DOĞAN DALVA

Ph.D. Thesis

2018

Co-Training using Prosodic, Lexical and Morphological Information for Automatic Sentence Segmentation of Turkish Spoken Language

DOĞAN DALVA

IŞIK UNIVERSITY
2018

Co-Training using Prosodic, Lexical and Morphological Information for Automatic Sentence Segmentation of Turkish Spoken Language

DOĞAN DALVA
B.S., Electronics Engineering, IŞIK UNIVERSITY, 2009
M.S., Electronics Engineering, IŞIK UNIVERSITY, 2012

Submitted to the Graduate School of Science and Engineering
in partial fulfillment of the requirements for the degree of
Doctorate of Science
in
Electronics Engineering

IŞIK UNIVERSITY
2018

IŞIK UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

Co-Training using Prosodic, Lexical and Morphological Information for Automatic Sentence Segmentation of Turkish Spoken Language

DOĞAN DALVA

APPROVED BY:

Assoc. Prof. Dr. Ümit Güz (Thesis Supervisor)	Işık University
Assoc. Prof. Dr. Hakan Gürkan (Thesis Co-Supervisor)	Bursa Technical University
Prof. Dr. Yorgo Istefanopulos	Işık University
Prof. Dr. Ercan Solak	Işık University
Prof. Dr. Murat Saraçlar	Boğaziçi University
Assoc. Prof. Dr. Cenk Demiroğlu	Özyeğin University

APPROVAL DATE: 15/01/2018

Co-Training using Prosodic, Lexical and Morphological Information for Automatic Sentence Segmentation of Turkish Spoken Language

Abstract

Sentence segmentation of speech aims detecting sentence boundaries in a stream of words output by the speech recognizer. Sentence segmentation is a preliminary step toward speech understanding. It is of particular importance for speech related applications, as most of the further processing steps; such as parsing, machine translation and information extraction, assume the presence of sentence boundaries.

Typically, statistical methods require a huge amount of manually labeled data, which is time and labor consuming process to prepare. In this work, novel multi-view semi-supervised learning strategies for the solution of sentence segmentation problem are proposed.

The aim of this work is to find effective semi-supervised machine learning strategies when only a small set of sentence boundary labeled data is available. This work proposes three-view co-training and committee-based strategies incorporating with agreement, disagreement and self-combined strategies using lexical, morphological and prosodic information, and investigates performance of the proposed learning strategies against baseline, self-training and co-training. The experimental results show that the proposed learning strategies highly improve the sentence segmentation problem, since data sets can be represented by three redundantly sufficient and disjoint feature sets.

Keywords: Boosting, Co-Training, Forced Alignment, Lexical Feature Extraction, Machine Learning, Morphology, Multi-View Semi-Supervised Learning, Prosody, Prosodic Feature Extraction, Sentence Segmentation, Self-Training

Bürüsel, Sözcüksel ve Biçimbilgisel Bilgiyi Kullanan Eş-Eğitim ile Türkçe Konuşma Dilinin Otomatik Cümle Bölütlemesi

Özet

Cümle bölütleme işlevi, standart Otomatik Konuşma Tanıma (OKT) sistemlerinin çıkışından elde edilen işlenmemiş kelime dizisi biçimindeki veriyi cümlelere ayırarak zenginleştirmeyi amaçlayan bir işlemdir. Cümle bölütleme; çözümleme, makine çevrimi, bilgi çıkarımı gibi cümle bölütlemenin yapıldığının varsayıldığı konuşma işlemenin daha ileri uygulamaları için bir ön adım olarak gerçekleştirilmektedir.

Cümle bölütlemede kullanılan standart yöntemler, model eğitimi aşamasında oldukça fazla etiketlenmiş veriye ihtiyaç duyar. El ile yapılan veri etiketleme işlemi; emek, dikkat ve zaman isteyen bir işlemdir. Bu çalışmada çok bakışlı yarı öğreticili yöntemler geliştirerek, daha az el ile etiketlenmiş veri ile standart yöntemlere göre daha yüksek başarımın sağlanması hedeflenmektedir.

Bu çalışmada çok bakışlı yarı öğreticili yöntemler geliştirerek, daha az el ile etiketlenmiş veri ile standart yöntemlere göre daha yüksek başarımın sağlanması hedeflenmektedir. Bu çalışmada sözcüksel, biçimbilgisel ve prozodik özellikleri kullanan, uzlaşma (agreement), uzlaşamama (disagreement) ve self-combined yöntemleri ile beraber çalışan yeni üç bakışlı eş eğitim (co-training) ve kurul tabanlı (committee-based) yöntemler geliştirildi. Yeni yöntemlerin performansları, iki bakışlı eş eğitim yöntemleri, kendi kendini eğitme (self-training) yöntemi ve standart yöntemler ile kıyaslandı. Deneysel sonuçlar, veri kümeleri yeterli ve ayrık özellik grupları kullanılarak ifade edilebildiği için, önerilen yöntemlerin cümle bölütleme başarımını oldukça arttırdığı göstermektedir.

Anahtar kelimeler: Biçimbilgisel Bilgi (Morfoloji), Eş Eğitim (Co-Training), Çok Bakışlı Yarı Öğreticili Öğrenme, Cümle bölütleme, Makine Öğrenmesi, Prozodi, Prozodik Özellik Çıkarımı, Sözlüksel (Lexical) Özellik Çıkarımı, Zorlanmış Hizalama

Acknowledgements

This material is based upon work supported by the Scientific and Technological Research Council of Turkey (TUBITAK) (Project Number: 107E182 and Project Number: 111E228) and Isik University Scientific Research Project Fund (Project Number: 09A301 and Project Number: 14A201). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and thesis supervisors, and do not necessarily reflect the views of the funding agencies. The author thanks his thesis supervisors Ümit Güz and Hakan Gürkan, and also Murat Saraçlar, Cenk Demiroğlu, Fatih Özaydın, Ercan Solak, Yorgo Istefanopulos, Erinç Dikici, Izel D. Revidi, Tonguç Yavuz, and Şafak Arslan for many helpful discussions.

