of-speech tags, the most successful feature so far, are based on categories

constructed by humans and a language-dependent set of hand-labeled data is required to train an automatic labeler. Obviously, successful selection of more generalizable features reduces need to develop new features for new languages. Statistical machine learning approaches are expected to be more appropriate across language boundaries compared to constructions that rely on human interpretation in greater extent.

Research of phrasal prominence has continued over decades. In its infancy, linguistic research searched for endless amount of deviating cases by hand to build rule-based classification systems. The research area revolutionized in 1990s due to development of various intermediate representations of prosody, increasing availability of speech corpora, and recent advances in machine learning. Ever since, a multitude of studies have attempted prominence detection and prediction based on acoustic events and text-based features separately or jointly. The community has seen a variety of algorithms from decision trees to maximum entropy models, while evolution in extraction of text-based features has been rather minimal. Features derived from part-of-speech tags have remained evidently the most superior approach thus far.

Contributions of this thesis are: evaluation of three new classifiers, proposal of various existing statistical and graph-based language models for this specific problem, and investigation of different ways to scale and preprocess input features. Support vector machines (SVMs) are evaluated with linear and radial basis kernels. Further, hidden Markov support vector machines (HM-SVMs) are evaluated to investigate the benefits of predicting sequences of phrasal prominence instead of prediction for individual words. The most remarkable result is the newly proposed features based on word vectors, a continuous-valued spatial language model, is shown to achieve better performance than the part-of-speech based counterparts. It is also noticed that even very simple features such as word lengths perform astonishingly well.

The thesis is organized as follows. Linguistic background is first explained in Section 2.1 from the computer scientist's point of view — expecting no prior knowledge of the field and motivated by rather inconsistent and confusing use of terminology in previous work. Then, the problem of phrasal prominence

prediction is defined with more details in Section 2.2. Motivation for this problem is discussed in Section 2.3.

Related work is surveyed in Section 3 covering not only text-based prediction task but also prominence detection based on acoustic events. The background is encompassed more comprehensively than would have been necessary to consider the experimental part. Nevertheless, the provided bigger picture is thought to be worthwhile due to so tightly related tasks. The most relevant studies that perform text-based prediction are outlined in Section 3.2, and Section 3.4 sums up the most important notions supporting the research in this thesis.

Section 4 reviews construction of proposed phrasal prominence models. Selection of previously studied and construction of the newly proposed features are outlined in Sections 4.1 and 4.2 respectively. Support vector machines and its extension to sequential classification are described and their use is discussed in Section 4.3. Features and classifiers are fitted together by testing various preprocessing and scaling techniques, which are described in Section 4.4.

Experiments are explained in Section 5. Selection of the Boston University Radio News Corpus as a data set is discussed in Section 5.1. The data is separated into an experiment set for model tuning and a held-out set for validation As described in Section 5.2. Evaluation is done in speaker-independent way with 6-fold cross validation. Section 5.3 explains selection of features and tuning of classifier parameters step by step. The results are then shown in Section 5.4.

Finally, results and future directions are discussed in Section 6, and conclusions are outlined in Section 7.

2 Problem

The thesis aims at analyzing written text to accurately predict the level of words' saliency as a part of natural speech. This section describes the necessary linguistic preliminaries, defines the targeted problem and discusses the motivation for such research. The section of preliminaries intends to clarify some rather ambiguously used terms, and give a quick introduction to

the linguistic basis of the problem for computer scientists.

2.1 Preliminaries

The primary purpose of speech is to convey a message via a sequence of words, but it contains a wide range of additional linguistic and paralinguistic information as well [Cha08]. The speaker's identity, gender, emotional state and other speaker-specific features are examples of paralinguistic information, whereas linguistic information inherently supports understanding the message conveyed by an utterance.

Prosody is an information layer of spoken language that bears supplemental linguistic contents pertaining to a conveyed message [Cha08]. In speech signal it appears as intonation, rhythm and pauses. It has a multitude of functions describing elements of language that are encoded by neither grammar nor choice of vocabulary. Prosody may resolve ambiguities [Gro83], guide the dialogue structure [Bol78], emphasize important terms, express contrasts, outline the focus of discourse, specify the type of the utterance (statement, question, or command), indicate the presence of irony or sarcasm. Furthermore, it can assist demarcating word and phrase boundaries, and in some languages distinguish between phonetically similar sounds [Leh72].

The precise way in which the term prosody is defined varies among researchers [Cha08]. The least agreed aspect of the term concerns the abstractness versus concreteness of prosody. At one end are those who define it as an abstract structure that organizes sound so that it is not coupled with any realization. This is opposed by those who use the term to refer to the realization itself, that is, to use it as a synonym for measurable prosodic features of the speech signal. Most of the researchers probably fall in between these two extremes by considering a linguistic structure coupled with its realization.

A more comprehensive definition given by Cutler et al. [CDv97] states that prosody is: "the linguistic structure which determines the suprasegmental properties of utterances." Prosodic features are said to be suprasegmental because they affect over the boundaries of phonemic segments. On the contrary, the sequence of words of an utterance, for example, is encoded in speech

signal as a sequence of phonemic segments. The domain of prosodic features ranges from syllables, up through words, phrases, sentences, paragraphs, to whole discourses. Furthermore, acoustic features of prosody are relative in nature and tend to require reference to the time-series of the speech to be meaningful [Jak67, Lad96]. As a result of the suprasegmental nature and the relativeness against time-series, modeling prosody is a more complex problem than modeling segments of phonemes with absolute acoustic parameters.

Prominence is an aspect of prosody which depicts the relative emphasis of a syllable within a word, or a word within a phrase [Roa02]. In respect to words that is, a word is said to be prominent, if it noticeably stands out of the rest of a phrase. Despite the decades of extensive research, there exist many areas of disagreement and a lack of understanding, which in turn is visible as inconsistent and confusing use of terminology. Accentuation and stress are rather ambiguously used multifunction terms describing prominence.

Accent has two different meanings. As related to prominence, it means the phonetic prominence of a syllable or a word. That is, a syllable is made prominent by movements of pitch. Confusingly, the term **pitch accent** is also widely used referring more specifically to the usage of pitch. Based on the existence of this term, it seems that accent might have a wider meaning than only the prominence produced by phonological variation. Alternatively, accent may refer to the different ways in which speakers sharing the same vocabulary and grammar pronounce the language. In this sense, accent is distinguished from dialects — different variations of a language which usually differ in vocabulary or grammar as well.

Stress denotes the speaker's relative effort of producing syllables or words [Roa02]. The term is used rather synonymously with accent, though some divergence exist too. Stressed syllables or words are signaled for instance by better articulation, greater intensity and longer vowels. Confusingly, stress is also said to be audible as pitch prominence which was the case with accent. Many writers have suggested that the term accent should be considered as some of the manifestations of stress — particularly pitch prominence. Nevertheless, despite the wide usage of the word, it has not seemingly acquired distinctive meaning of its own.

Stress is divisible into two separable aspects: *prosodic stress* and **word stress** [CDv97]. Prosodic stress, synonymously *phrasal stress* or *sentence stress*, denotes the prosodic prominence of words. The level of stress given to a word in some context comes from prosody. This abstract representation has granularity of a word and rules of word stress needs to be considered to form the concrete representation of spoken stress.

Word stress, or **lexical stress**, concerns the way in which syllables are stressed within words. Some languages have **fixed-stress**, where the stressed syllable has a fixed position or its location is constrained by a simple rule. For example, in Finnish the first syllable bears stress, and in Polish the stress is positioned on the penultimate syllable. Other languages have *variable-stress* in which case the word-internal positioning of stress is truly lexical: it must be learned for each individual word as part of its pronunciation. English and Russian are examples of such languages.

Prosodic realizations are continuous-valued and highly dependent on the individual speaker's style, gender, dialects and other phonological factors. Therefore, utilizing such representation directly to analyze prominence is in lesser extent useful. The non-uniform factors of prosody need to be eliminated while prosodic patterns of interest should somehow be represented in an interpretable way.

To overcome these complexities, a number of symbolic or parametric intermediate representation schemes of prosody have been developed. Such schemes include: Tones and Break Indices (ToBI) [SBP$^+$92], TILT intonational model [Tay98], Fujisaki model [FH82], Intonational Variation in English (IViE) [GNF98], and International Transcription System for Intonation (INTSINT) [HIV94]. These prosodic labeling approaches provide a common vocabulary and framework for researchers to characterize prosody, allow transcriptions of speech corpora and enhance the comparison of research results. From computational point of view, these frameworks act as a discretization of more complex continuous data. Anyway, it has been shown that some applications benefit from avoiding intermediate categorical representations by directly using prosodic correlates of raw or normalized speech signal.

The Tones and Break Indices framework is a symbolic intermediate rep-

resentation of intonation and breaks developed by a large group of experts from multiple research sites during four workshops held in 1990s. It is based on Pierrehumbert's intonational phonology [PH90]. A significant majority of corpus-based computational research has been built upon it, which was the intention of its creators and increases its importance. The framework was initially developed for American English, but later it has been extended to other languages. It was targeted to support different kinds of research. On the other hand, the area was only partially understood at the time of development and hence attention was paid not to build a too complex or complete system. Two decades have passed since the creation of ToBI.

The ToBI framework contains a representation of pitch movements more than a representation of prominence. Nevertheless, the information about prominence can be extracted to some extent from those tones that are located at the stressed syllables of words. However, it should be noted that there is no consensus of how the prominence should be represented and even how many different classes of prominence that representation should include. The ToBI framework is used in the experiments of this thesis. The framework is further discussed in Section 5.1 along with the conversion of the pitch movements to acquire labels of phrasal prominence.

The reasonable number of phrasal prominence levels is a significant and still open question. Binary representation describing existence versus absence of prominence is widely used, but other alternatives exist as well. Mehrabani et al. [MMC13] hypothesize that the number of levels is optimal, when every pair of the levels are perceptually distinct from each other. On the other hand, too few levels would leave important information out. Mehrabani et al. [MMC13] experiment this by clustering segments of text based on prominence, which is realized as pitch, duration and energy. They take advantage of previous studies on Just-Noticeable-Differences (JND) of speech prosody to indicate which vectors in the feature space are perceptually distinct. By increasing the number of clusters and calculating the distances between the cluster centroids, they show that four clusters are enough to achieve the wanted distances. Furthermore, the conducted perceptual experiment studying the naturalness and expressiveness of synthesized speech shows no

statistically significant superiority between models with three and four levels. However, the results trend towards the four levels of prominence.

2.2 Definition

The objective in this thesis is to predict phrasal prominence for a discourse as a sequence of words. This is approached by techniques of data mining. That is, the predicting algorithm is statistically learned from recordings of speech as spoken language is the natural source for prosody. Strictly speaking, this thesis omits direct analysis of speech data and instead concentrates on modeling prominence based on input texts. However, these are quite tightly coupled problems and therefore, although not included in the experiments, analysis of speech is still introduced, included in the literature review and discussed. Consideration of word-internal prominence is excluded to restrict the complexity of the problem and because it is possible to solve word prominence separately given the information of phrasal prominence.

As whole, the general problem is divisible into two subproblems from which the latter one is the primary interest in this thesis:

1. Prominence detection: Analyzing the acoustic correlations of speech data to **detect** language-dependent usage of phrasal prominence and represent it in terms of an intermediate representation.

2. Prominence prediction: Utilizing the detected representation and corresponding text documents to statistically learn a model for **predicting** phrasal prominence.

As a prerequisite, there must exist a representation for the phrasal prominence detected from speech. The outcome would either be a continuous-valued representation or a categorical discretization of phrasal prominence. Utilizing a discrete symbolic intermediate representation enhances the decoupling of the subproblems compared to more complex continuous-valued counterparts. It might be desirable to avoid choosing a representation with meaningless intricacy as the field of phrasal prominence is still somewhat poorly understood.

Empirical research results show that humans tend to be able to separate four distinct classes of phrasal prominence [MMC13]. Thus there seems to be no need of finer-grained information for a system aimed to be observed by humans. In this thesis, a categorical representation is chosen because of its simplicity and sufficiency. This choice turns the whole problem — and both of the subproblems — into a classification problem, where each word of an input speech or text is assigned with a prominence class.

The prominence detection task is mandatory to generate knowledge base and hence be able to solve the second problem of predicting phrasal prominence, which is based solely on text-based information. In practice, the result of the prominence detection phase is a corpus of text labeled with assignments of phrasal prominence classes. Unsupervised learning approaches are necessary to achieve such classification of the acoustic events in speech signal. Of course, knowledge generation could be performed manually by human experts, but this is a far too slow process for creation of bigger corpora. If there already exists a sufficiently large manually collected corpus, it could possibly be extended in supervised manner to generate a larger one.

Prominence prediction is a supervised classification problem, where, given a sequence of words, the objective is to find the most probable sequence of prominence labels according to the learned data. This is formally defined by a probabilistic objective function in Definition 1. Almost equivalent, definitions are found throughout the surveyed research. The probabilistic way of defining the objective function gives a formal, but very general definition. It is not directly applicable to the algorithms in this thesis. However, it eases observations and reasoning about which dependencies are relevant to this problem.

Definition 1. Probabilistic objective function for prominence prediction: Let $C = \{c_1, \ldots, c_n\}$, $n \in \mathbb{N}$, be a set of discrete symbols that quantize phrasal prominence. Let $W = (w_1, \ldots, w_N)$ be a sequence of words representing a text document of length N. Let $L = (l_1, \ldots, l_N)$, where $l_i \in C \forall i$ such that $0 \leq i \leq N$, be a sequence of assigned phrasal prominence classes. Furhter denote by ϕ a mapping from a word to a feature vector. Let $k \in \mathbb{N}$ denote a number of preceding and succeeding words in approximating

context.

Now, given the text W and the input mapping ϕ the objective function $J_{W,\phi}\colon C^N \to \mathbb{R}$ is:

$$J_{W,\phi}(L) = P(L \mid W) \tag{1}$$

$$\approx \prod_{i=1}^{N} p(l_i \mid W) \tag{2}$$

$$\approx \prod_{i=1}^{N} p(l_i \mid \phi(w_1), \cdots, \phi(w_N)) \tag{3}$$

$$\approx \prod_{i=1}^{N} p(l_i \mid \phi(w_{i-k}), \cdots, \phi(t_{i+k})) \tag{4}$$

Then, prediction of phrasal prominence is an optimization problem, where the optimal solution is a label sequence L^* that maximizes the objective function:

$$L^* = \arg\max_{L \in C^N} J_{W,\phi}(L).$$

The objective function is outlined as a function of a sequence of phrasal prominence labels given a sequence of words and a mapping from words to feature vectors with numerical attributes. The function represents conditional probabilities of label sequences given a constant input text, which makes the situation equivalent to modeling joint probabilities of those two sequences as the prior probabilities for constantly defined words are uniformly distributed. The objective is to find a feasible solution that maximizes value of the objective function.

Without any constraining assumptions the probability shown in Equation 1 is very hard to solve. In most of the previous research, prominence labels are assumed to be independent and identically distributed (i.i.d.). That is, given any two labels $l_i, l_j \in L$, the posterior probabilities of these labels are independent and they can be calculated separately: $P(l_i, l_j \mid W) = P(l_i \mid W) \cdot P(l_j \mid W)$. This leads to the situation described in Equation 2, where prediction of a single label depends only on the word sequence. In practice, such assumption is made, when a simple supervised model, such as decision tree, support vector machine or maximum entropy model, is being applied.

The independence assumption is rather dubious, because phrasal prominence tends to express relative saliency of words within a sentence. There

might be several possible sequences of prominence for one sentence, and each word of the sentence might be considered prominent in one of those alternatives. Nevertheless, making all of the words or none of them prominent would be very improbable. Several sequential classifiers have been previously evaluated to take the dependencies of nearby prosody labels into account. These include: hidden Markov models combined with decision trees [RO96] and conditional random fields [GA04, NLX11]. The experiments of this thesis continue the list by one more model, hidden Markov support vector machine.

Another simplifying approximation of the objective function is defined in Equation (4). This approximation relies on the assumption that a symbol in the sequence L does not significantly depend on all of the vectors in T but instead it depends only on some of the nearest words. One solution would be to consider the data within a context of a sliding window worthwhile. The formal definition introduces the variable k which depicts the window size as the number of preceding and succeeding words. Practically, this is taken into more complex level in this thesis as different features have different window sizes and the numbers of preceding and succeeding words are modeled with two distinct variables.

The performance of experimented models is evaluated by measuring accuracy and F-measure. These are commonly used measures in previous work and evaluation of machine learning models in general. Accuracy simply denotes the number of correctly predicted words divided by the number of all predicted words. It is easy to interpret and compare but its descriptiveness is also confined.

F-measure considers both precision p and recall r of the tests. Precision p denotes the number of true positives (TP) — samples correctly predicted as positive — divided by the number of samples predicted as positive — true and false positives ($TP + FP$). While recall (r) denotes the number of true positives divided by the number of true positives and false negatives (FN) — all the samples that should have predicted as positive. Thus, when precision approaches one, the number of words incorrectly predicted as prominent decreases, whereas increase of recall means decrease of the number of words that were incorrectly predicted as non-prominent. Formally, precision is

defined as $p = \frac{TP}{TP+FP}$, and recall as $r = \frac{TP}{TP+FN}$. Finally, the F-measure is computed as an unweighted harmonic mean of precision and recall: $F_1 = 2\frac{p \cdot r}{p+r}$.

2.3 Motivation

The primary motivation arises from the importance of phrasal prominence as a layer of additive information in spoken language. Phrasal prominence supports spoken language by emphasizing important words, outlining contrastiveness, supporting overall understandability of discourse and even resolving ambiguities in conveyed messages. Especially spontaneous speech is usually not as well-formed as written text in which case the sole sequence of words and choices of vocabulary are more likely to be incoherent or even ambiguous. In such conditions, phrasal prominence, and prosody in general, plays more important role. There are multiple applications capable of benefiting from these advantages. Basically, phrasal prominence is either produced to support synthesized speech or analyzed to support speech recognition and understanding.

When looking at text-to-speech synthesizers, phrasal prominence is perceived by human perception. Hence, the most important goals for this application are the naturalness and interpretability of the synthesized speech. State-of-the-art speech synthesizers are capable of pronouncing distinct words quite well, but synthesizing phrases or even longer unrestricted texts is the current bottleneck in natural sounding synthesis. Phrases synthesized without phrasal prominence appear to be rather monotonous, and making too many words stressed is similarly bad or even worse solution.

To achieve a more fluent synthesis, phrasal prominence needs to be predicted based on features extracted from text. Obviously, this is an optimization problem, because any input text can be correctly spoken out with differing interpretations of prominence. The input text may even contain ambiguities that are unresolvable in written text but would have been unambiguous in the corresponding spoken utterance. These difficulties are well expressed by Bolinger's [Bol72] pessimistic view that one needs to be a mind reader to predict accentuation, which is accepted by most of the researchers. Although text-based prediction of phrasal prominence has its limitations, those mind-

reading attempts have continued over decades resulting in more accurate solutions. It must be noted that humans are able to speak out any piece of text with reasonable layer of prosody even if they really do not understand the content.

Another application is speech recognition where the analysis of phrasal prominence is performed by computers. The purposes of such analysis are rather similar to those with human perception. It might help with understanding the conveyed message of a spoken utterance recalling that the phrasal prominence provides additional information. For instance, knowing which words are emphasized or de-emphasized could enhance extraction of keywords.

Performance of automatic speech recognition (ASR) can be improved in many ways by taking into account the effects of phrasal prominence. Humans tend to pronounce some syllables more clearly than others. This phenomenon is strongly related with prominence. Stressed syllables are articulated with greater intent, and therefore are more reliably correctly recognized. Further, this allows better guesses about the identity of unstressed syllables against vocabulary.

Prominence might support boundary detection task as well. For instance, considering languages with fixed-stress — like Finnish in which the stress is always on the first syllable of a word — make it possible to utilize the detection of stressed syllables to improve the detection of word boundaries. Note that boundary detection task significantly benefits from analysis of phrasal boundary tones as part of prosody while the improvement described here rely on the phrasal prominence.

On the other hand, assuming a language with lexical-stress (e.g. English) gives yet another way to improve word detection [AN07]. Assume that the speech recognition system in question produces a list of best candidate words for a word being recognized. Now, if the word being recognized is given prominence, these hypotheses can be rescored due to their lexical stress patterns which are available in word pronunciation dictionaries. The perceived sequence of word prominence statuses can be compared to the lexical stress patterns of each candidate word. Scores of the candidate words are decreased, if their lexical stress patterns do not equal the perceived

patterns. Considering phrasal prominence for similar purposes could improve selection of hypotheses consisting of multiple words even further. This could be accomplished by comparing which words ought to bear prominence against the observed phrasal prominence. Experimental results conducted by Ananthakrishnan and Narayanan [AN07] show modest but still statistically significant reduction in word error rate (WER) of 1.3% (relative) compared to the baseline recognition system. Their results were achieved through comparison of the lexical stress patterns alone.

In speech-to-speech translation a spoken utterance of a source language is translated and spoken out in another language. This combines the two previous applications. A typical state-of-the-art speech-to-speech translator first utilizes automatic speech recognition to produce text in the source language [RNB06]. This is followed by feeding the text to a translator and then to a natural speech synthesizer of the target language. Here, the prosody of the output is predicted from the target language and therefore the information contained in the prosody of the source speech is lost.

The true interest towards speech translation has arisen from more sophisticated approach, where the prosody of the source language is also recognized and translated to the other language [RNB06]. Of course, due to differences in usage of prosody this is not possible between every language. Even in such cases improving speech understanding on the source-side could improve the process of translation. Currently, the key requirement towards such solutions is better understanding and reliable representations of prosody.

3 Related work

Research of phrasal prominence has continued over decades. In its early years, linguistics researched phenomena by hand — searching and identifying the endless amount of deviating cases. The methods of the whole research area revolutionized in 1990s due to construction of various intermediate representations of prosody, which in turn allowed creation of prosody labeled speech corpora. Since the corpus-based statistical approaches — benefiting from advances in computer science — have evolved and become a common

method in linguistic research. The great impact of the computer science in such research is indicated, for instance, by Hirschberg [Hir93] who mentions that creation of a hand-crafted set of rules aimed at phrasal prominence prediction took several months of intensive work compared to the automatically generated decision tree with even better performance.

Many different approaches to detect and predict phrasal prominence have been attempted during the previous two decades. These attempts vary in many aspects such as: used language and information sources, preferred output and its granularity, applied algorithms, selected dataset and arrangements of the evaluation and validation.

The desired outcomes of the approaches vary from the binary existence of phrasal prominence or pitch accent to multi-class representations determining the type of the accentuation as well. The assignments are made for a domain of words or syllables, from which the latter describes more information than is required in terms of phrasal prominence. Accordingly, words, syllables, vowels or short-term frames are considered as the granularity for the sequences of the input data. In the majority of the experiments, pitch accents and boundary tones are detected or predicted together but that is not covered here.

Surveyed research is restricted to consists only of experiments targeting prominence of English. Several speech corpora are used to provide training and test datasets. The Boston University Radio News Corpus (BURNC) is used in most of the research reviewed here, and it is the most widely used corpus for this specific task in general to the best of the author's knowledge. Use of this corpus is assumed in the experiments described in this section unless another corpus is explicitly specified.

The survey covers such a large amount of technologies that it was decided to exclude any further explanations. However, the most relevant parts are described later, and other details are available in the referred literature. The survey is separated into three parts according to the used input data: prominence detection concerning only acoustic correlates of prosody is considered first, followed by prediction approaches utilizing text-based features, and finally the detection approaches combining both of the information sources. The surveyed literature is summarized and the most important concepts are

explained in Section 3.4.

3.1 Acoustic detection

Early attempts to detect phrasal prominence based on acoustic correlates involve analysis over short-term frames along with experimenting with Hidden Markov Models (HMMs). Such ideas were first introduced by Chen and Withgott [CW92] who applied the HMM model to features based on smoothed pitch and intensity. Wightman and Ostendorf [WO94] approached detection of prosodic patterns by combining two models. They used a decision tree to provide estimates of probability distributions for the observations, and a discrete hidden Markov model for sequential modeling. Speaker-dependent experiments for detection of syllable-level existence of prominence were reported to achieve accuracy of 86% on a subset of the BURNC corpus. Conkie et al. [CRR99] approached syllable-level binary pitch accent detection by using hidden Markov model for speaker normalized pitch and energy values. They used a ToBI labeled data set consisting of only 1166 words, and reported accuracy rate of 82.8%. Ananthakrishnan and Narayanan [AN05] assume that acoustic correlates of prosody consist of multiple streams of information that are further assumed to be correlated but not always synchronous. They address correlatedness and asynchrony by applying coupled hidden Markov model (CHMM) to prosody, which is realized in three streams of energy, pitch and durations. Experiments were reported to achieve accuracies of 72.03% and 73.97% for word-level and syllable-level respectively.

Maghbouleh [Mag96] adapts a logistic regression model for syllable-level binary prominence detection. The usage of products-of-sums (POS) model is reasoned by rather easy computability related to earlier models and need for quite small data set. In experiments, features such as energy, identity of nearest phonemes, lexical stress and position measures result with accuracy of 86%. Thus, the model is said to achieve 68% of the possible accuracy between the baseline of always deaccenting with 69% accuracy and the human performance of 94%. Ostendorf and Ross [OR97] propose a stochastic modeling framework for syllable-level detection of prominence. The framework is based on pitch, energy and duration features along with segmental characteristics of

syllable sequence. Sun [Sun02] applies ensemble machine learning algorithms to classify syllables into four classes of prominence: high, down-stepped high, low and unaccented. Here, classification and regression trees (CARTs) are aggregated with methods of bagging and boosting to detect prominence based on the fundamental frequency contour, pitch targets, energy and segmental duration. The experiments — conducted with the data from one female speaker (F2B) from the Boston University Radio News Corpus — show that the decision tree alone achieves an accuracy of 82.89%, whereas after either bagging or boosting the accuracy is 84.71%. Rosenberg and Hirschberg [RH07] experiment various filtered energy based predictors for binary prediction of word-level prominence. Their best performing two-stage classifier tested on Boston Direction Corpus (BDC) and Topic Detection and Tracking (TDT-4) was able to detect prominence in read (BDC-R), spontaneous (BDC-S) and broadcast news speech at 84.0%, 88.3% and 88.5% accuracy, respectively. The first stage of the classifier involves extraction of energy-based features from multiple frequency sub-bands, and the second stage attempts to correct the predictions with pitch and duration features. Finally, a majority voting classifier is used for the corrected predictions. Chen et al. [CHC04] pursue more generalized approach in terms of intra-speaker and inter-speaker variation. They apply a Gaussian mixture model (GMM) to pitch, energy and duration features, which are preprocessed by principal component analysis (PCA). The leave-one-speaker-out evaluation task of the model resulted with 77.34% accuracy for binary pitch accent detection at the syllable level. Chan [Cha08] applies a maximum entropy model to solve binary pitch accent detection problem. He uses Locality Preserving Projections (LPP) — a linear dimensionality reduction technique — to preprocess the set of acoustic features. An accuracy rate of 87.25% for word-level detection is achieved with less than half of the original dimensions, and it performs even slightly better than with the original feature set. Thus, the LPP method seems to provide more robust model due to noise reduction and reduces the computational cost. Chan further discusses usage of raw acoustic features to support speech recognition without using any symbolic intermediate representations.

Several experimental results advocate use of neural networks to solve this

problem. Ananthakrishnan and Narayanan [AN08b] compare binary pitch accent detection performance of Gaussian mixture model with 18 components and two-layer feed-forward neural network. Their experiments show that the neural network performs better achieving a syllable-level accuracy rate of 74.10% (compared to 72.18%) with speaker-independent five-fold cross validation. Likewise, Jeon and Liu [JL09] compare performance of various algorithms and contribution of features from four classes: pitch range, energy range, pitch slope and duration. In their experiments, neural networks outperform decision trees, Gaussian mixture models and maximum entropy models by achieving a syllable-level accuracy rate of 83.53%. Moreover, experiment conducted by Ni et al. [NLX11] support the superiority of neural networks against decision trees. In their results, the presented neural network detects syllable-level binary pitch accent with 83.95% accuracy, compared to their decision tree with accuracy of 81.45%.

Continuous wavelet transform (CWT) is a widely used mathematical tool for analyzing and visualizing various simultaneous temporal scales of a signal. It has been successfully used for several applications of speech analysis. Vainio et al. [VSA13] apply continuous wavelet transform to analysis of speech prosody — especially prominence. They apply the CWT to intonation in form of the fundamental frequency contour, and conclude that the local maxima at the level of prosodic words correlate strongly with the perceived prominence judged by listeners. The further direction of the research is suggested to include development of visualization and analysis tools, discretization of the CWT result for higher level applications, applying CWT to other prosodic features and studying the human auditory processing against CWT analysis.

Mehrabani et al. [MMC13] discard use of the ToBI by automatically constructing their own intermediate representation for phrasal prominence. They apply an unsupervised K-means clustering algorithm with varying number of clusters to find the optimal number of prominence levels. As their goal is to enhance the naturalness of speech synthesis, they hypothesize that the number of levels is optimal if the levels are perceptually distinct. They use Just-Noticeable-Difference (JND) — the smallest perceivable difference between two levels of a sensory stimulus — to pitch, energy and duration of

the resulting cluster centroids. They show that with four levels of prominence all of the features differ less than the corresponding experimental JND value. Furthermore, they provide results of a perceptual experiment where speech was synthesized with different number of levels. The experiment did not show statistically significant results between 3-level and 4-level models, albeit the results trend towards the 4-level model.