To my family and friends…

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AdaBoost	Adaptive Boosting
ADJ	**A**djective
ASR	**A**utomatic **S**peech **R**ecognizer
A3sg	Third Person Singular Agreement
BN	**B**roadcast **N**ews
CTM	**C**onversational **T**ime **M**arks
CRF	**C**onditional **R**andom **F**ield
DB	**D**erivational **B**oundaries
EM	**E**xpectation **M**aximization
Fut	Future Tense
FutPart	Future Participle
HELM	**H**idden **E**vent **L**anguage **M**odels
HMM	**H**idden **M**arkov **M**odels
HTK	**H**idden **M**arkov Model **T**oolkit
L	**L**abeled Set
l	**L**exical View
LVCSR	**L**arge-**V**ocabulary **C**ontinuous **S**peech **R**ecognition
LEX	Lexical Feature Set
LM	**L**anguage **M**odels
m	**M**orphological View
MAP	**M**aximum **A** **P**osteriori
MaxEnt	Maximum Entropy
MFCC	**M**el-**F**requency **C**epstral **C**oefficients
MORP	Morphological Feature Set

NIST	**N**ational **I**nstitute of **S**tandards **T**echnology
n	**N**on-sentence Boundary Hypothesis
Nom	Nominal Case
NLP	**N**atural **L**anguage **P**rocessing
IG	**I**nflectional **G**roup
p	**P**rosodic View
P1sg	First Person Singular Agreement
PAC	**P**robably **A**pproximately **C**orrect
Past	Past Tense
Pron	No Possesive Agreement
PROS	Prosodic Feature Set
POS	**P**art **O**f **S**peech
Pos	Positive Polarity
s	**S**entence Boundary Hypothesis
SB	**S**entence **B**oundary
SOV	**S**ubject + **O**bject + **V**erb
STM	**S**egment **T**ime **M**arks
SU	**S**entence **U**nits
Strategy 1	Three-View Co-Training Strategy 1
Strategy 2	Three-View Co-Training Strategy 2
Strategy 3	Three-View Co-Training Strategy 3
Strategy 4	Three-View Co-Training Strategy 4
Strategy 5	Three-View Co-Training Strategy 5
Strategy 6	Three-View Co-Training Strategy 6
Strategy 7	Three-View Co-Training Strategy 7
Strategy 8	Committee-Based Learning Strategy 8
Strategy 9	Committee-Based Learning Strategy 9
U	**U**nlabeled Set
VOA	**V**oice **o**f **A**merica
WER	**W**ord **E**rror **R**ate

Chapter 1

Introduction

Automatic Speech Recognition (ASR) systems provide transcriptions of spoken words without punctuation signs. ASR systems are widely used in several human-machine interactions such as using smart phones with little commands, call-center decision trees, smart home systems. On the other hand, when ASR output of a long speech is considered, gathering information from such output is almost impossible, even for humans, without segmenting this raw output into sentences. In addition, manually sentence segmenting is time and labor consuming for a huge data. Following example illustrates typical output of an ASR system.

■

“automatic speech recognition asr systems provide transcriptions of spoken words without punctuation signs asr systems are widely used in several human machine interaction such as using smart phones with little commands call center decision trees smart home systems on the other hand when asr output of a long speech is considered asr systems provide raw text transcripts of recognized words without any punctuation signs gathering information from such output is next to impossible even for humans without segmenting this raw output into sentences segmenting raw asr output into sentences manually is time and labor consuming for a huge data” . . .

■

In the literature it has been shown that sentence boundaries are very crucial for legibility of speech transcripts [1]. Moreover, missing sentence boundaries cause meaning ambiguity for some utterances. For instance the following utterance "no jobs are running" has two completely different possible interpretations such that "No jobs are running" and "No. Jobs are running." [2]. This example shows that using only lexical features may not be sufficient for sentence segmentation. It has been shown that prosodic information in speech (acoustic model), and morphological information in text (language model) [3] provide complementary information to lexical information for segmentation of speech into sentences [3, 4, 5].

In previous studies, supervised methods have been employed for sentence segmentation. Training binary classifiers with supervised methods require huge amounts of manually labeled training data, which is time and labor consuming to prepare. Supervised model adaptation methods proposed in [5], divide training set into two subsets which are called labeled (in-domain) and unlabeled (out-of-domain) data, where the size of the former is significantly smaller than the latter. Moreover in [6], the effects of single pass co-training algorithms on sentence segmentation problem have been analyzed using prosodic and lexical information. In [7], the performance of multi-view semi-supervised models, which exploit unlabeled data using prosodic and lexical features were compared to several semi-supervised learning methods, such as self-training and co-training on sentence segmentation task. In that work, two-view co-training approach was employed on the International Computer Science Institute (ICSI), Meeting Recorder Dialog Act (MRDA) Corpus [8, 9], and it has been shown that iterating two-view co-training algorithm using the agreement, disagreement and self-combined strategies outperformes single pass two-view co-training and self-training algorithms even at the first iteration.

This work and [10] propose a better sentence segmentation system, which extend two-view semi-supervised co-training approach into three-view co-training strategies (Strategies 1 to 7) incorporating agreement, disagreement, and self-combined strategies. In this work, it has been shown that three-view co-training strategies, which are considered as either combined or extended versions of two-view co-training strategies, are very appropriate for the sentence segmentation problem, since data sets can be represented by three redundantly sufficient and disjoint feature sets such as prosodic, morphological and lexical information. In addition, this work proposes committee-based learning strategies over different feature sets in Committee-Based Learning Strategy 8 (Strategy 8), and over committee-based learning strategies (Strategies 2, 3, 5, 6, 7, 8) in Committee-Based Learning Strategy 9 (Strategy 9).

The organization of this thesis is follows as: Chapter 2 presents previous related studies, Chapter 3 presents data collection and annotation methods, Chapter 4 presents prosodic and lexical feature extraction methods and contents of prosodic, lexical and morphological information, Chapter 5 presents semi-supervised methods and experimental setup, Chapter 6 presents experimental results based on different features and strategies, and finally Chapter 7 presents the conclusion.

Chapter 2

Literature Survey

Before presenting the details of the semi-supervised learning methods employed in this work, first the related work for sentence segmentation and the literature review on the semi-supervised learning methods are presented.