3.2 Text-based prediction

Text-based phrasal prominence predictions utilize various syntactic and lexical features. Hirschberg [Hir93] presents a hand-derived set of rules and an automatically generated decision tree to predict binary phrasal prominence. She models the discourse structure with a global focus and a local focus each filled with word roots during the analysis. This model is aimed to predict whether a word in the text is determined as given — already discussed in the discourse — or new otherwise. Other features in her experiments were part-of-speech tags, broader word classes derived from POS tags, surface position information and complex nominal analysis. The experiments were conducted on several rather small corpora, one of them being FM Radio News Corpus (FM-RNC) — a predecessor of the BURNC corpus. The experiments achieved an accuracy of 82.4%, which was reported to be much better than the simple function word versus content word distinction used in earlier speech synthesizers. Ross and Ostendorf [RO96] extend this approach by using a hidden Markov model in conjunction with the posterior probabilities approximated by a decision tree. They utilized a multi-stage approach in which the pitch accent placement is predicted first followed by pitch accent type prediction and phrasal boundary intonation assignment. Predictions are based on features derived from part-of-speech, word's new/given status, lexical stress, prosodic phrase and paragraph structures. According to their evaluation with a single speaker (F2B), such model achieves pitch accent location prediction accuracy of 87.7% at syllable-level and 82.5% at word-level. Conkie et al. [CRR99] utilize a stochastic finite-state transducer (FST) to estimate the joint probabilities of part-of-speech tags and binary pitch accent to provide the most probable sequence of binary pitch accent labels.

Their experiment evaluated with only 1166 words resulted with accuracy of 84.0% at syllable-level. Sun [Sun02] attempts to enhance the performance of decision tree with ensemble machine learning methods — bagging and boosting. Predictions are made at syllable level for four types of pitch accent: high, low, down-stepped high, and unaccented, which were used also by Ross and Ostendorf [RO96]. The features considered in the experiments are: vowel identity, syllable stress, syllable positions within a word, word position within a sentence, the number of syllables in the current and previous word, part-of-speech, and combination of the POS tag and syllable stress. The experimental results show only minor improvements as the bagging and boosting decision trees achieves accuracies 80.64% and 80.50%, respectively, compared to the baseline decision tree with an accuracy of 80.47%. The results were evaluated with only a single speaker F2B of the BURNC corpus.

Gregory and Altun [GA04] propose use of conditional random fields (CRFs) to phrasal prominence prediction for conversational speech. Furthermore, they introduce a bunch of new predictors categorized as syntactic, probabilistic and phonological variables. The probabilistic variables — aimed to incorporate the information content of a word and collocation measures — include the unigram word frequency, the probabilities of a word given the preceding and succeeding words separately (bigram frequencies), and the two joint probabilities of a word with its preceding and succeeding words separately. The syntactic information consist of four classes derived from part-of-speech tags, whereas the phonological predictors involving rhyme and timing include: the number of syllables and phones, the length of the utterance and the word position in it. Experiments performed on Switchboard Corpus show that their discriminative model can predict world-level prominence with 76.36% accuracy. Phrasal prominence prediction for conversational speech is also targeted by Nenkova et al. [NBK$^+$07] who introduce and evaluate a new feature - accent ratio. The accent ratio is the probability for a word to be accented given its identity. The words with insignificant number of appearances in the data are detected with binomial test cut-off, and their probabilities are set to 0.5. Accent ratio is compared to other features — namely unigrams, bigrams, word givenness, stopwords, TF-IDF, TF-IDF2, ... — and all possible subsets of the

features with decision tree. The experiments on the Switchboard Corpus show that accenting words with accent ratio being greater than 0.38 outperforms other single feature classifiers with an accuracy of 75.59% — compared to unigram with an accuracy of 73.77%. Even the subsets of the other features do not perform much better than that. Nevertheless, as the ratio of words not in the accent ratio dictionary increases, the performance presumably decreases because each unknown word is given a probability of 0.5 leading to its accentuation.

Chen et al. [CHC04] propose use of multi-layer perceptron (MLP) to word-level prediction. Their neural network is trained using standard error back-propagation algorithm. The considered feature set is simple — consisting of part-of-speech tags and the number of syntactic phrases a word initiates and terminates. The experiments are evaluated with four speakers (F1A, F2B, M1B and M2B) of the BURNC corpus by leave-one-speaker-out cross-validation, which is only 3-fold as the speaker F2B is reported being always contained in the training set. The speaker-independent evaluation results with 82.7% accuracy. Note that the results are not validated with any held-out data and the number of round in the cross-validation is low. Rangarajan et al. [RNB06] apply maximum entropy model to text-based prediction of word-level prominence. They introduce use of supertags as additional feature beside part-of-speech information and content versus function word status obtained from POS tags. The speaker-independent experiments show that this model predicts binary prominence with 85.22% accuracy, and the use of supertags show only marginal improvement. These results are evaluated with the same speakers and cross-validation procedure that Chen et al. [CHC04] used.

Jeon and Liu [JL09] show performance comparison of multiple algorithms to predict syllable-level binary classification of prominence. Part-of-speech, syllable identity, lexical stress and binary word boundary indicator were used as lexical and syntactic features to test applicability of decision tree, neural network, maximum entropy model, and support vector machine. The experiment shows that support vector machine with polynomial kernel outperformed others achieving a prediction accuracy of 87.92%. The best results

were obtained by analyzing POS, syllable identity, lexical stress and word boundaries extracted from the preceding and succeeding contexts of length 2. Moreover, Ni et al. [NLX11] compare the applicability of decision tree, support vector machine and conditional random fields to the same problem at the syllable level. They utilize the number of syntactic phrases a word initiates and terminates as two additional features to the set used by Jeon and Liu [JL09]. Here, conditional random fields beat the other algorithms in the conducted experiments with 88.54% accuracy rate, which is quite near to the performance of the support vector machine perceived by Jeon and Liu [JL09]. Confusingly, in the experiments of Ni et al. [NLX11], their support vector machine (84.28% accuracy) performs worse than decision tree (86.34% accuracy).

3.3 Combined acoustic and text-based models

Models that combine analysis of acoustic and text-based features are constructed in various ways. Some of them are based on the simplifying assumption that the probabilities of text-based and acoustic features are conditionally independent given the labels of prosody. Such models consist of different models applied to different information sources learned separately. Alternatively, this assumption is discarded and the same model is used for both sources. Nevertheless, such approaches could combine different models to further improve the performance.

Conkie et al. [CRR99] combine acoustic-prosodic hidden Markov model with stochastic syntactic-prosodic model. They report an accuracy of 88.3% that was evaluated with a very small data set. Ananthakrishnan and Narayanan [AN05] combine a acoustic-prosodic coupled hidden Markov model and a language model based on the syntactic information of part-of-speech tags. The language model utilizes back-off trigram LM to supply joint probabilities of part-of-speech tags and pitch accent labels. They report a word-level accuracy rate of 79.50% with 13.21% false positives, and respectively, a syllable-level accuracy rate of 74.84% with 17.34% false positives. The performance of the combined model does not improve much from the syntactic-prosodic model with 79.70% accuracy at the word level, and like-

wise the improvement from the syllable-level acoustic-prosodic model to the combined one is less than a percentage (absolute).

Sun [Sun02] — targeting pitch accent detection of four classes — compares the effects of bagging and boosting to the performance of decision trees with acoustic, syntactic and lexical features. In contrary to the minor improvements reported for the separate models, the combination of the input sources is reported to improve the detection accuracy from 84.26% to 87.17% due to utilization of the boosting method, while bagging decision trees achieves 86.89% accuracy. Chen et al. [CHC04] apply an acoustic Gaussian mixture model along with syntactic artificial neural network to syllable-level prominence detection. Their combined acoustic-syntactic model is reported to achieve 84% accuracy, when evaluated in speaker-independent way. Ananthakrishnan and Narayanan [AN08a] use maximum a posterior framework, and the assumption that acoustic and syntactic-lexical features are conditionally independent given the sequence of prosodic labels. Hence, they combine the neural network based acoustic model with n-gram based syntactic and lexical prosodic language model by production of the probabilities assumed independent. The experimental results show that the presented approach achieves binary detection of syllable-level pitch accent with an accuracy of 86.75% and a false positive rate of $8,08\%$. They also provide the corresponding word-level accuracy and false positive rate of 84.59% and $9,33\%$, respectively.

In order to support speech-to-speech translation, Rangarajan et al. [RNB06] combine syntactic-prosodic maximum entropy model with acoustic-prosodic hidden Markov model to detect phrasal prominence. The acoustic-prosodic model is based on discretized features — including deltas and second order deltas — derived from pitch and energy contours over 10ms frames. According to the experimental results, this approach achieves binary classification accuracy of 86.01% at the word level.

In addition to syntactic-prosodic and acoustic-prosodic models, Jeon and Liu [JL09] provide experimental results for a combined classifier, which models the acoustic-prosodic component with a neural network and the syntactic-prosodic component with a support vector machine. The two models are

combined as a maximum likelihood classifier assuming the independence between acoustic and syntactic observations given the sequence of prosodic labels. Their evaluation shows a binary pitch accent classification accuracy of 89.8% at the syllable level. Fernandez and Ramabhadram [FR10] apply conditional random fields restricted to first-order chains to detect existence of pitch accent at the word level. They use a single model with a large set of fully automatically extracted acoustic and linguistic features that are quantized with K-means clustering. The feature set includes several ratios based on the accent ratio but for different features. They explore simple unsupervised approaches to adapt the model for data without labels, and further experiment the effects of reducing the amount of training data to the results. The experimental results are concluded to show robustness of the presented model with the best F-measure of 83.5%. Moreover, for small training data sets the performance is improved by the adaptation, albeit for larger training set the effect might be negative. Ni et al. [NLX11] propose a complementary model to pitch accent classification. Unlike in the majority of the research referred here, Ni et al. [NLX11] discard the assumption of conditional independence between syntactic, lexical and acoustic observations given the prosody. Features from different sources are modeled together instead of using different models for different information sources. Nevertheless, different models are combined to construct a complementary model. The reported experimental results show that the complementary model combining Boosting CART* and CRFs is able to classify pitch accent with 91.40% accuracy, while the CRFs achieve 90.40% accuracy alone.

Gonzalez-Ferreras et al. [GEVC12] present a multi-class classifier of pitch accent based on syntactic and acoustic features. They divide the more complex problem of multi-class classification into several subproblems of pairwise binary classification. The multi-class model is then built by combining those pairwise classifiers. Each pair-wise classifier is a coupled classifier consisting of a decision tree and a neural network. The reported experimental results show that the approach achieves an accuracy of 70.8%, when classifying word-level pitch accents into eight classes.

3.4 Summary

Research concerning text-based prediction is the most essential part and directly used in the experiments of this thesis. The other research is surveyed to give more complete view of the targeted problem as whole. Table 1 summarizes the reviewed attempts towards text-based prediction of prominence.

Paper	**Corpus**	**Classes**	**Model**	**Accuracy**	
				Word	**Syllable**
Hirschberg [Hir93]	FM-RNC	2	decision-tree	82.4	-
Ross and Ostendorf [RO96]	FM-RNC	2 (and 4)	HMM, CART	82.5	87.7
Conkie et al. [CRR99]	news (ToBI)	2	stochastic FST	84.0	-
Sun [Sun02]	BURNC (F2B)	4	Boosting CART	-	80.50
Sun [Sun02]	BURNC (F2B)	4	Bagging CART	-	80.64
Gregory and Altun [GA04]	Switchboard	2	CRFs	76.36	-
Chen et al. [CHC04]	BURNC	2	MLP	**82.67**	-
Rangarajan et al. [RNB06]	BURNC	2	MaxEnt	**85.22**	-
Nenkova et al. [NBK+07]	Switchboard	2	accent ratio	75.59	-
Jeon and Liu [JL09]	BURNC	2	SVM	-	87.92
Ni et al. [NLX11]	BURNC	2	CRFs	-	88.54

Table 1: Summarization of text-based prominence prediction approaches. The emphasized results are compared to the new experimental results as described in this section.

Comparability of the reviewed studies is restricted by the many differences in the targeted problem definitions and conducted experiments. Most of

the incomparability arise from differing output granularity from word to syllable level and the number of output labels. Another influential factor is the used corpus, and even the same data set can be organized in differing ways to train and test the models. For instance, only few studies involved use of a held-out validation set to ensure generalizable results. Use of cross-validation appears more commonly but the number of the validation rounds vary. Furthermore, some of the experiments are carried out in speaker-independent way, where the training and test sets never contain data from a common speaker [CHC04, RNB06].

There are three major aspects to be compared to previous results: features, classifiers and their combinations as complete models. Most of the research has targeted prediction of binary level placement of prominence while only few studies provide evaluation of prediction performance on word-level. Including those studies with binary phrasal prominence prediction on word-level and excluding those that used very small data sets, and mostly lacked cross-validation, results with a total of four studies [GA04, CHC04, RNB06, NBK+07]. The decision of using news data from the BURNC corpus constrains the comparable set further leaving only two studies of Chan et al. [CHC04] and Rangarajan et al. [RNB06]. Therefore, the experimental results of this thesis are compared against these two previous studies that are carried out with the same subset of the BURNC corpus as described in Section 3.2. Note also that Rangarajan et al. [RNB06] report the overall best results for word-level prediction.

The comparable studies perform cross-validation in a speaker-independent manner, which is argued by increased generalizability. This sounds reasonable because those studies aim to support speech recognition and speech synthesis [CHC04] or particularly speech-to-speech translation [RNB06]. Note that following the speaker-independence in this thesis makes it more comparable to those experiments. The only exception is that they don't use held-out validation set.

What comes to features, different constructions of word classed based on part-of-speech tags appear to be the most widely used and distinctly the most efficient approach previously. This observation is common for studies of both

syllable and word level prediction. Exceptionally, accent ratio is reported to achieve better performance with the Switchboard corpus [NBK+07]. However, as mentioned, this feature is measured for each word separately and therefore is not very efficient for previously unseen words. The same study shows good performance for unigrams — a simple statistical measure, which motivated consideration of statistical approaches and features simple as possible.

4 Phrasal prominence modeling

This section describes the construction of the studied phrasal prominence models. A model consists of one or more features extracted from input text, preprocessing and scaling techniques applied to these features and a supervised machine learning classifier. Generalizability and minimal amount of human-intervention were the most emphasized criteria when selecting those techniques.

Selection and extraction of the features is covered in two sections. Previously experimented features are first described and rationalized in Section 4.1 followed by the newly proposed features in Section 4.2. Some of the previously well-performed features were chosen due to two reasons: to achieve viable comparisons between new and state-of-the-art features, and being able to properly evaluate the algorithms relying on a good basis of the data representation. Spatial and graph-based approaches were assumed appropriate based on the author's personal beliefs, and therefore new ideas were searched from the literature addressing applications of information retrieval, keyword extraction and language modeling techniques. Such models were thought to fit well with the considered criteria.

For most of the features, the feature vectors are constructed from a context of nearby words. This is necessary, because word prominence is conditionally dependent from its context as demonstrated in Definition 1 (sec. 2.2). Consideration of contexts is common in the literature and is also experimented here with varying context sizes. The context size is considered here to be a part of feature's configuration, which allows different contexts for different features.