2.1 Sentence Segmentation

Automatic segmentation of speech into sentence units (SU) is a very important task and essential for many natural language processing (NLP) methods. In the literature, one of the most typical approaches of sentence segmentation is classification of words into two classes such that whether the current word is followed by a sentence boundary (s) or a non-sentence boundary (n). Therefore, this task can be considered as a binary classification problem by finding probability $argmax_T P(T|F)$ which is the most likely boundary tag sequence T given by the features F [11].

Raw or unformatted word transcript of an utterance processed by an ASR is as follows [12]:

”hi bill it’s tracy at around three thirty PM just got an apartment for one thousand three thirty one thousand four hundred a month my number is five five five eight eight eight eight extension is three thirty bye”.

Corresponding formatted transcript version processed by human annotators of the transcript above should be in the following form:

"Hi Bill. It's Tracy. At around three thirty PM just got an apartment for one thousand three thirty one thousand four hundred a month. My number is five five five eight eight eight eight. Extension is three thirty. Bye."

Several different methods such as discriminative, generative or hybrid approaches are employed on sentence segmentation problem for different spoken languages. In [13], hidden event language model (HELM), which is one of the well-known generative approaches, has been proposed. In that work, probability of the current word followed by a hidden event was modeled as $P(Y_t|W_t, Y_{t-1}, W_{t-1})$, where Y_t represents the boundary that follows the current word W_t. In that work they proposed a new language model which classifies disfluencies such as filled pauses (typically "uh" or "um"), repetitions (contiguous repeated words) and deletions. In [3], HELM extended to factored HELM (fHELM) for sentence segmentation of Turkish spoken language. In that work, probability of the current word followed by a sentence boundary was modeled as
$P(Y_t|W_t, M_t, Y_{t-1}, W_{t-1}, M_{t-1})$, where M_i represents morphological information based on pseudo morphological features, which include last three letters of the current word. Last three letters may include inflectional and derivational suffixes, which provide important cues about location of sentence boundaries. Similar pseudo morphological features were also used in [14] for Czech spoken language. In [4, 15], prosodic and lexical information were used in hybrid models based on decision trees to improve the performance of either sentence or topic segmentation. In that work fundamental prosodic features (duration, pitch, pause and other features) were described, and feature selection mechanisms such that leave-one-out and beam search methods were used, to obtain effective prosodic feature subsets.

In discriminative classification approaches, conditional random fields (CRFs) which directly estimate posterior boundary label probabilities [16], boosting which trains a final strong classifier using weighted sum of weak classifier decisions [5], and hybrid approaches were developed with emphasis on boosting and maximum entropy (MaxEnt) in [17]. In addition, a maximum-likelihood approach for automatically constructing maximum entropy models were proposed and efficient implementation methods of in several natural language processing (NLP) problems were described in [17]. In [4], performance of the following methods: posterior probability interpolation, integrated HMM and HMM posteriors as decision tree features, have been compared. In another work [18], performances of several classification algorithms such as HELMs, MaxEnt and boosting (using BoosTexter) were compared for sentence segmentation problem in English and Mandarin spoken languages using speaker change information, lexical features, and prosodic features. In that work it has been shown that boosting-only was the most effective method compared to HELM-only and MaxEnt-only methods. In binary combination of those methods, hybrid usage of HELM and Boosting outperformed other binary combinations when lexical and prosodic features are used. For Spanish and Portuguese spoken languages, [19, 20] used language models, part of speech (POS) tags, pause duration and speaker change information to classify full stops, comma and question marks using maximum entropy (ME) models.

Adaptation methods were first proposed in [21]. In that work, several adaptation methods such as data concatenation, logistic regression, and boosting were used for sentence segmentation of conversational telephone speech (CTS) of Switchboard corpus. Moreover in [22], hybrid combination of prosodic features and feature selection for multilingual sentence segmentation and logistic regression were used to analyze the effect of model adaptation for dialog act tagging.

Distinct modeling approaches such as Hidden Markov Model (HMM), MaxEnt, and CRFs were used in [23, 24] for disfluency detection and sentence segmentation problems. In [14], language models were constructed by N-grams and prosodic models were used to classify each inter-word boundary into several classes such as sentence boundary and short pauses for Czech spoken language. In [25], sentence utterance detection was implemented using Multi-level language analysis for Chinese spoken language. Moreover in [26], performances of trained models by using several subsets of prosodic features were compared for English, Mandarin and Arabic spoken languages.

One of the main differences between English and Turkish spoken languages is the productive and agglutinative morphology of Turkish [27]. Therefore number of possible word forms derived from a root word is much larger compared to English. This increases the complexity when we want to develop a statistical language model for sentence segmentation. To alleviate this problem, [28] used morphological information in sentence segmentation of Turkish spoken language. In that work it has been shown that combination of lexical and morphological information outperformed lexical-only models. Novel methods for discriminative, generative and hybrid sequence classification were presented on sentence segmentation of Turkish in [3]. In that work it has been shown that discriminative classification approaches; CRF and boosting, provided the best results for sentence segmentation of Turkish, since CRF provided better results with prosodic and lexical features only and morphological features provided additional significant information for boosting. Moreover, application of multi-view semi-supervised learning algorithm by using prosodic and lexical information were investigated in [6] and [7].

Additionally for Turkish broadcast news (BN) data, prosodic features were extracted using Purdue Prosodic Feature Extraction Tool based on Praat, developed by [29], then these features were used on sentence segmentation in [30, 31, 32]. Finally, in [33] lexical, prosodic and morphological features were extracted in order to use them in sentence segmentation of Turkish BN data.

2.2 Semi-Supervised Learning

In conventional machine learning approaches only manually labeled data is used to train the classifiers or statistical models. Labeling data manually is an inefficient process. On the other hand it is easy to gather unlabeled data but there are few unsupervised methods to use them. Semi-supervised learning addresses this problem by using huge amount of unlabeled data, together with a small set of manually labeled data, to build better classifiers. Automatic labeling based on semi-supervised learning provides high accuracy with reduced human effort and it is of great interest both in theory and in practice [34].

Semi-supervised methods, which require small amount of initial manually labeled data, are preferred in order to increase efficiency of most real-life applications. In semi-supervised methods, the training set is divided into two parts called in-domain labeled data and out-of-domain unlabeled data, to build better classifiers. Typically the size of the former is relatively much more smaller than the latter [34]. During the process, the labels (classes) of unlabeled portions of the data are estimated with confidence scores, then, a certain amount of most confident automatically classified examples are moved to the in-domain data with their hypothesized labels. This process is iterated until performance of the final model decreases or converges [35].

2.2.1 Self-Training

Self-training is one of the well known semi-supervised learning methods. This method improves the initial model by hypothesizing classes for the unlabeled portion of the training data with confidence scores and moves certain amount of most confident examples to labeled portion of the training data and retrain the model in each iteration [35]. It should be noted that the classifier uses its own predictions to supervise itself. This is a “hard” version of the mixture model and Expectation Maximization (EM) algorithm. Self-training is also called self-teaching, or bootstrapping in the Natural Language Processing (NLP) community. The main disadvantage of this training is that a classification mistake can reinforce itself.

Self-training method related to several unsupervised model adaptations is typically employed for speech processing systems. For instance maximum a posteriori (MAP) adaptation is one of the most popular approaches [36]. Moreover in [37] and [38] Language Model (LM) and speaker adaptation were employed on voice mail transcription application and call center spoken dialog acts application respectively. Moreover self-training applied to several language processing tasks such as speech tagging [39], word-sense disambiguation [40] and syntactic parsing [41].

2.2.2 Co-Training

The major difference between multi-view and traditional machine learning concept is that, multi-view approach consists of two or more distinguishable and sufficient feature subsets (views) rather than a single-view. The aim of multi-view approach is to improve performance of a supervised learning algorithm by incorporating large amounts of unlabeled data into the training data set. Multi-view algorithms work by generating two or more classifiers trained on different views of the labeled data that used to label the unlabeled data separately. Similar with self-training algorithm, most confidently labeled examples i.e. examples with highest confidence score are added to the manually labeled data in several iterations in order to improve the performance of the final model. However, confidence score of each example is determined by different models, which are trained in different iterations. They provide reliable confidence scores.

Co-training is one of the most effective multi-view approaches, which was first introduced in [42]. The theorems and proofs described in that work are summarized below.

Let X represent a data set which consists of two different views X_1 and X_2 such that $X = X_1 \times X_2$, and let $x \in (x_1, x_2)$ represent a single example that belongs to the data set $X = X_1 \times X_2$.

Let the distribution D be consistent with unbiased target functions such that $f_1 \in C_1$ and $f_2 \in C_2$ with $P_D[f_1(x_1) \neq f_2(x_2)] = 0$, where C_1 and C_2 are binary target classes, and let $h(X_1)$ and $h(X_2)$ are initial weak predictors. Therefore probability of weak predictor corresponds to the first view that classifies an example as class "1" given the target function described in Equation 2.1.

$$p = P_D[f(x) = 1] \geq \epsilon \tag{2.1a}$$

$$q = P_D[f(x) = 1 | h(x_1) = 1] > p + \epsilon \tag{2.1b}$$

$$c = P_D[h(x_1) = 1] \tag{2.1c}$$

$$P_D[h(x_1) = 1 | f(x) = 1] = \frac{qc}{p} \tag{2.1d}$$

$$P_D[h(x_1) = 1 | f(x) = 0] = \frac{(1-q)c}{p} \tag{2.1e}$$

Let α denote the occurrence probability of false positives and β denote the occurrence probability of false negatives. If views X_1 and X_2 provide sufficient information to train individual models which have $\alpha + \beta < 1$, and if X_2 is conditionally independent of X_1 over the distribution D given the classification, $h(x_1)$ will be independent of x_2 given the target function $f = f(f_1, f_2) \in C_1 \times C_2$. Therefore if $h(x_1)$ correspond to a noisy portion of X_2, the error rate probability of the second view is described in Equation 2.2.

$$\alpha + \beta = \left(1 - \frac{qc}{p} + \frac{(1-q)c}{1-p}\right) = \left(1 - c\frac{q-p}{p(1-p)}\right) \tag{2.2a}$$

$$\alpha + \beta \leq \left(1 - \frac{\epsilon^2}{p(1-p)}\right) \leq 1 - 4\epsilon^2 \tag{2.2b}$$

Hence, C_2 is learnable in the probably approximately correct (PAC) model with (α, β) classification noise rate, and if conditional independence assumption is satisfied, then (C_1, C_2) is learnable from unlabeled data given an initial $h(x_1)$.

In [42, 43], the aim was to identify the web pages of academic courses from a large collection of web pages from several computer science departments. The data used in that work has two natural feature sets: the words present in the course web page, and the words used in the links pointing to that web page. It has been shown that co-training was PAC learnable when the two views were individually sufficient for classification and conditionally independent given the class.

In [44], a greedy algorithm to maximize the agreement on unlabeled data was proposed. It has been shown that the rate of disagreement between two classifiers with weak independence is an upper bound on the co-training error rate and co-training was still effective under a weaker independence assumption. In [45], it has been shown that the performance of the co-training was sensitive to the learning algorithm used by applying co-training to the email classification task. Unfortunately, in that work co-training with Naive Bayes did not provide improvement. However, this situation was explained with the inability of the Naive Bayes to deal with large sparse data sets and was confirmed by improved results after feature selection. Furthermore, in [35] the relationship between the expectation-maximization (EM) algorithm and the semi-supervised learning methods was demonstrated. In addition, a hybrid approach called Co-EM, which was an iterative semi-supervised learning method, was proposed in which all the unlabeled data were exploited iteratively. In [6] co-training algorithms were extended by using two example selection mechanisms: agreement and disagreement, where in the former the examples are labeled with high confidence by both classifier, and in the latter examples that are labeled with high confidence by one classifier while labeled with low confidence in the other, and those examples are moved to the labeled in-domain data from the unlabeled out-of-domain data. In that work, prosodic and lexical information have been used in sentence segmentation problem. In [7], self-combined approach was proposed in dialog act segmentation of speech by using prosodic and lexical features, which is a combination of self-training and co-training algorithms and it has been shown that self-combined method outperformed co-training and self-training for the first iteration, however after multiple iterations, co-training resulted in better performance. Instead of simply adding machine labeled data to the set of manually labeled data as in the co-training algorithm, existing model is adapted using the machine labeled data in the co-adaptation algorithm proposed in [46]. In [47] two different views corresponding to the acoustic and lexical/syntactic knowledge sources in the Boston Radio News corpus were used in the co-training framework for automatic prosodic event labeling task. A committee-based semi-supervised approach using randomized decision trees was proposed to decrease word error rate (WER) for large-vocabulary continuous speech recognition (LVCSR) problem in [48].