Proposed models are learned from data with support vector machines and hidden Markov support vector machines, which are discussed in Section 4.3 in details. Several preprocessing techniques and value scaling functions were used to adjust the extracted features to work properly with the chosen algorithms. Those are described in the final Subsection 4.4.

4.1 Features drawn from literature

A wide range of syntactic, semantic and lexical features have been experimented in previous work from which part-of-speech tags, unigrams, bigrams, word's position within a sentence and word's givenness status were selected to be evaluated in this thesis.

Part-of-speech (POS) tags represent the human-derived word classes such as nouns, adjectives and verbs. In the literature, part-of-speech tags [Hir93, RO96, Sun02, NBK+07, RNB06, JL09, NLX11] and its derivatives such as the content versus function word separation [Hir93, RNB06] or some other broader word classes [GA04, NBK+07] are the most frequently used and the best performing type of features. POS tags are used in this thesis, because, according to the literature, they alone provide a good basis for modeling prominence, covering prominence's syntactic aspects. Furthermore, as state-of-the-art feature, it gives a reasonable baseline for comparison for new features. The downside is that, although they can be predicted automatically with supervised machine learning modeling, prior knowledge is required about the target language. That is, the word classes must be derived and a set of data has to be labeled by hand for each target language.

POS tags form a categorical feature with one category per word except the clitized words, where two words are emerged together and there is a category for both of them. For example, pronoun *it* and verb *is* could form a cilitized word *it's*, which would then get categorized into both of the mentioned word classes. The POS tags are taken from the BURNC corpus, where they have been automatically generated and hand-corrected. Following general conventions of data mining, the categorical attribute is transformed into n binary attributes, where n is the number of the categories and each binary value denotes assignment of the specific category.

More strictly speaking, three approaches were considered for the conversion of POS tags into a binary vector. In one approach, the POS tags available in the BURNC corpus were directly mapped to binary attributes as is, which means separate attributes for each combination of the categories of clitized words. In the other two approaches, categories of clitized words were encoded by two attributes. The third approach is further structured by splitting POS tags into major word classes and modifiers that are then encoded as binary attributes. For instance, nouns (NN), adjectives (JJ) and verbs (VB) are considered major word classes, while plural, denoted by suffix S, is an example of a modifier. Hence, a plural noun would be represented with the two attributes corresponding to noun and plural. Adding such structure to the binary interpretation of the categories reduces the dimensionality of the produced feature vectors. The last approach were chosen based on the reduced dimensionality and experienced slightly better performance results. The complete set of classes contained NN, IN, NP, DT, JJ, VB, RB, CC, CD, TO, PP, MD, POS, WP and RP, while the modifiers were W, R, S, Z, G, D, N, P and $ resulting with a total of 24-dimensions. The constructed binary vectors were finally normalized to unit norm.

The other syntactic features used in literature include but do not limit to: stopwords [NBK+07], accent ratio [NBK+07], surface position information [Hir93], complex nominal analysis [Hir93], and supertags [RNB06]. Furthermore, location of words have been expressed by word position within a sentence [Sun02, GA04], length of a sentence [GA04] and the number of syntactic phrases a word initiates and terminates [NLX11].

Word positions within sentences are evaluated in this thesis too due to the marginal improvement shown in the previous research. It is also a very simple feature and trivial to implement. Lengths of sentences were not used as the author expected that it would evoke sparsity in the used data set with relatively small number of sentences. But instead, another positional indicator that measures the **relative position within a sentence** was considered.

Semantic features proposed in previous work aim at extracting the salient topics, focus contrastiveness and other discourse-level information. Prominence emphasizes introduction of new topics, and thus word givenness has

been repeatedly considered. That is, word existence in the preceding context has been checked in deterministic way [Hir93]. The **word's givenness status** is represented here as a single binary value — denoting whether a word has appeared earlier in the discourse or not. The search is done in case-insensitive way, but no other preprocessing steps are taken. This feature is further compared to a newly proposed continuous-valued counterpart that is based on cosine similarity of word vectors.

Previously, semantic information has also been accumulated from statistical data of the language such as: unigrams and bigrams [NBK+07]. In addition to these background statistics, discourses have also been analyzed in terms of TF-IDF and TF-IDF2 measures [NBK+07].

Unigrams and bigrams, generally N-grams, are frequencies of sequences of N adjacent words in text. **Unigram** (1-gram) and **bigram** (2-gram) frequencies, taken from the United States section of the Google's publicly available N-gram corpus, are used directly as real-valued attributes in this thesis. The frequencies are also scaled with a variety of scaling functions as described later in Section 4.4. The N-gram modeling fits into the philosophy of choosing statistical and unsupervised techniques. Furthermore, these frequencies are widely used in natural language processing although usually they are refined into more sophisticated features.

4.2 Newly proposed features

The features proposed in this thesis were selected to target more of the semantic characteristics of prominence due to the already existing strong syntactic basis. Although, it is not always obvious which aspects of the language a feature might reflect. Language-independence and features requiring minimal human intervention in construction were preferred. Following the recent trends of natural language processing, the experiments focus on graph-based and spatial language models.

The newly proposed features are based on a spatial word vector representation and a graph-based keyword weighting model named DegExt. These models and their extraction methodologies are described in the following subsections.

In addition, applicability of a simpler feature, the number of letters witihin words, were tested. Like other continuous-valued features not already limited to a small numeric range, lengths of words were scaled with different scaling functions described in Section 4.4. This feature was motivated by its simplicity and similar features in literature. Syllable-level prominence prediction often involved analysis of lexical and phonological features. Identities of syllables, vowels or phonemes were proposed [Sun02, JL09] along with utilizing a dictionary of lexical stress for stress patterns of syllables [RO96, Sun02, JL09]. Also, number of syllables, vowels and phonemes within words were considered [GA04].

4.2.1 Word vectors

Distributed word representations have been applied to a variety of natural language processing tasks [BDVJ03]. A distributed representation is motivated, for instance, by the curse of dimensionality problem in modeling of joint probabilities of word sequences. To illustrate this problem, consider modeling of joint probabilities for word sequences of length k within a vocabulary V of size $|V|$. In this case, the size of the feature space is $|V|^k$, which would count as much as 10^{50} degree of freedom if a context length of $k = 10$ and vocabulary size of 10^5 were chosen. Thus, many interesting sequences are not present in the training data and due to lack of information the model is unable to assign any proper probabilities for such sequences. A distributed representation of words provides a way of measuring word similarities, which makes it possible to assign a probability for an unseen sequence through the joint probabilities of the most similar known sequences.

To overcome the curse of dimensionality, Bengio et al. [BDVJ03] propose simultaneous computation of a continuous vector space representation of words along with the joint probabilities of word sequences. In continuous vector space, a word w is represented as a n-dimensional real-valued vector $w \in \mathbb{R}^n$. The joint probability function of the word sequences can be modeled with a probabilistic neural network. The network's input consists of vocabulary indices of the k context words, which are projected to k feature vectors in the first hidden layer. This is a linear layer as it consists of a $|V| \times n$ matrix of

weights from which only k feature vectors are active at a time. These vectors are then forwarded to another hidden layer with non-linearity to model the probabilities. The output is computed through softmax to normalize the probabilities, and it is interpreted as the conditional probability:

$$\hat{P}(w_k \mid w_{k-1}, \ldots, w_1),$$

where $(w_1, w_2, \ldots, w_k)$ is a word sequence.

This work has been followed by many others and it has been shown that the word vectors can be computed without modeling the complete neural network. Recently proposed use of a much simpler log-linear model to learn word vectors reduces the required computational complexity allowing use of larger data sets and learning higher dimensional vectors in reasonable time [MCCD13]. Continuous bag-of-words and continuous skip-gram models were proposed to be used with the log-linear model. The continuous bag-of-words is similar to the conditional probabilities considered with the neural networks. Here, the conditional probabilities of the current word given a context word are averaged over all the context words. The skip-gram model is a reversed version as conditional probabilities for the context words given the current word are learned. Despite of the weaker modeling of the language, the availability of larger data sets and higher dimensionality enable learning of high-quality word vectors that have state-of-the-art performance on measuring syntactic and semantic word similarities.

Very recent research shows that the syntactic and semantic relations between words can be modeled by simple linear operations of the normalized word vectors [MCCD13, MYZ13]. For example, computing $king - man + woman$ results in a vector that is the most similar with the vector of the word queen except the calculated words (king, man and woman). Thus, the vector $woman - man$ seems to model semantic transition between counterparts over sexes. Correspondingly, this method applies also to other semantic relations, and further to syntactic relations such as singular nouns versus plurals or adjectives versus comparative adjectives.

The similarity between two word vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ can be computed as cosine similarity: $cos(\mathbf{x}, \mathbf{y}) = \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\|\mathbf{x}\| \|\mathbf{y}\|}$. Normalizing the vectors to unit length further simplifies this to $\langle \mathbf{x}, \mathbf{y} \rangle$ — the inner product in the vector space. The

normalization projects the vectors to the surface of multi-dimensional unit ball. Therefore, Euclidean distance can be equivalently used to measure distance between words as well, because if vectors $\mathbf{x}$ and $\mathbf{y}$ are more similar than $\mathbf{x}$ and $\mathbf{z}$, more precisely $\langle \mathbf{x}, \mathbf{y} \rangle \geq \langle \mathbf{x}, \mathbf{z} \rangle$, then reverse inequality holds for the Euclidean distances: $\|\mathbf{x} - \mathbf{y}\| \leq \|\mathbf{x} - \mathbf{z}\|$ and its square form.

In this thesis, the word vectors are taken from the precomputed archive published on the Internet [1]. This archive is trained with part of the Google News Corpus and it contains three million vectors in 300-dimensional space. The words are learned with the skip-gram model over a context of length 10, and softmax is replaced with negative sampling as described by Mikolov et al. [MSC+13].

Word vectors were directly used as feature vectors (with or without normalization) as they seem to provide the intriguing syntactic and semantic information. Using the linear kernel of SVM yields inner products of the word vectors, which is equivalent to the previously proposed cosine similarity measure. This applies to the RBF kernels to some extent due to the explained correlation between Euclidean distances and cosine similarities of normalized vectors. Word vectors were looked up from the precomputed archive by applying case-sensitive search first and then again case-insensitively. Words missing corresponding vectors were assigned with vector 0, which is equally distant from all of the normalized vectors.

Cosine similarity of word vectors were used to construct two additional features. One models the similarities between the predicted word and the words surrounding it, while the other is a continuous-valued extension of the discrete word givenness status used in literature. That is, each predicted word is compared to the whole preceding context of a discourse and the maximum of the cosine similarity is chosen to represent the givenness status. In case of missing word vectors, the similarities are assigned to 0, which is in the middle of the scale $[-1, 1]$.

[1] http://code.google.com/p/word2vec/

4.2.2 Decomposed word-vectors

Decomposition techniques were applied to the word vectors to prevent sparsity of the data in a high-dimensional space. Sparsity becomes even a greater problem when the word vectors of several words are combined into one feature vector. Mikolov et al. [MSC+13] conclude that the parameters, for example the dimension, of the word vector representation are dependent of the targeted task. As it would have been too time consuming to generate and test multiple representations with varying dimensions in the scope of this thesis, independent component analysis were chosen to decompose the vectors into lower-dimensional spaces. Decomposition could reduce noise of the data, and it does not necessarily produce comparable results with decreasing the dimensionality of the whole representation in first place.

Independent component analysis is carried out by FastICA algorithm described by Hyvärinen and Oja [HO00]. Implementation of the Sklearn Python package was used. Functions logcosh, cube, and exp were tested to approximate the neg-entropy in the ICA. It turned out that their performances differ only a little and lacking any further reasoning the exp function was selected. Word vectors were normalized before decomposition, and the optimal number of components was searched with precision of 10 for both RBF and linear SVMs separately. The best performing number of components from the linear case was adopted to HM-SVM, which uses the same kernel.

4.2.3 DegExt — a keyword weighting model

DegExt is a language-independent keyword extraction model recently proposed by Litvak et al. [LLK12]. It does not require corpus based learning but instead is constructed solely from an input text document. Thus, it is thought to represent characteristics of text and at least happens to bear knowledge of words' saliency or documents' topics. The method is reported to outperform the two state-of-the-art keyword extraction models: TextRank and GenEx [LLK12]. They were tested against collections of benchmark summaries in English and Hebrew.

The method requires lemmatization and stopword removal as its preprocessing step, which somewhat challenges the language-independency. Still,

the rest is truly unsupervised. After the preprocessing of a document the remaining words form a sequence $D = (w_1, w_2, \ldots, w_n)$. Now, every remaining unique word is considered as a node $v \in V = \{w_i \mid 0 < i <= n\}$ in a graph $G = (V, E)$. The set of directed edges E represented order-relationships between words. if two words $u, v \in V$ appear as adjacent within any sentence of the document D, then there is an edge $(u, v) \in E$ between these nodes. The edges are further labeled with a set of IDs of sentences that provided the edge to the graph.

The keywords are weighted simply by the degrees, the number of adjacent edges, of the corresponding nodes. Keywords tend to occur more often in the text and presumably in different contexts of words generating more edges connected to that node. More complex degree centrality measures for this simple graph representation were reported to perform worse. In addition, multi-word key phrases are constructed up to length 3 by combining vertices connected by edges labeled with same sentence IDs. The key phrases are weighted with the average weight of its nodes.

Here, instead of extracting a set of keywords, the degree of a word available in the graph is used as a feature for prominence prediction. The value is set to zero for stop words, which were excluded from the graph. They were removed in first place based on the assumption that they do not represent the topics of a text.

The feature is scaled in various ways to perform better with SVM. The values are divided by the maximum value of a document or scaled with functions described in detail later in Section 4.4.

4.3 Algorithms

Support vector machines (SVMs) with different kernel functions and an extension with Hidden Markov Models (HMMs) were selected to be evaluated. Spatial and graph-based language models and continuous-valued features are believed by the author to outperform discrete attributes. Thus, classifiers that directly take advantage of continuous-valued features without any quantization methods are seen more promising. For instance, quantizing the word vectors — one of the newly proposed features — is rather impossible and such attempts

would decay the benefits of the whole representation. Support vector machines could provide generalizable models, because they measure the similarities between the continuous feature vectors.

Previous works evaluated classifiers from decision trees, neural networks, conditional random fields, support vector machines to maximum entropy models. Further, ensemble methods and hidden Markov models were used to boost the performance of decision trees. Although the SVMs have already been tested, the results reported by different authors, [JL09] and [NLX11], were inconsistent, and hence additional experiments are considered appropriate. This is motivated even more due to the fact that SVMs are easily used in inappropriate way, because of the importance of kernel selection, parameter tuning and data preprocessing. This is further discussed in Section 4.4.

4.3.1 Support vector machines

Support vector machines (SVMs) are a supervised statistical learning algorithm that separates a multi-dimensional feature space into two classes by hyperplanes. SVM is a maximum-margin classifier as it tries to maximize the separating margin between the data points of the two classes. By maximizing the margin, SVMs minimize the generalization error, which differs from neural networks that try to minimize the classification error instead. Non-linear hard-margin SVMs were proposed by Boser et al. [BGV92] in 1992 albeit much of the theoretical background is older. This model requires strictly separable data allowing no errors in the training data. The model was extended by Cortes and Vapnik [CV95] in 1995 to soft-margin SVM, which can manage also data with errors.

SVMs have become a popular classification technique. Much of its power arises from the non-linearity as the feature vectors $x \in \mathbb{R}^n$, $n \in \mathbb{N}$, are not necessarily linearly separable in the input space. In the non-linear SVM [CV95], the feature space $\mathbb{R}^n$ is non-linearly mapped to a high-dimensional Hilbert space $\mathcal{H}$. Then, linearly separating the transformed vectors in $\mathcal{H}$ allows classification of more complex tasks, because the target space $\mathcal{H}$ is higher in dimensions, or even infinite dimensional, and because of the non-linearity of the mapping function.

In practice, a non-linear mapping $\psi\colon \mathbb{R}^n \to \mathcal{H}$ is implicitly defined by defining a kernel function $k\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$. The relation between the kernel function k and the mapping ψ is defined by the equation: $k(\mathbf{x}, \mathbf{y}) = \langle \psi(\mathbf{x}), \psi(\mathbf{y}) \rangle_{\mathcal{H}} \ \forall\, \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$. Thus, a kernel function allows easy computation of the inner product in the high-dimensional space given the vectors of the original space. Note that not all functions $\mathbb{R}^n \to \mathcal{H}$ are valid kernels.

A variety of kernels have been proposed for SVMs. Given feature vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ the most common kernels are:

- **Linear kernel**: $k(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle$
- **Radial basis function (RBF)**: $k(\mathbf{x}, \mathbf{y}) = \exp -\gamma \|\mathbf{x} - \mathbf{y}\|^2$, where $\gamma > 0$
- Polynomial: $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + r)^d$, where $r \in \mathbb{R}$ and $d \in \mathbb{N}$, $d > 1$.
- Sigmoid (hyperbolic tangent): $k(\mathbf{x}, \mathbf{y}) = \tanh(\kappa \mathbf{x} \cdot \mathbf{y} + c)$, for some, but not every, $\kappa > 0$ and $c < 0$.

Solving a SVM is a quadratic programming problem. The separating hyperplane is linearly determined by the training vectors closest to the margin. These vectors are called support vectors, and usually they consist of a small portion of the vectors in training data.

More formally, given a training data set $\{(\mathbf{x_1}, y_1), \ldots, (\mathbf{x_m}, y_m)\}$ of feature vectors $\mathbf{x_i} \in \mathbb{R}^n$ and their labels $y_i \in \{-1, 1\}$, SVM algorithm outputs a function $f\colon \mathbb{R}^n \to \mathbb{R}$ of form:

$$f(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i \mathbf{x_i} k(\mathbf{x_i}, \mathbf{x}),$$

where $\alpha_i \geq 0 \ \forall\, i$. Thus, a vector $\mathbf{x_i}$ in the training set is a support vector if and only if $\alpha_i \neq 0$. The classifier is defined based on the decision function f as $g(x) = sign(f(\mathbf{x}))$.

Although SVMs are basically binary classifiers, several approaches for SVM-based multi-class classification have been proposed. Multi-class prediction problems are typically solved by decomposing the problem into multiple binary classification tasks. But it is also possible to learn multi-class classifiers directly with a single model [CS02].

Support vector machines have one common hyperparameter C, which determines the weight of how much each support vector affects the classification. The other possible hyperparameters are parameters of the kernel functions, for instance, γ for RBF and sigmoid kernels, d for polynomial kernel and r for sigmoid and polynomial kernels.

To achieve acceptable results with SVM, a practical guide [HCL03] suggests cross-validated grid search for the hyperparameters. The guide considers the RBF kernel as a reasonable first choice, because it transforms the input vectors into infinite dimensional space and it brings only one additional hyperparameter to estimate. For very high dimensional feature vectors, a linear kernel could be appropriate choice instead as it is computationally simpler. Testing an exponential range over values of the hyperparameters is suggested first, followed by another round of search with tighter range close to the best parameters found in the first search. It is recommended to use only portion of the training data to the exponential parameter search and the whole data to the fine tuning search.

The practical guide further highly recommends some sort of scaling of the features. The values should be scaled between ranges $[-1, 1]$ or $[0, 1]$ to avoid attributes with greater values dominating those with values in smaller numeric ranges. Too large values may also cause computational problems, for example with the polynomial kernel. For categorical features, a representation of binary parameters with values 0 and 1 is suggested.

In this thesis, the SVM implementation of the Python machine learning package **scikit-learn** is used [PVG+11].

4.3.2 Hidden Markov support vector machines

Support vector machine predicts only one label at a time. This would be appropriate, if the assumption of conditional independence between prominence labels, shown in Definition 1 (sec. 2.2), is really applicable. Such assumption, in spite of its widespread use, is questionable considering the nature of phrasal prominence. Therefore, a classifier that predicts sequences of labels were thought worth of research.

Hidden Markov support vector machine (HM-SVM) proposed by Altun et

al. [ATH03] combines hidden Markov models and support vector machines into a single optimization problem. In this model, instead of predicting individual class labels, output consists of finite-length sequences of labels. Sequential dependencies between labels are modeled via the Markov chain dependency structure for which Viterbi decoding gives an efficient dynamic programming formulation. The other side of the model is the linear kernel-based maximum margin classification technique derived from SVMs. In HM-SVM, support vectors are constructed from sequences of input vectors and corresponding labels. The authors of the technique note that the resulting model might be extremely sparse, because only a small portion of the negative samples end up as support vectors.

The author of this thesis is in the impression that this classifier is not earlier used for phrasal prominence prediction. However, the same line of research have seen conditional random fields and a combination of HMM and maximum entropy models from which the conditional random fields were experimented for text-based prominence prediction [GA04, NLX11]. And even earlier, probabilities from a decision tree were given to a hidden Markov model [RO96].

The implementation called SVM-Struct is used in the experiments [TJH$^+$05]. Sequences are delimited by all punctuation marks including quotes and colons. The order of dependencies transitions in HMM was set to 3 being the maximal value, and correspondingly, the order of dependencies in emissions was set to 1. The threshold error value for terminating the iterative maximization process is set to 0.001, which seemed to result with infinitely continuing process for some of the weakly performing models. HM-SVM introduces parameter C similar to other support vector machines but its interpretation is different, because in HM-SVM it trades-off effect of sequences, not singular cases.

4.4 Preprocessing and scaling

Attributes need to be scaled to achieve accurate models with support vector machines. Kernel functions, such as RBF, polynomial and linear kernels, tend to rely on some sort of a distance function like inner product or Euclidean distance. Those measures are greatly sensitive to the varying numeric ranges

of the individual attributes. That is, an attribute with a greater numeric range easily dominates the variation in the concerned measures, and therefore dominates the whole model. This is problematic especially in tasks that combine various types of features, for example, if classification is based on two attributes: one binary attribute converted to values 0 or 1, and one attribute that bears the number of letters within a word. Then, obviously the difference between words of lengths 3 and 10 is more affecting than change in the binary attribute. Furthermore, large values might increase the computational complexity of the kernel functions compared to small constrained numeric ranges.

To avoid those stumbling blocks, Hsu et al. [HCL03] recommend linear scaling of the attributes to ranges $[0, 1]$ or $[-1, 1]$. However, not every attribute is limited with a global maximum value, which would allow a simple linear transformation by dividing with the maximum. Further, if the distribution of an attribute is remarkably uneven, that could disadvantageously affect the distances measured by the kernel functions. Consider word lengths for instance, where the maximum length could easily be very high. But if it turns out that only the differences between the shortest words has any effect, those values populate a small portion of the rescaled range and thus the distances between interesting values are shorter than is expected.

To overcome the described problems, a set of non-linear transformations were chosen to be experimented in addition to the simple linear scaling. These transformations allow rescaling to a specific range without use of varying maximums. For example, inverse of word lengths emphasizes the variation among shorter words compared to the longer words, which might be reasonable. At least, the transformation is stable across all of the training and test data without any changing maximums.

The rescaled range was chosen to be either $[0, 1]$ or $[-1, 1]$ based on the feature in question. That is, if the original values contain negative values, the negative part of the range is included. It is assumed by the author that the differences in those scales do not significantly affect performance.

Given an attribute $x \geq 0$, the proposed value scaling functions are:

- none: $x \mapsto x$

- linear: $x \mapsto \frac{x - \min x}{\max x - \min x} \in [0, 1]$
- sentence relative: $x \mapsto \frac{x}{m} \in]0, 1]$, where m is the sentence-wide maximum value.
- exp: $x \mapsto e^{-x} \in]0, 1]$
- inverse: $x \mapsto \frac{1}{x+1} \in]0, 1]$
- inverse square root: $x \mapsto \frac{1}{\sqrt{x+1}} \in]0, 1]$
- logarithm: $x \mapsto \log x$

Efficiency of two preprocessing techniques, namely standardization and normalization, are evaluated in addition. Normalization refers to scaling the length of an input vector to one, and in standardization the distribution of each separate attribute is transformed to zero mean and unit variance. These are applied to the already scaled values of the input vectors. These are commonly used preprocessing steps in data mining.

5 Experiments on prominence prediction

The experimental part of the thesis is described in this section. In practice, the models described in the previous Section 4 are trained and tested with the data from the Boston University Radio News Corpus, which is represented in Section 5.1. The following Section 5.2 continues by discussing the procedures for evaluation of the models, and the efforts made to ensure valid and generalizable results. Tuning of classifier parameters and selection of features and their configurations are described step by step in Section 5.3, and finally the achieved results are shown in Section 5.4.

5.1 Dataset

A dataset consisting of text transcribed with sentence level information of prominence is required. Full text documents are necessary since prominence has discourse-wide effects. The most restricting criterion in choosing a corpus

is the selection of proper representation of prominence and availability of data with such transcriptions.

Prominence is not a very well understood area of research, and multiple representations have been proposed. Probably the most used representation in corpus-based prominence research is the Tones and Break Indices framework but even that does not directly provide labels for phrasal prominence. Instead, it offers a symbolic representation of pitch movements and breaks in speech data. These pitch movements can in turn be utilized to detect whether a word is given prominence or not by analyzing their location within a word. A pitch movement that is located on a stressed syllable of a word is called a pitch accent, which indicates that the word is spoken with prominence. The ToBI is used in this work due to its widespread use in the research community and the easy interpretation of the discrete symbols.

The ToBI framework consists of four separate information tiers: orthographic transcriptions, tones, break indices and miscellaneous non-speech events such as disfluencies, breathiness and laughter. The tonal tier is a symbolic representation of the intonation contour. It marks three types of pitch events: boundary tones located near intonational phrase boundaries; pitch accent events associated with accented syllables; and two additional labels to support investigation of peak alignment and phrasal pitch range. The tonal tier has two primary tones: high tone (H) denoting local maxima in pitch contour and low tone (L) denoting local minima. The rest of the symbols are combinations of the high and low tones. Tones marked with diacritic (*) are aligned onto stressed syllables and hence are pitch accents. The categories of pitch accents in the ToBI framework consist of: two symbols ('L*' and 'H+!H*') indicating low or falling accent tones; two ('H*' and 'L+H*') indicating high and rise to a peak; and a scooped accent ('L*+H') marking a local minima on the accented syllable followed by a peak.

Other important criteria for choosing a dataset are the size of the set, the quality of the transcriptions and the genre of the documents. Firstly, more data usually provides better results but reliably labeling tones of the ToBI framework is a time-consuming process, which restrains sizes. Secondly, the genre of the documents could have strong correlation to the quality and

	Total	Speakers					
		f2b	**m1b**	**f1a**	**m2b**	**f3a**	**m3b**
Stories	92	39	14	14	8	13	4
Sentences	1644	675	252	248	198	158	113
Words	30600	12855	5029	4386	3431	2805	2094
Prominent (%)	55.4	55.5	54.3	56.6	57.8	55.2	52.0

Table 2: The basic statistics from the BURNC corpus.

consistency of the labels. For example, audio books and news are spoken by professionals and thus could have a more correctly formed layer of prosody than spontaneous conversations. Radio announcers and audio book performers strive to be informative and to sound natural. Moreover, different genres may emphasize different uses or manifestations of prominence.

The experiments in this thesis are conducted with the Boston University Radio News Corpus (BURNC). It consists of radio news spoken in American English by professional radio announcers and a part of it is annotated with ToBI framework by human labelers. This corpus is used, because it contains speech of professional speakers and it has been widely used in earlier works making comparisons much easier.

The annotated part of the BURNC corpus consists of 92 stories with 1644 sentences and 30600 words. There are only 18 questions and a single exclamation. Thus, the data contains significantly only statements. The news stories are spoken by seven speakers of which six have ToBI labels annotated. The recordings were captured in a radio studio and laboratory environments. Table 2 shows the number of stories, sentences, words and proportion of prominent words. These values are given in total and for each separate speaker. The speakers are divisible into females (f) and males (m). The set of tonal symbols present in the tonal tier of the BURNC corpus and their distribution over the whole data are shown in Figure 1. Symbols *, *? and X*? denote partly annotated or unsure pitch accentuation.

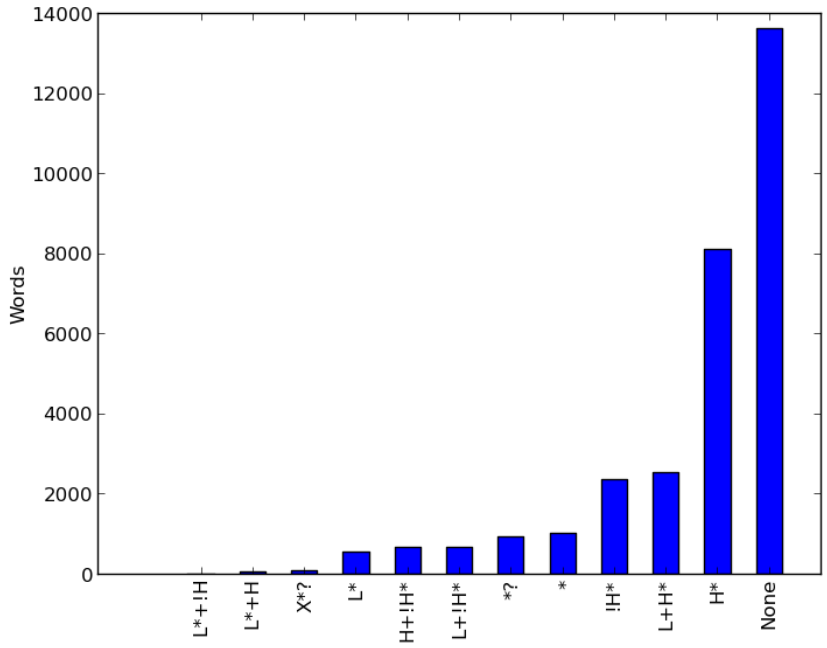

Figure 1: Summary of the pitch accent labels and their distribution over the data set. The proportion of unaccented words is referred to as none.