Chapter 3

Data Collection and Annotation

In this work, the Voice of America (VOA)[1] Turkish Broadcast News (BN) data has been used. There are 42 Turkish BN records where each of the BN records are approximately 30 minutes long. These records include 14 speakers (7 female and 7 male) which were recorded at different acoustical environments such as studio (approximately 73% of the data), telephone conversation (approximately 14% of the data) and noisy environments (approximately 13% of the data). The BN records were recorded at a 16 bit, 16 kHz sampling rate, and corresponding transcription files, which are segment time mark (STM) and conversation time mark (CTM) files extracted in the Bosporus University BUSIM Laboratory[2]. Processing steps of CTM and STM files and overall data profile are described in sections 3.1 and 3.2, respectively. The Linguistic Data Consortium (LDC) release of this data[3] is available in [49, 50].

3.1 Pre-processing Data

Prosodic feature extraction tool which was used in this work and described in section 4.1.2 require audio files and speaker based word and phoneme transcriptions with timing informations. To provide required transcriptions i.e. forced

[1] http://www.voanews.com/turkish

[2] http://www.busim.ee.boun.edu.tr

[3] https://catalog.ldc.upenn.edu/LDC2012S06

alignments, HTK[4] (Hidden Markov Model Toolkit) based ASR system has been used in the previous work [32]. Figures 3.1 and 3.2 presents the overview and output of this process, respectively. In Fig 3.1 MFCC represents Mel-Frequency Cepstral Coefficients, HCopy and HVite represents specific tools of the Hidden Markov Model Toolkit (HTK).

Before describing the process of extraction of forced alignments using HTK as an Automatic Speech Recognition (ASR) system, first fundamental descriptions of ASR systems will be presented.

The goal of Automatic Speech Recognition (ASR) systems is to obtain text transcriptions of spoken words given a speech segment. In other words, the aim is to estimate the probability of each word given valid expressions that represents corresponding waveform (such as Mel-Frequency Cepstral Coefficients that described below) such as $W = argmax_x P(W|X)$. Using the well-known Bayes formulation this expressions can be expressed in terms of acoustic model $P_A(X|W)$ and language model $P_L W$, as shown in Equation 3.1, since $P(X)$ is independent from $P(W)$.

$$P(W|X) = \frac{P(X,W)}{P(X)} = \frac{P(X|W)P(W)}{P(X)} \tag{3.1a}$$

$$P(W|X) = P_A(X|W)P_L W \tag{3.1b}$$

The complexity of an ASR system depends on vocabulary size and length of given speech segments. For instance, Isolated Word Recognition (IWR) systems are designed to recognize words in short messages given a vocabulary list that contains a few words, such as recognizing one-digit numbers given a sequence that contains only one-digit numbers. On the other hand, Continuous Speech

[4] http://htk.eng.cam.ac.uk

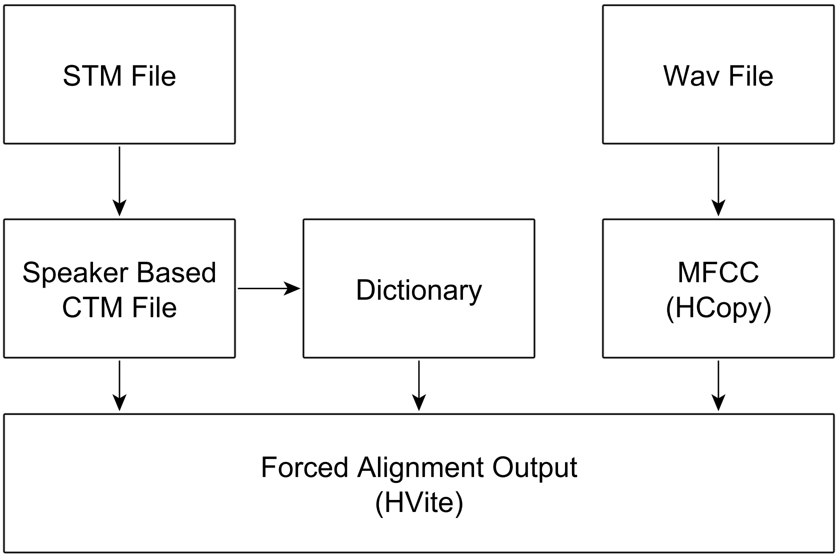


Figure 3.1: Prosodic Feature Extraction Scheme

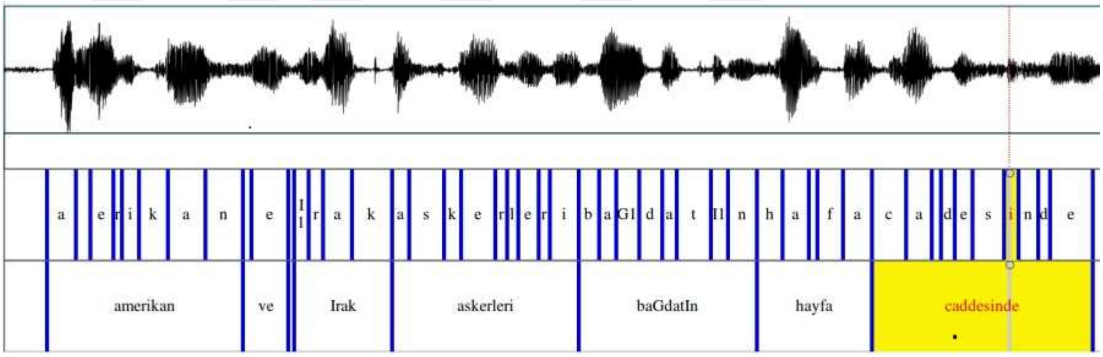


Figure 3.2: Graphical Representation of Forced Alignment Output

Recognizing (CSR) Systems has been designed to transcribe longer segments given a larger size of vocabulary, such as transcribing words in BN records.