Binary labels for existence versus absence of phrasal prominence are generated by assigning a word as prominent if it carries any pitch accent. In terms of ToBI labels, this means that each word labeled with a symbol containing a diacritic is assumed prominent. Also, equally to the research of Rangarajan et al. [RNB06], the partly annotated and unsure pitch accent symbols are included — considered prominent. Labeling the data in the described manner provides 55.4 percents of prominent words, which introduces the lower bound for the success rate of prominence prediction as it is easily reached by constantly predicting each word prominent. Ratios of prominent words are shown for each speaker separately and in total in Table 2.

5.2 Evaluation and validation methodology

Performances of experimented models are evaluated with accuracy and F-measure. Accuracy is used to compare the results throughout the research

from parameter tuning to feature selection. F-measures are provided to give more information and comparability.

Following good machine learning practices, the dataset is first split into two subsets: an experiment set for experimenting with models and a held-out validation set for validating the evaluation of the final models. Each of the sets contains complete news stories — they are not split, because predicting a story is one indivisible task.

Three possible procedures for separating the held-out set from the experiment set were considered:

1. Randomly drawing stories from each speaker with approximately the same ratio.

2. Randomly drawing a story out of the experiment set until the requested ratio is reached.

3. Choosing a single speaker to act as the held-out validation set.

The first procedure was chosen yielding a held-out set that contains data from all of the speakers. Speaker-independency is guaranteed by evaluating each portion of the validation set with a model trained from all other speakers of the experiment and validation set, which enables more training data for the validation step compared to model experimentations. High variation of the news story lengths and the number of samples from different speakers makes it quite hard to split the data somewhat randomly but still preserving completeness of the stories and the requested split ratio.

The third option would provide even stronger speaker-independent validation as the speaker of the validation data is completely unknown when tuning the models with the experiment set. The downside of this procedure is, however, that the choice of the validation speaker could affect the results greatly. There might exist some other factors, for instance varying transcription quality, in the data that make it unbalanced. The second option falls somewhere between the other two options. Randomly choosing the validation stories from the whole data might leave some speakers out of the validation set, and those speakers are most probably the ones with the least data.

Speaker	**Stories**		**Words** (%)	
	experiment set	held-out set	experiment set	held-out set
f3a	10	3	2260 (9.4)	545 (8.4)
m1b	10	4	3796 (15.7)	1233 (19.1)
f2b	30	9	10248 (42.5)	2607 (40.3)
f1a	10	4	3441 (14.3)	945 (14.6)
m3b	3	1	1657 (6.9)	437 (6.8)
m2b	7	1	2727 (11.3)	704 (10.9)
Total	70	22	24129	6471
	92		30600	

Table 3: The distribution of the dataset over speakers and the experiment versus held-out validation sets.

The distribution of stories and words over speakers resulted from the performed separation are shown in Table 3. As the table shows, the realized ratio for the split slightly differs from the planned 80 : 20 ratio, which is an obvious consequence of randomly splitting indivisible stories of varying lengths.

Models are evaluated using leave-one-out cross-validation. That is, the data of each speaker is in turn used as a test set, while training with the rest of the speakers experiment set. Thus, models are evaluated in six phases — one for each of the six speakers. This ensures that no data from the same speaker exists in the training and the test sets. The evaluation is said to be speaker-independent, which makes the results more generalizable and complicates the problem on the other hand. Furthermore, such six-fold cross-validation enables use of larger amount of training data as the test sets can be smaller.

Accuracies, F-measures and confusion matrices are calculated so that the cross-validation process is first done completely and the measures are taken from the whole data. This is reasonable, because the number of samples from different speakers and thus the sizes of the test sets vary drastically. Just averaging over the results of separate cross-validation rounds would provide unbalanced measures.

The final results were validated with the held-out set. In addition to the

speaker-independent six-fold cross validation, results were also validated with two other data configurations. Firstly, results were evaluated following the instructions by Rangarajan et al. [RNB06] and Chen et al. [CHC04], who provided the previously best performing approaches. Secondly, the effect of relaxing the speaker-independence restriction between training and test sets is investigated by training models with the whole experiment set and testing it with the whole held-out set.

5.3 Parameter tuning and feature selection

The aim is to search for accurate models combining classifiers with different parameters and data representations with varying configurations. Degree of freedom for the whole configuration of models would be too much to be searched in reasonable amount of time. Thus, different aspects are experimented in separate steps that hopefully cover a reasonable part of the search space. Features are first optimized by varying context sizes, value scaling functions and preprocessing techniques, namely normalization and standardization. The feature configurations are searched with support vector machine using both RBF and linear kernels. The results from the linear case are assumed directly applicable to the sequential classifier HM-SVM, which uses a linear kernel as well. Then, parameters of the classifiers are tuned with some of the best performing features separately and in combinations.

The following list summarizes the process of parameter tuning and feature selection, which is then elaborated step by step in the rest of the section:

1. Selection of initial classifier parameters
2. Search for optimal configurations for each feature
3. Classifier parameter tuning with some individual features
4. Search for efficient multi-featured models
5. Another round of classifier parameter tuning with multi-featured models

To bootstrap evaluation of support vector machines, some initial kernel parameters are required. As there is no prior knowledge about suitable kernel

nor parameterization for this specific problem, the initial values were chosen according to the earlier general experience of the classifiers. Two kernels were chosen: the RBF kernel and the linear kernel. The RBF function has been suggested as a reliable first choice [HCL03], and the linear kernel was chosen due to its computational simplicity and comparability to the HM-SVM classifier.

SVMs and HM-SVM use general parameter C to trade-off between minimal misclassifications and simplicity of the decision surface. Higher values mean smoother margin due to lesser amount of support vectors, whereas the other end implies more complex model and probably less misclassification errors. Additionally, the RBF kernel defines the γ parameter, which is a constant modifier to kernel distances. The lower the gamma value is, the higher are the values of the kernel, and hence effect of each support vector reaches farther. Lacking better reasoning for selection of the initial parameters, the values were chosen to be the default parameters suggested by authors of the Scikit-learn implementation [PVG+11]. That is, the initial values are: $C = 1$ and $\gamma = \frac{1}{n}$, where $n \in \mathbb{N}$ denotes the dimension of the input vectors.

Some data and accurately performing features are needed in order to tune the classifier parameters for the specific problem. Consequently, feature configurations were optimized with the classifiers using their initial parameters. The features were inspected carefully at this point, because performance of both features and classifier parameterizations depends on: the dimension of the data vectors, preprocessing techniques, scaling of feature values, applicability of the feature to model phrasal prominence, error rates of the data and so forth. For instance, inappropriate selection of preprocessing or scaling techniques might completely ruin the usability of a feature. Moreover, context of multiple words was used to build input vectors and the context size is the major cause for changes in dimensionality. The context size introduces a trade-off between amount of useful data against increasing sparsity and level of noise. Wider context provides more information but unnecessarily increasing the context might also lower the performance and increase the time complexity of training for sure.

Features were tested with different scaling functions and varying sizes of

context. They were tested with and without standardization and normalization of feature vectors. The number of preceding and succeeding words was optimized by grid search over the set: $\{0, 5, 10, 15, 20, 25, 30, 35\}$. Then, another grid search was applied around the best performing context lengths from the first search. The second search searched all possible modifiers to the result of the first round within range $[-5, +5]$.

Classifier parameters were evaluated with several features to avoid ungeneralizable effects of an individual feature. Part-of-speech tags, word vectors, decomposed word vectors, word lengths and DegExt performed better than the simple discrete word givenness status and were therefore considered to form a comprehensive test basis with their best performing parameterizations.

To tune the kernel parameters, Hsu et al. [HCL03] suggest use of grid search over the kernel parameters in their practical guide for SVMs. Following their instructions, the search was done for exponentially growing parameter values. For the non-linear SVM, the parameters C and γ were firstly both drawn from set: $\{2^{-12}, 2^{-9}, 2^{-6}, \ldots, 2^{1}, \ldots, 2^{6}, 2^{9}\}$. Then, they were further tuned by finer-grained search near the best performing values from the coarser search. Given the best parameter p from the coarse search, the second search was done across the set of $\{p*2^{-2}, p*2^{-1}, p*2^{-0.5}, p*2^{-0.25}, p*2^{0.25}, \ldots, p*2^{2}\}$. In the case of linear SVM, values between 0 and 1 were tested with resolution of 0.05 along with greater values $2, 8, 64, 128$ and 256. In the sequential HM-SVM the parameter C is used as a parameter for sequences of vectors that makes the situation completely different from the original SVM. The parameter is first searched across a very coarse set $\{10^{0}, 10^{1}, \ldots, 10^{4}\}$, and then across finer-grained set $\{10^{x-0.5}, 10^{x-0.4}, \ldots, 10^{x}, \ldots, 10^{x+0.4}, 10^{x+0.5}\}$ near the best resulting value 10^{x} from the first search.

Changes of the parameter C turned out to have very little effect on the performance of SVMs with both linear and RBF kernels. Only values near zero and much greater than one significantly decreased performance. In linear case, the differences between the default value and the best performing values were not even visible with precision of permille except word vectors which differed 0.4 percents. Correspondingly, the non-linear classifier followed the same trends. As C parameter has no direct interpretation and due to the

minimal differences, the default value is still used when testing the combined models.

However, the parameter C influenced the performance of the HM-SVM classifier in much greater extent. Small values, including the default $C = 1$, invariably resulted with the worst performances. Use of the default value totally ruined performance of features based on decomposed word vectors and word lengths, while for part-of-speech tags the change was significantly the least, being less than one percentage. Therefore, the default value must be replaced to successfully evaluate combinations of the features. The resulted best values for C varied dramatically among the features ranging from $10^{1.5}$ to $10^{3.4}$, where the highest values were resulted by the very same features that performed badly with the default C. Still the accuracies did not change much in the second search. The first search resulted with values of either 10^2 or 10^3, and further observations of the results from the second search showed that the values near $10^{2.5}$ performed generally rather well. Thus, the combined features were tested with that compromising value $10^{2.5}$, which at least is far better choice than the default.

For RBF kernel, the gamma parameter affected performance in greater extent compared to C. Values above one quickly ruined the performance, and classification with different features resulted with significantly varying values for γ. To shed more light on the reasons for the experimented variation, the resulted values were transformed into a form of $\gamma = \Gamma\frac{1}{n}$, where $n \in \mathbb{N}$ denotes the number of dimensions. In this formulation, the Γ mainly varied around 1 and 2 except part-of-speech tags, which resulted with a multiplier $\Gamma \approx 30$. Therefore, it seems that most of the variation is explained by the number of dimensions.

The deviation of gamma does not affect the prediction performance based solely on part-of-speech tags but efficiently merging the tags into multi-feature classifiers with one common parameter value encouraged further inspection of the phenomenon. The γ constant of the RBF function acts as a modifier to distances between feature vectors, and hence, modifying the vector distances of the data would allow efficient use of different, more widely suitable gamma. Considering the binary vector representation of part-of-speech tags shows

that the squared Euclidean distances of such vectors are within range $[0, 2^2]$, because norm of the vectors is one. On the other hand, the squared distance between two n-dimensional standardized word vectors is $2n$ in average, which is defined in Lemma 2 and briefly proven below. Attributes of standardized vectors are assumed to be independent, which is underpinned by empirical observations.

Lemma 2. Expectation value of squared Euclidean distance between standardized vectors*:*
Let $x, y \in \mathbb{R}^n$, $n \in \mathbb{N}$, be n-dimensional standardized vectors. Assuming that the attributes of standardized vectors are independent, the expectation value for squared Euclidean distance is $\|x - y\| = 2n$, and thus it is directly proportional to the number of dimensions.

Proof. Let X, Y be independent random variables such that $E(X) = E(Y) = 0$ and $\operatorname{Var} X = \operatorname{Var} Y = 1$. Definition of variance gives: $\operatorname{Var} X = E(X^2) - E(X)^2 \implies E(X^2) = \operatorname{Var} X + E(X)^2$. Therefore it holds that:

$$\begin{aligned} E((X-Y)^2) &= E(X^2 && -2XY + y^2) && (5)\\ &= E(X^2) && -E(2XY) + E(Y^2) && (6)\\ &= \operatorname{Var} X + E(X)^2 && -2E(X)E(Y) + \operatorname{Var} Y + E(Y)^2 && (7)\\ &= 1 + 0 && -2 * 0 + 1 + 0 && (8)\\ &= 2 && && (9) \end{aligned}$$

Now, let $x, y \in \mathbb{R}^n$, $n \in \mathbb{N}$, be n-dimensional standardized vectors. Then, assuming that the independence holds, the expectation value for the squared Euclidean distance $\|x - y\|^2$ becomes:

$$\begin{aligned} E(\|x-y\|^2) &= E(\sum_{i=1}^{n}(x_i - y_i)^2) && (10)\\ &= \sum_{i=1}^{n} E((x_i - y_i)^2) && (11)\\ &= \sum_{i=1}^{n} 2 && (12)\\ &= 2n && (13) \end{aligned}$$

□ □

This motivated scaling the binary vectors to get distances similar to those of standardized data, like rest of the best performing features with non-linear prediction. As only a constant number of dimensions differ from zero by a constant, the binary vectors were multiplied with $\sqrt{n}$, where $n \in \mathbb{N}$ denotes the number of dimensions. Then, the distance between two distinct vectors $x, y \in \{0, \sqrt{n}\}^n$, $n \in \mathbb{N}$ becomes $\|x - y\|^2 = 2\sqrt{n}^2 = 2n$ and the image of the RBF function becomes: $\{e^{-2}, e^0\}$. This was also tested in practice. Parameter tuning with the rescaled part-of-speech feature provided values of Γ similar to the other features.

The described phenomenon arose yet another question about whether the standardized features performed better due to the selection of the initial gamma value which was relative to the number of dimensions. Normalized word vectors were tested without standardization and with $\gamma = 1$, and it turned out to perform as accurately as standardized word vectors with the parameter $\gamma = \frac{1}{n}$. Other features were accordingly tested with normalization and unit gamma but they seemed to perform worse. Nevertheless, it seems that dimension-dependent gamma is appropriate for standardized features, whereas normalized features perform well with $\gamma = 1$ or another nearby constant.

The next step in the experiments was evaluation of multi-featured models. All non-binary attributes were standardized for non-linear SVM due to observed better performances. Therefore, initial value for the parameter gamma was chosen to be relative to the number of dimensions. The very initial parameter value $\gamma = \frac{1}{n}$ was used again in absence of any obvious way of finer tuning. To fit into this setting, feature vectors constructed from part-of-speech tags were scaled in the way explained above. In the case of linear SVM, no standardization, and thus no binary vector rescaling, were used, because it would break the inner product. The initial value of parameter C was unchanged for the SVMs due to absence of good reasoning for use of any other value.

Multi-featured models were evaluated in two steps, because searching across all subsets of the features would be too time-consuming. Firstly, a subset of the best performing features was selected to test all of their possible

combinations. This set was limited to those features that performed better than the word givenness status. Such set was small enough to be searched in reasonable amount of time. Secondly, the other, weaker features were added one by one to the best performing combination from the first step.

The classifier parameters were considered one more time for the best performing multi-featured models. The very same procedures were followed for each classifier as in the earlier tuning steps. It turned out that the default gamma for RBF and parameter $C = 10^{2.5}$ for HM-SVM still resulted with the best performances. Therefore, no other models were tuned and those values were used in the final validated evaluation.

The final optimized feature configurations are summarized for the non-linear RBF-based SVM in Table 5, and for both linear models, namely Linear-SVM and HM-SVM, in Table 4. These configurations are used in all of the models presented in the results. Independent component analysis (ICA), which is tested only to preprocess word vectors, is introduced and its parameterization is discussed in Section 4.2.2. The features are later on referred by their abbreviations specified in the tables.

Normalization was effective for word vectors overall, DegExt weights for the linear kernel and binary vectors (e.g. part-of-speech tags) for the non-linear case. Standardization was more widely useful as it was successfully applied to every continuous-valued attribute when modeling with the RBF kernel. For linear kernel and binary attributes, standardization made results dramatically worse. Non-linear scaling were observed to outperform linear transformation, at least when evaluating word lengths, DegExt and sentence positions. The most successful scaling functions were e^{-x} and $\frac{1}{x}$. Context sizes varied across features and classifiers. Word vectors, being the highest dimensional feature, did not benefit from bigger context than three words possibly because rapidly increasing sparsity. Bigger contexts were effective for the part-of-speech tags and the biggest for word lengths and DegExt weights, which were lowest in per word dimensions.

Feature	**Abbr.**	**Context**	**Preprocessing**
Word vectors	WV	$[-1, +1]$	normalized
Decomposed word vectors	DWV	$[-1, +1]$	ICA: $\mathbb{R}^{300} \rightarrow \mathbb{R}^{300}$
Part-of-speech tags	POS	$[-9, +4]$	-
Word lengths	WLen	$[-29, +24]$	$x \mapsto \frac{1}{x}$
DegExt weights	DegExt	$[-22, +31]$	$x \mapsto e^{-x}$, normalized
Relative sentence position	SPRel	$[-0, +0]$	-
Discrete word givenness	WGiven	$[-0, +0]$	-

Table 4: Best performing parameterizations for features when predicting with the linear classifiers (Linear-SVM and HM-SVM).

Feature	**Abbr.**	**Context**	**Preprocessing**
Word vectors	WV	$[-1, +1]$	normalized
Decomposed word vectors	DWV	$[-1, +1]$	ICA: $\mathbb{R}^{300} \rightarrow \mathbb{R}^{90}$
Part-of-speech tags	POS	$[-4, +8]$	$x * \sqrt{24}$
Word lengths	WLen	$[-28, +34]$	$x \mapsto \frac{1}{x}$
DegExt weights	DegExt	$[-31, +34]$	$x \mapsto e^{-x}$
Relative sentence position	SPRel	$[-0, +0]$	-
Discrete word givenness	WGiven	$[-0, +0]$	-

Table 5: Best performing parameterizations for features when predicting with the non-linear SVM (RBF-SVM).

5.4 Results

This section focuses on how different features and classifiers performed in the prominence prediction task. The results are elaborated starting from the individual features, then proceeding to multi-featured models, and finally concluding with the results of the best performing model along with comparison to previous results.

Performances of the single features (Table 6) were dominated by word vectors, which alone achieved an accuracy of 84.16%. This is remarkably better than the earlier state-of-the-art feature, namely part-of-speech tags that achieved an accuracy of 82.58%. A permutation-based procedure was used to test whether the difference between the accuracies is statistically significant [Coh95]. Null hypothesis is that accuracies of the two models come from the same distribution. Labels of the tested models were randomly permuted 10000 times to obtain a sample of the test statistic under the null hypothesis. A two-tailed p-value was obtained as the proportion of the absolute difference between the accuracies that were equal of greater than the observed difference. The null hypothesis was rejected for part-of-speech tags and word vectors ($p \leq 0.0001$), so the observed difference is statistically significant.

Results of the best single features are shown in Table 6 for each classifier. Decomposing word vectors by ICA affected performance minimally, while for RBF-based prediction it reduced dramatically the dimensionality. Word vectors utilized a smaller context of words (cf. Tables 4-5) and still provided more accurate predictions than part-of-speech tags. Reducing dimensionality by decomposition did not allow use of bigger contexts. Keyword weighing with DegExt performed a little better than word lengths with RBF and HM-SVM.

Search for efficient combinations of features resulted with different optimal sets for different algorithms. Thus, those results are separated by the classifier into three tables (7-9). The displayed results were selected to consist of: the best performing single feature, the best combination of the strong features with and without part-of-speech tags, and lastly the best multi-feature model including the weak features (see Section 5.3 for explanation of the strong and weak features). Note that features are selected based on prediction results

Feature	**Accuracy**		
	HM-SVM	Linear-SVM	RBF-SVM
WV	82.10	81.78	84.16
DWV	81.97	82.15	84.14
POS	80.88	80.54	82.58
WLen	77.47	77.10	78.80
DegExt	80.06	75.91	79.01

Table 6: Best performing single features.

with the experiment set but the displayed values are validated accuracies. The models are ordered from weakest to strongest, although this does not fully generalize to the results from the held-out set. The differences between the models adjacent in the tables are statistically significant (permutation test; $p \leq 0.001$), unless otherwise specified.

Use of multiple features improved classification performance to a lesser extent. The improvements were smallest with the non-linear SVM, where word vectors achieved accuracy of 84.16% compared to the best performing multi-feature model resulting with 84.42% accuracy. However, the only statistically insignificant difference was caused by inclusion of POS tags. The linear counterparts showed a bit more variation possibly because separating a space with a linear hyperplane is harder than separating higher-dimensional space as a product of a non-linear mapping. Adding more features increases dimensionality and thus separability of the problem space. Differences with linear SVM were statistically significant except for addition of word givenness status. HM-SVM was the most unstable classifier as the models benefitted significantly from neither POS tags nor sentence position.

The non-linear SVM with RBF kernel (Table 9) achieved the best performance also with multi-featured models. Correspondingly, the lowest accuracies resulted from the SVM with linear kernel as shown in Table 7. The performance of the linear classifier improved only slightly when the hidden Markov extension was applied (Table 8).

The best model evaluated in this thesis achieved better performance than any of the earlier models as far as the author knows. The proposed model

Features	**Accuracy**	**F-measure**
DWV	82.15	84.14
DWV, WLen, DegExt	82.60	84.70
DWV, WLen, DegExt, POS	82.88	84.88
DWV, WLen, DegExt, POS, WGiven	83.08	85.04

Table 7: Best performing models for linear SVM.

Features	**Accuracy**	**F-measure**
WV	82.10	83.98
WV, WLen, DegExt	83.26	85.10
DWV, Wlen, DegExt, POS	82.66	84.78
DWV, Wlen, DegExt, POS, SPRel	82.77	84.89

Table 8: Best performing models for HM-SVM.

Features	**Accuracy**	**F-measure**
WV	84.16	86.03
WV, DWV, WLen	84.36	86.26
WV, DWV, POS	84.44	86.32
WV, DWV, POS, WGiven	84.42	86.32

Table 9: Best performing models for RBF-based SVM.

Actual	**Predicted**	
	Non-prominent	Prominent
Non-prominent	8556	2446
Prominent	971	12156

Table 10: Confusion matrix for the best performing model computed from the experiment set.

slightly surpasses the maximum entropy model of Rangarajan et al. [RNB06] by increasing the prediction accuracy from 85.22% to 85.30%. As displayed on the last row of the Table 9, this was accomplished by applying RBF-based SVM classifier to word vectors with and without decomposition accompanied by previously proposed POS tags and word givenness status. Confusion matrix for the best model computed from the experiment set is shown in Table 10. The number of words incorrectly classified as prominent is over a half more than those incorrectly non-prominent.

In addition to the primary speaker-independent six-fold cross-validation with the held-out set, some of the best performing models were further evaluated with two other arrangements of training and testing. Firstly, for comparability, models were evaluated equivalently to the procedure taken by Chen et al. [CHC04] and Rangarajan et al. [RNB06]. That is, the speakers f3a and m3b with the least number of words were left out of the cross-validation, and f2b, the speaker with the most data, was used only for training. Moreover, the whole data set was used in evaluation instead of the held-out set. Such evaluation with only a subset of the corpus produced the highest accuracies. Table 11 shows the best model from two of previously the most successful studies, which are compared to the best model from the experiments of this thesis. The model based on POS tags is also included for better comparison between the classifiers. Word vectors are the best new feature and naturally compared to POS tags as the previously best feature.

Secondly, for experimental purposes, models were trained with the whole experiment set and tested with the held-out set, which means the largest possible amount of training samples and breaking speaker-independence. Such evaluation showed the greatest variation, which was expected as the

Algorithm	**Features**	**Accuracy (%)**
MLP [CHC04]	POS	82.67
MaxEnt [RNB06]	POS	85.22
RBF-SVM	POS	83.54
RBF-SVM	WV	84.59
RBF-SVM	WV, DWV, POS, WGiven	85.30

Table 11: Accuracies of the best performing models evaluated equivalently and compared to the previous results.

evaluation contains only one round of training and testing. It was observed that the more accurate a model was the less varied its performance between different arrangements of evaluation. In spite of larger set of training data and the simplified task, this experimental setup produced lower accuracies compared to the evaluation adopted from literature. Six-fold cross-validation with speaker-independence resulted the lowest accuracies of all the attempted arrangements but the differences were minor overall.

6 Discussion

Non-linear support vector machines with RBF kernel were observed to achieve the best results among the experimented classifiers overall. When comparing models that utilize part-of-speech tags, SVM with RBF kernel is slightly more accurate than what Chen et al. [CHC04] reported for their multi-layer perceptron. Correspondingly, the maximum entropy model is reported by Rangarajan et al. [RNB06] to perform better than the SVMs but their classifier is intended more for binary data than kernel based approaches are. The perceptron outperformed both linear SVM and HM-SVM on predictions based on POS tags, whereas the newly proposed classifiers performed better with other combinations of the newly proposed features. HM-SVM classifier mostly performed better than the original linear SVM, and the differences were more apparent for weaker features such as word lengths and DegExt. Seemingly modeling conditional dependencies of word sequences is beneficial but HM-SVM struggles to take advantage of it or needs more sequences than

those available in the BURNC corpus.

Although linear models are weaker than the RBF-based support vector machine, they are also tremendously faster to compute. As a practical example, average durations for one cross-validation round with a 900-dimensional input space consisting of word vectors are 6 and 800 seconds for the linear and non-linear SVMs respectively. Therefore, the amount of training data for the linear models could be increased much more than would be practical with the RBF kernel. It seems that natural language processing can benefit from relatively simple language models (e.g. vector representations, word lengths and DegExt), and thus linear SVM with a much larger data set could still be of interest.

Parameter tuning of the evaluated classifiers turned out not to be as influential as expected. Applying grid-search to optimize the parameters for RBF-based SVM seems exaggeratedly complex in light of the observed results. It is sufficient to choose the γ parameter in such way that it restricts the squared Euclidean distances of the RBF function into a small constant sized range, $[0, 2]$ for instance. In other words, value of $\frac{1}{n}$ is an eligible choice for n-dimensional input vectors with norms relative to n, and correspondingly $\gamma = 1$ suits vectors with a small constant norm well. The dimension dependent option is therefore eligible with standardization, which was observed to be an efficient preprocessing step for any continuous-valued attribute. Furthermore, binary vectors can be easily fit to such model by modifying their norms to n with a simple multiplication.

HM-SVM appeared to be the hardest classifier to tune as its C parameter varied greatly, especially for single-featured modeling. Fortunately, this variation became steadier when more features were combined together making selection of a suitable value easier. For the original support vector machines, eligible values of C varied only between zero and one, and affected accuracy quite minimally. The C parameter is harder to interpret and tune for HM-SVM, because that model compares sequences instead of singular words.

In experiments evaluation was applied in such way that training and test sets did not contain data from common speakers. This arrangement is reasonable when prominence prediction aims at supporting speech recogni-

tion for instance. However, if the targeted application is speech synthesis, speaker-independence complicates the task unnecessarily. Speech recognition is obviously targeted to recognize speech of many different speakers while for speech synthesis the desired result is one voice with its own way of giving prominence. The results showed that accuracies did not improve much when the speaker-independence constraint was relaxed. The data still consisted of discourses spoken by multiple speakers, who probably speak with remarkably unique styles.

The BURNC data corpus appears to be highly prominent as 55.4 percent of the words bear pitch accent. Furthermore, predicting prominence based only on the number of letters within words performed surprisingly well. What if prominence in spoken news stories is caused by some special factors? It would be logical that news announcers tend to speak with especially good articulation and hence make words more easily prominent. At least, based on author's subjective experience of conversations about prominence with non-experts while working on the thesis, people apparently easily over-stress their speech when consciously trying to put prominence on words. And why the number of letters within words — the simplest feature the author even could think of — is achieving so good results? Do news announcers speak with rhythm, or are news texts produced in such way that they show rhythm or other specific structuring.

Observed performance with part-of-speech tags shows that most of prominence in the data has strong syntactic basis, which presumably holds also for many other text genres. News script writers probably try to avoid ambiguities as much as it is possible by linguistic means at text level. Structures of sentences could be more restricted due to less freedom of choice in constructing news scripts. For instance, grammatical tenses, more constrained use of adjectives, and continuingly referring to different sources of information sound like reasonable restrictions. Such issues may reduce the need for phrasal prominence in its full diversity. Perhaps more subtle semantic features are overwhelmed by noise and syntactic reasons in too conscious or professionally precise speech.

An important question though is which genres would show more distinctly

non-syntactically caused instances of prominence. News announcers attempt to speak more accurately than pleasantly. For the author it sounds like announcers have something very important to say with every word they utter. More appropriate source for prominence modeling could be audio books, where texts are more freely written and texts are primarily intended to be read, not spoken. Audio books are also produced by professional speakers, who try to empathize to the story more than news announcers do. Further advantages are that audio books have been produced for many languages and they contain a lot of data spoken by a single speaker. Real applications of future natural speech synthesis could even benefit from learning different aspects of prominence from different styles of text, and then somehow combine them together.

Observed dominance of syntactic causes for prominence may also be induced by classifying only on binary level. Syntactic phenomena are probably more visible in prominence placement while semantic aspects are manifested by finer-grained levels of prominence. The Tones and Break Indices framework utilized in the majority of prominence modeling gives no obvious way to derive finer levels. Thus, based on the author's beliefs, currently the most important direction of future work is developing a better intermediate representation of prominence addressing the problem of automated prominence detection.