STM (Segment Time Marks) files include several information belonging to a time-segment of speech such as initial time, final time, corresponding text transcripts, speaker-id, native/non-native speaker, gender of the speaker and acoustical background conditions of that time segment. STM files are prepared manually as reference files in order to evaluate the performance of the ASR system. An example STM file is illustrated in Table 3.1. CTM (Conversation Time Mark) files represent the output of the ASR system. CTM files include initial time, duration

```
Transcriber export by tstm.tcl,v 1.21 on Fri Nov 23 04:58:11 PM EET 2007 with encoding ISO-8859-9 transcribed by , version 3 of 071105
CATEGORY “0”
LABEL “O” “Overall” “Overall”
CATEGORY “1” “Hub4 Focus Conditions”
LABEL “F0” “Baseline Broadcast Speech”
LABEL “F1” “Spontaneous Broadcast Speech”
LABEL “F2” “Speech Over Telephone Channels”
LABEL “F3” “Speech in the Presence of Background Music”
LABEL “F4” “Speech Under Degraded Acoustic Conditions”
LABEL “F5” “Speech from Non-Native Speakers”
LABEL “FX” “All other speech”
CATEGORY “2” “Speaker Sex”
LABEL “female” “Female”
LABEL “male” “Male”
LABEL “unknown” “Unknown”
CATEGORY “3” “Topic”
LABEL “ozet” “Ozet”
LABEL “spor” “Spor”
LABEL “hava” “Hava Durumu”
LABEL “isitme” “Isitme Engelliler”
LABEL “demec” “Demec”
LABEL “ekonomi” “Ekonomi”
LABEL “haberler” “Haberler”
LABEL “unknown” “Unknown”

FM1028_0108_063000 1 excluded_region 0.000 1.800 (o,,unknown) FM1028_0108_063000 1 FM1028_0108_063000_VOA_Speaker_1
1.800 10.450 (o,f0,female,Haber) geCen hafta toplanan yeni kongrede CoGunluGu oluSturan demokrat partili Uyeler Iraka daha fazla asker
gOnderilmesine karSI CIkIyor.
FM1028_0108_063000 1 FM1028_0108_063000_VOA_Speaker_1 10.450 18.265 (o,f0,female,Haber) temsilciler meclisi baSkanI
nancy pelosi Iraktaki mevcut askerlere daha fazla Odenek ayrIlmasInI desteklediklerini bildirdi.
FM1028_0108_063000 1 FM1028_0108_063000_VOA_Speaker_1 18.265 23.575 (o,f0,female,Haber) bununla birlikte pelosi
baSkan bushdan ek asker gOnderilmesini OngOren planI
FM1028_0108_063000 1 FM1028_0108_063000_VOA_Speaker_1 23.575 28.619 (o,f0,female,Haber) ve istediGi tahsisat
hakkInda gerekCeler gOstermesi gerektiGini bildirdi.
FM1028_0108_063000 1 FM1028_0108_063000_VOA_Speaker_1 28.619 35.825 (o,f0,female,Haber) pelosi amerikan halkInIn
sonu belli olmayan bir savaSI desteklemeye mecbur edilemeyeceGini de belirtti.
FM1028_0108_063000 1 FM1028_0108_063000_VOA_Speaker_1 35.825 40.553 (o,f0,female,Haber) amerikan anayasasI baSkana
askeri kararlar alma konusunda yetki veriyor
FM1028_0108_063000 1 FM1028_0108_063000_VOA_Speaker_1 40.553 45.550 (o,f0,female,Haber) ancak savunma harcamalarInIn
arttIrIlmasI kongrenin yetkisine giriyor.
```

Table 3.1: Example of a STM File

and channel number information of each word. Table 3.2 illustrates an example of a CTM file. In addition, Mel-Frequency Cepstal Coefficients (MFCC) of audio files were extracted by HCopy tool of HTK. HVite tool requires MFCC, word transcriptions and a dictionary which includes words with corresponding phonemes to extract forced alignments. In this process HVite tool uses Viterbi Algorithm. Different force alignments were generated for different speakers based on different acoustical conditions to use open source and speaker based prosodic feature extraction tool properly. Table 3.3 presents a typical forced alignment output for the word “geçen” in terms of 100 nanoseconds.

```
FM1028_0108_063000 1 1.80 0.36 Z1
FM1028_0108_063000 1 2.16 0.31 geCen
FM1028_0108_063000 1 2.47 0.30 hafta
FM1028_0108_063000 1 2.77 0.59 toplanan
FM1028_0108_063000 1 3.36 0.26 yeni
FM1028_0108_063000 1 3.62 0.56 kongrede
FM1028_0108_063000 1 4.18 0.55 CoGunluGu
FM1028_0108_063000 1 4.73 0.47 oluSturan
FM1028_0108_063000 1 5.20 0.41 demokrat
FM1028_0108_063000 1 5.61 0.04 Z1
FM1028_0108_063000 1 5.65 0.37 partili
FM1028_0108_063000 1 6.02 0.46 Uyeler
FM1028_0108_063000 1 6.48 0.42 Z1
FM1028_0108_063000 1 6.90 0.55 Iraka
FM1028_0108_063000 1 7.45 0.23 Z1
FM1028_0108_063000 1 7.68 0.28 daha
FM1028_0108_063000 1 7.96 0.38 fazla
FM1028_0108_063000 1 8.34 0.36 asker
FM1028_0108_063000 1 8.70 0.74 gOnderilmesine
FM1028_0108_063000 1 9.44 0.32 karSI
FM1028_0108_063000 1 9.76 0.01 Z1
FM1028_0108_063000 1 9.77 0.43 CIkIyor
```

Table 3.2: Example of a CTM file

```
0 700000 silence
700000 1600000 g gecen
1600000 1900000 e
1900000 2500000 c
2500000 2800000 e
2800000 3100000 n
3100000 3900000 short pause
```

Table 3.3: Example of a Forced Alignment Output

Fundamental steps of forced alignment extraction are as follows:

Step 1: Segmentation of STM and corresponding audio files.

The open-source prosodic feature extraction tool is that was used in this work (Purdue Prosodic Feature Extraction tool described in section 4.1.2) requires speaker based speech waveforms with corresponding forced alignment outputs at the input, since this tool operates under speaker-based input assumption. Moreover, presence of different acoustical conditions in an input audio waveform will affect either word or frame based energy and pitch calculations, which are considered as a subset of prosodic features.

Original STM File WavFile1 0.000 5.000 Speaker1 transcribed words for the first segment here . WavFile1 5.000 10.000 Speaker2 transcribed words for the second segment here . WavFile1 10.000 15.000 Speaker1 transcribed words for the third segment here . WavFile1 15.000 20.000 Speaker2 transcribed words for the fourth segment here .
Constituted STM File 1 WavFile1 0.00 10.00 Speaker1 transcribed words for the first segment here transcribed words for the third segment here .
Constituted STM File 2 WavFile1 0.00 10.00 Speaker2 transcribed words for the second segment here transcribed words for the fourth segment here .

Table 3.4: Example of Generation of Speaker Based STM Files

Table 3.1 presents an example of a STM file. Those files consist of approximately 5 second speech segments with name of the corresponding audio file in "wav" format, initial and final times of the segment in terms of seconds, different speaker ID tags for different speakers and corresponding word transcripts. Speaker and acoustical condition based new STM files were constituted. Table 3.4 illustrates this process.

Corresponding waveform files has been arranged by using the linux based "**sox**" tool. This tool requires input waveform file, initial time and duration in terms of minutes and seconds as a floating number with two digits (10 milliseconds sense) and provides a corresponding waveform file of the desired segment. This tool also concatenates different audio waveform files into a single audio waveform file.

At the end of this process, at least 30 minutes long speaker and acoustical condition based STM and corresponding waveform files have been constituted. Note that each segment in the re-organized STM files should contain at least 10 - 15 minutes long speech segment to avoid existence of additional silence clues as much as possible. Afterwards master label files (MLF) are constituted from those STM files. Table 3.5 illustrates an example of a master label file.

In the example shown in Table 3.5 the labels 0 and 1000 represent indexes of corresponding MFCC vectors in the output of HCopy tool, which is described in the following step. Those labels are located in a MFC list file. Table 3.6 illustrates

```
data_WavFile1_0_1000.lab
transcribed
words
for
the
first
segment
here
transcribed
words
for
the
third
segment
here
.
```

Table 3.5: Example of a Master Label File (MLF)

```
data_WavFile1_0_1000 = WavFile1[0,1000]
data_WavFile1_1000_2000 = WavFile1[1000,2000]
```

Table 3.6: Example of a MFC List File

an example of a MFC list file. Numerical labels represent time in terms of 10 milliseconds, since each MFCC vector represents a 10 ms Hamming filtered frame.

Step 2: Extraction of Mel-Frequency Cepstral Coefficients (MFCC)

MFCC is considered as the output of a mathematical model of how the shape of vocal tract is manifested in the envelope of the short power spectrum to produce phonemes or sounds. On the other hand mel-scale translates actual frequency of a pure tone into spiral shaped human cochlea perception frequency scale such that $Mel(f) = 1125 \ln(1 + f/700)$. In most of ASR applications, extraction of MFC Coefficients is the preliminary step to express waveforms in terms of numerical values.

The implementation steps of MFCC extraction are as follows: The audio waveform is divided into 25 milliseconds frames with 15 ms overlap, since speech signal can be assumed as quasi-periodic in 20-40 ms. frames, and Hamming filter is applied into each frame to eliminate discontinuities at the edges. For each frame, periodogram estimates of power spectrum is calculated. Mel-filterbank is applied into power spectra and energy of each filter. The logarithms of each filterbank

```
Almanya a l m a n y a sil
Almanya a l m a n y a sp
arayan a r I1 y a n sil
arayan a r I1 y a n sp
Brezilya b r e z i l y a sil
Brezilya b r e z i l y a sp
Danimarka d a n i m a r k a sil
Danimarka d a n i m a r k a sp
Gelmeyen g e l m i y e n sil
Gelmeyen g e l m i y e n sp
İspanya i s p a n y a sil
İspanya i s p a n y a sp
Türkiye t U1 r k i y e sil
Türkiye t U1 r k i y e sp
. sil
```

Table 3.7: Example of a Dictionary File

energies is taken and discrete cosine transforms (DCT) of log filterbanks are calculated. Typically first 13 coefficients are considered as mel-frequency cepstral coefficients. Moreover, delta and delta-delta features represent differential and acceleration of MFCC vectors and they are calculated by using MFCC vectors.

The HCopy tool of HTK has been used to extract MFC coefficients. HCopy requires a configuration file, which includes feature extraction specifications and an input waveform file. The output format of HCopy is binary file. Therefore, one can use HList tool of HTK after using HCopy tool to obtain MFC Coefficients in terms of numerical values.

Step 3: Construction of the Dictionary file.

The dictionary file contains each transcribed word with corresponding phoneme list, in alphabetical order. In other words, when each phoneme is assumed as states of a left-right Markov model, dictionary provides state transitions for each word. Dictionary files are also used to optimize initial distribution, state transition matrix and symbol probability distribution of states to maximize the probability of observation (word) given the Hidden Markov Model (HMM). Table 3.7 illustrates an example to a dictionary file content for several words in Turkish spoken language.

Since most of Turkish words are written as it is pronounced, letters also represents phonemes (monophones). Therefore dictionary of a Turkish speech segment can be extracted easily by using a simple script. In this example in Table 3.7 the states "sp" and "sil" represents short pause and silence models, respectively.

Step 4: Extraction of forced alignments using HVite tool of HTK

The HVite tool requires a configuration file which includes specifications, master label files, MFCC list files which contains MFCC vector indexes of each segment, dictionary file and finally phoneme (monophone) list of the processed spoken language at the input. HVite provides phoneme and word durations of each transcribed word of corresponding STM files in terms of 100 nanoseconds at the output.

3.2 Data Profile

The data that has been used in this work consists of 104458 words with their corresponding prosodic, lexical, morphological features, and original labels (if it is a sentence boundary or not). The definition of features and feature extraction processes are described in Chapter 4. There are 6881 sentence boundaries at total. The data was split into a training set (61375 words and 4043 sentences, approximately 60% of the data), a development set (21533 words, 1438 sentences, approximately 20% of the data) and a test set (21550 words, 1400 sentences, approximately 20% of the data) as shown in Table 3.8.