7 Conclusions

The target of this thesis was to improve performance of text-based phrasal prominence modeling. In practice, new classifiers and new models from different fields of natural language processing were explored. Applicability of spatial or graph-based language models was personally considered promising and has not been tested before. This lead to selection of such features as word vectors, a high-dimensional word representation, and DegExt, a keyword weighting method. Support vector machine was chosen due to its widespread suitability to supervised classification tasks with high-dimensional continuous-valued input. Linear inner product and non-linear RBF kernels were tested, and additionally hidden Markov support vector machine was evaluated to

investigate benefits of sequential classification. SVMs were earlier tested only with polynomial kernel and for syllable-level prominence.

Non-linear SVM with RBF kernel was substantially the best of the tested classifiers. HM-SVM mostly performed slightly better than the original SVM with linear kernel advocating use of sequential modeling. However, HM-SVM was found to be the hardest classifier to tune, and it seems not to generalize very well as is evident from the variation of the validated results. Linear SVMs can be trained orders of magnitudes faster than RBF-based, and could therefore be worth of further testing with larger sets of data.

Evaluation of the proposed models was successful in two major ways: the best performing model appeared to slightly surpass the previously best prediction accuracy and a new state-of-the-art feature was invented. The best performing model resulted with accuracy of 85.30% when evaluated in the comparable way. It is only slightly higher than the accuracy of 85.22% reported by Rangarajan et al. [RNB06]. Their maximum entropy model is based on hand-corrected part-of-speech tags, whereas the outperformer is a SVM with radial-basis kernel function and input additionally consisting of: word vectors, decomposed word vectors, part-of-speech tags, and word givenness status. As far as the author knows, these are the state-of-the-art models for word-level binary prediction of phrasal prominence. Though, the achieved improvement is quite insignificant, and the new model used more features.

Nevertheless, a more important result is that the features based on word vectors performed surprisingly well — even significantly better than part-of-speech tags, which has been inevitably the most successful feature so far. Using normalized word vectors directly as input for SVM with the RBF kernel performed nearly as well as the best combination of the features. Moreover, when evaluated equivalently to what Rangarajan et al. [RNB06] did, word vectors achieved almost as high accuracies as their model that utilized hand-corrected POS tags from a context of seven words. So, word vectors utilized a smaller context of only three words and still performed better or almost as well. The word vectors also benefit from decomposition by independent component analysis, which reduced the number of dimensions to 30% of the

original without significantly affecting the performance.

Spatial representations of words have several advantages compared to part-of-speech tags. Word vectors are a product of completely unsupervised process compared to the automatically generated but human-derived word classes. Furthermore, word vectors are able to describe more sensitive phenomena than those wide word classes. On the other hand, this also means greater level of sparseness. However, this is compensated by the smoothing of kernel-based classifiers as they take into account the distances of the data points. Distances of word vectors are more meaningful compared to categorical binary vectors whose distances are constant across the data.

The accomplished experiments also provided more evidence for applicability of vector representations to different problems in natural language processing. Mikolov et al. [MCCD13] demonstrated the success of word vectors by answering questions in the form of: "What is the word that is similar to x in the same sense as y is similar to z?" They tested simple linear calculations of words in vector space against a hand-collected set of word pairs with syntactic or semantic linguistic relationships. Their notion of word vectors' suitability to natural language processing is supported in this thesis by the experiments showing good performance for a real world task with evaluation on real world data. It is known that phrasal prominence highly depends on syntactic aspects and word vectors performed well as an alternative source of information to part-of-speech tags. Word vectors and part-of-speech tags separately performed rather as well, and combining them into a single model did not improve accuracy much further. This might indicate that word vectors are able to provide similar syntactic information than POS tags do.

Word vectors utilized less input words than part-of-speech tags. Training with larger data set could allow use of wider contexts and thus improve the accuracy of the feature even more. Anyhow, results with word vectors are very promising, because the features tested here are very simple. What could be possible with more sophisticated models? It is known that different relationships between words in vector space can be encoded by subtracting the related word vectors from each other. Such subtractions result with a

set of vectors pointing into different directions with different lengths, and leading to words differently related to the original word. However, the tested kernel functions are spherical in nature meaning that they are not able to fully utilize the available information. It could be beneficial to consider some set of relationship-encoding vectors to unevenly weigh the effect of vectors' dimensions to the measure. Or considering a simpler kernel function that would make equally distant vectors more similar if their subtraction is more unevenly distributed over the dimensions.

Prominence prediction benefitted also from much simpler features. Lengths of words and the keyword weighing with DegExt were observed to perform relatively well, especially considering their simplicity. Together with the success of POS tags this questions applicability of the used corpus, or more generally, use of news data to learn phrasal prominence. In the future, it would be reasonable to test other text genres, use a larger data set, and representing prominence more informatively. This in turn requires improving automatic detection of prominence from speech as well as introduction of a more sophisticated intermediate representation of prominence that is designed directly for the required purposes instead of pitch movements. As discussed earlier, audio books were personally considered to be a promising source of data as they are spoken by professionals like news stories but with more emphasis on naturalness and interpretation of more freely written texts.

References

[AN05] Ananthakrishnan, S. and Narayanan, S. S., An automatic prosody recognizer using a coupled multi-stream acoustic model and a syntactic-prosodic language model. *Proceedings of the 30th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1-5. IEEE, 2005, pages 269–272. Philadelphia, USA.

[AN07] Ananthakrishnan, S. and Narayanan, S., Improved speech recognition using acoustic and lexical correlates of pitch accent in a n-best rescoring framework. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 4, 2007, pages 873–876. Honolulu, USA.

[AN08a] Ananthakrishnan, S. and Narayanan, S., Automatic prosodic event detection using acoustic, lexical, and syntactic evidence. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16,1(2008), pages 216–228.

[AN08b] Ananthakrishnan, S. and Narayanan, S., Fine-grained pitch accent and boundary tone labeling with parametric f0 features. *Proceedings of the 33th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1-12, 2008, pages 4545–4548. Las Vegas, USA.

[ATH03] Altun, Y., Tsochantaridis, I. and Hofmann, T., Hidden markov support vector machines. *Proceedings of the 20th International Conference on Machine Learning ICML*, volume 3, 2003, pages 3–10. Washington DC, USA.

[BDVJ03] Bengio, Y., Ducharme, R., Vincent, P. and Jauvin, C., A neural probabilistic language model. *The Journal of Machine Learning Research*, 3,(3)(2003), pages 1137–1155.

[BGV92] Boser, B. E., Guyon, I. M. and Vapnik, V. N., A training algorithm for optimal margin classifiers. *Proceedings of the Fifth Annual*

Workshop on Computational Learning Theory, COLT '92. ACM, 1992, pages 144–152. New York, USA.

[Bol72] Bolinger, D., Accent is predictable (if you're a mind-reader). *Language*, 48,3(1972), pages 633–644.

[Bol78] Bolinger, D., Intonation across languages. *Universals of human language*, 2 (1972), pages 471–524.

[CDv97] Cutler, A., Dahan, D. and vanDonselaar, W., Prosody in the comprehension of spoken language: A literature review. *Language and Speech*, 40,2(1997), pages 141–201.

[Cha08] Chan, O., *Prosodic features for a maximum entropy language model*. School of Electrical, Electronic and Computer Engineering, The University of Western Australia, 2008.

[CHC04] Chen, K., Hasegawa-Johnson, M. and Cohen, A., An automatic prosody labeling system using ANN-based syntactic-prosodic model and GMM-based acoustic-prosodic model. *Proceedings of the 29th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1. IEEE, May 2004, pages I–509–12 vol.1. Montreal, Canada.

[Coh95] Cohen, P., *Empirical Methods for Artificial Intelligence*. Bradford Books. MIT Press, 1995.

[CRR99] Conkie, A., Riccardi, G. and Rose, R. C., Prosody recognition from speech utterances using acoustic and linguistic based models of prosodic events. *Proceedings of the 6th European Conference on Speech Communication and Technology (EUROSPEECH)*, volume 523, 1999, page 526. Budapest, Hungary.

[CS02] Crammer, K. and Singer, Y., On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research*, 2,(3)(2002), pages 265–292.

[CV95] Cortes, C. and Vapnik, V., Support-vector networks. *Machine Learning*, 20,(3)(1995), pages 273–297.

[CW92] Chen, F. and Withgott, M., The use of emphasis to automatically summarize a spoken discourse. *Proceedings of the 17th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, 1992, pages 229–232. San Francisco, USA.

[FH82] Fujisaki, H. and Hirose, K., Modelling the dynamic characteristics of voice fundamental frequency with application to analysis and synthesis of intonation. *Proceedings of the 13th International Congress of Linguists (ICL)*, 1982, pages 57–70. Tokyo, Japan.

[FR10] Fernandez, R. and Ramabhadran, B., Discriminative training and unsupervised adaptation for labeling prosodic events with limited training data. *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2010.

[GA04] Gregory, M. L. and Altun, Y., Using conditional random fields to predict pitch accents in conversational speech. *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (ACL) 2004*, ACL '04, 2004. Barcelona, Spain.

[GEVC12] Gonzalez-Ferreras, C., Escudero-Mancebo, D., Vivaracho-Pascual, C. and Cardenoso-Payo, V., Improving automatic classification of prosodic events by pairwise coupling. *IEEE Transactions on Audio Speech and Language Processing*, 20,7(2012), pages 2045–2058.

[GNF98] Grabe, E., Nolan, F. and Farrar, K. J., IVie-a comparative transcription system for intonational variation in english. *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP)*, 1998. Sydney, Australia.

[Gro83] Grosjean, F., How long is the sentence? Prediction and prosody in the on-line processing of language. *Linguistics*, 21,3(1983), pages 501–530.

[HCL03] Hsu, C.-W., Chang, C.-C. and Lin, C.-J., *A practical guide to support vector classification*. 2003.

[Hir93] Hirschberg, J., Pitch accent in context - predicting intonational prominence from text. *Artificial Intelligence*, 63,1-2(1993), pages 305–340.

[HIV94] Hirst, D., Ide, N. and Véronis, J., Coding fundamental frequency patterns for multi-lingual synthesis with INTSINT in the MULTEXT project. *The Second ESCA/IEEE Workshop on Speech Synthesis*, 1994. New Paltz, USA.

[HO00] Hyvarinen, A. and Oja, E., Independent component analysis: algorithms and applications. *Neural Networks*, 13,4(2000), pages 411–430.

[Jak67] Jakobson, R., *Preliminaries to speech analysis: the distinctive features and their correlates*. Cambridge, Mass.: M.I.T. Press, 1967.

[JL09] Jeon, J. H. and Liu, Y., Automatic prosodic events detection using syllable-based acoustic and syntactic features. *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1-8. IEEE, 2009, pages 4565–4568. Taipei, Taiwan.

[Lad96] Ladd, D. R., *Intonational Phonology*. Cambridge studies in linguistics. Cambridge University Press, 1996.

[Leh72] Lehiste, I., The timing of utterances and linguistic boundaries. *The Journal of the Acoustical Society of America*, 51,6B(1972), pages 2018–2024.

[LLK12] Litvak, M., Last, M. and Kandel, A., DegExt: a language-independent keyphrase extractor. *Journal of Ambient Intelligence and Humanized Computing*, 4,3(2012), pages 377–387.

[Mag96] Maghbouleh, A., A logistic regression model for detecting prominences. *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, volume 4, 1996, pages 2443–2445. Philadelphia, USA.

[MCCD13] Mikolov, T., Chen, K., Corrado, G. and Dean, J., Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781.* 2013.

[MMC13] Mehrabani, M., Mishra, T. and Conkie, A., Unsupervised prominence prediction for speech synthesis. *Power*, 2,1.6(2013), pages 1–3.

[MSC+13] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. and Dean, J., Distributed representations of words and phrases and their compositionality. *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2013, pages 3111–3119. Lake Tahoe, USA.

[MYZ13] Mikolov, T., Yih, W. and Zweig, G., Linguistic regularities in continuous space word representations. *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, 2013, pages 746–751. Atlanta, USA.

[NBK+07] Nenkova, A., Brenier, J. M., Kothari, A., Calhoun, S., Whitton, L., Beaver, D. and Jurafsky, D., To memorize or to predict: Prominence labeling in conversational speech. *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, 2007, pages 9–16. Rochester, USA.

[NLX11] Ni, C., Liu, W. and Xu, B., Automatic prosodic events detection by using syllable-based acoustic, lexical and syntactic features. *Proceedings of the 12th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2011, pages 2017–2020. Florence, Italy.

[OR97] Ostendorf, M. and Ross, K., A multi-level model for recognition of intonation labels. In *Computing Prosody*, Springer US, 1997, pages 291–308.

[PH90] Pierrehumbert, J. and Hirschberg, J., The meaning of intonational contours in the interpretation of discourse. *Proceedings of Interdisciplinary Workshop on Intentions and Plans in Communication and Discourse*, System Development Foundation Benchmark Series, 1990, pages 271–311. Monterey, Canada 1987.

[PVG+11] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E., Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12,(11)(2011), pages 2825–2830.

[RH07] Rosenberg, A. and Hirschberg, J., Detecting pitch accent using pitch-corrected energy-based predictors. *Proceedings of the 8th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, August 2007, pages 2777–2780. Antwerp, Belgium.

[RNB06] Rangarajan, V., Narayanan, S. and Bangalore, S., Acoustic-syntactic maximum entropy model for automatic prosody labeling. *Proceedings of IEEE Spoken Language Technology Workshop (SLT)*, 2006, pages 74–77. Palm Beach, Aruba.

[RO96] Ross, K. and Ostendorf, M., Prediction of abstract prosodic labels for speech synthesis. *Computer Speech and Language*, 10,3(1996), pages 155–185.

[Roa02] Roach, P., A little encyclopaedia of phonetics. *University of Reading, UK.* 2002.

[SBP+92] Silverman, K. E., Beckman, M. E., Pitrelli, J. F., Ostendorf, M., Wightman, C. W., Price, P., Pierrehumbert, J. B. and Hirschberg, J., TOBI: a standard for labeling english prosody. *Proceedings of the 2nd International Conference on Spoken Language Processing (ICSLP)*, volume 2, 1992, pages 867–870. Banff, Canada.

[Sun02] Sun, X., Pitch accent prediction using ensemble machine learning. *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP)*, 2002, pages 953–956. Denver, USA.

[Tay98] Taylor, P., The tilt intonation model. *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP)*, 1998, pages 1383–1386. Sydney, Australia.

[TJH+05] Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y. and Singer, Y., Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6,9(2005).

[VSA13] Vainio, M., Suni, A. and Aalto, D., Continuous wavelet transform for analysis of speech prosody. *TRASP 2013-Tools and Resources for the Analysys of Speech Prosody, An Interspeech 2013 satellite event, Laboratoire Parole et Language, Proceedings.* Aix-en-Provence, France 2013.

[WO94] Wightman, C. W. and Ostendorf, M., Automatic labeling of prosodic patterns. *IEEE Transactions on Speech and Audio Processing*, 2,4(1994), pages 469–481.