This data includes voice records from different speakers (1 anchorwoman, 1 anchorman, 6 female and 6 male native speakers) recorded at different acoustical conditions. Tables 3.9 - 3.12 illustrate percentage distributions of training, development and test sets in terms of different speakers and different acoustical conditions.

Three different random orderings of the training set have been used and the average performance has been reported in order to get different feature distributions

	Non-sentence Boundaries	Sentence Boundaries	Total
Training Set	57332	4043	61375
Development Set	20095	1438	21533
Test Set	20150	1400	21550

Table 3.8: Number of Sentence and Non-sentence Boundaries in the Data Sets

Speaker	Gender	Environment	Non-sentence Boundary(%)	Sentence Boundary(%)	Total(%)
Alparslan Esmer	M	Studio	15.55	1.14	16.69
Alparslan Esmer	M	Noise	4.22	0.36	4.58
Arzu Çakır	F	Phone	4.87	0.29	5.15
Aydan Kızıldağlı	M	Studio	0.98	0.06	1.04
Barış Ornallı	M	Studio	2.53	0.19	2.72
Cem Dalaman	M	Studio	5.68	0.29	5.97
Cem Dalaman	M	Noise	0.51	0.03	0.54
Değer Akal	F	Phone	4.72	0.28	5.00
Değer Akal	F	Noise	1.15	0.07	1.23
Devrim Çubukçu	M	Studio	6.37	0.48	6.85
Devrim Çubukçu	M	Noise	1.02	0.10	1.12
Elif Özmenek	F	Stüdyo	4.21	0.24	4.45
Elif Ural	F	Phone	1.75	0.12	1.86
Güven Özalp	M	Studio	4.14	0.26	4.40
Hale Ebiri	F	Studio	25.56	1.91	27.47
Hale Ebiri	F	Noise	4.17	0.36	4.52
Hülya Polat	F	Studio	1.18	0.09	1.27
Mevlüt Katık	M	Phone	2.22	0.11	2.33
Özge Övün	F	Studio	1.83	0.13	1.96
Özge Övün	F	Noise	0.77	0.09	0.86
Total			93.41	6.59	100

Table 3.9: Distribution of the Training Set in terms of Speakers and Acoustical Conditions

	Non-sentence Boundary	Sentence Boundary	Total
Ordering 1	934	66	1000
	2801	199	3000
	5605	395	6000
Ordering 2	933	67	1000
	2800	200	3000
	5610	390	6000
Ordering 3	931	69	1000
	2800	200	3000
	5584	416	6000

Table 3.10: Number of Sentence and Non-sentence Boundaries at the Initial Labeled Set for each Different Random Ordering

and remove biasing effect. The word and sentence boundary distributions of initial labeled data sets for each random orderings are shown in Table 3.10. On the other hand, the development and test sets were kept same for all the experiments.

Speaker	Gender	Environment	Non-sentence Boundary(%)	Sentence Boundary(%)	Total(%)
Alparslan Esmer	M	Studio	16.34	1.19	17.53
Alparslan Esmer	M	Noise	5.45	0.52	5.97
Arzu Çakır	F	Phone	2.26	0.16	2.42
Aydan Kızıldağlı	M	Studio	0.74	0.06	0.79
Barı Ornallı	M	Studio	1.44	0.09	1.53
Cem Dalaman	M	Studio	4.93	0.24	5.17
Cem Dalaman	M	Noise	10.38	0.57	10.95
Değer Akal	F	Phone	7.70	0.46	8.16
Devrim Cubukçu	M	Studio	7.45	0.57	8.02
Devrim Cubukçu	M	Noise	0.84	0.07	0.91
Elif Özmenek	F	Studio	3.20	0.21	3.41
Elif Ural	F	Phone	2.01	0.11	2.12
Güven Özalp	M	Studio	3.84	0.25	4.09
Hale Ebiri	F	Studio	15.02	1.18	16.20
Hale Ebiri	F	Noise	5.42	0.53	5.94
Mevlut Katık	M	Phone	4.67	0.24	4.92
Özge Övün	F	Studio	0.87	0.06	0.94
Özge Övün	F	Noise	0.85	0.07	0.92
Total			93.40	6.60	100

Table 3.11: Distribution of the Development Set in terms of Speakers and Acoustical Conditions

Speaker	Gender	Environment	Non-sentence Boundary(%)	Sentence Boundary(%)	Total(%)
Alparslan Esmer	M	Studio	14.67	1.09	15.76
Alparslan Esmer	M	Noise	2.77	0.27	3.04
Arzu Çakır	F	Phone	3.47	0.18	3.65
Aydan Kızıldağlı	M	Studio	1.5	0.15	1.65
Barı Ornallı	M	Studio	3.36	0.27	3.63
Cem Dalaman	M	Studio	7.77	0.33	8.1
Cem Dalaman	M	Noise	2.37	0.15	2.51
Değer Akal	F	Phone	10.6	0.63	11.23
Değer Akal	F	Noise	2.05	0.13	2.18
Devrim Çubukçu	M	Studio	3.93	0.35	4.28
Devrim Çubukçu	M	Noise	0.9	0.1	0.99
Elif Özmenek	F	Studio	1.22	0.08	1.3
Elif Ural	F	Phone	1.83	0.11	1.95
Güven Özalp	M	Studio	4.8	0.31	5.11
Hale Ebiri	F	Studio	19.5	1.45	20.95
Hale Ebiri	F	Noise	4.24	0.36	4.6
Hülya Polat	F	Studio	1.03	0.07	1.1
Mevlüt Katık	M	Phone	2.23	0.11	2.33
Özge Övün	F	Studio	4.43	0.32	4.74
Özge Övün	F	Noise	0.81	0.11	0.91
Total			93.47	6.53	100

Table 3.12: Distribution of the Test Set in terms of Speakers and Acoustical Conditions

Chapter 4

Extraction of Prosodic, Morphological and Lexical Features for Sentence Segmentation Problem

In this chapter, feature extraction processes is summarized and feature contents are described for prosodic, morphological and lexical features in Sections 4.1, 4.2 and 4.3, respectively.

4.1 Prosodic Features

Prosody of speech includes melody, tone, rhythm and emphasis based on linguistic rules of the spoken language (syntactic structure, metrical rhythm, lexical tone, stress [51]) and current psychological conditions of the speaker. Several prosodic clues are determined by notations in written language. On the other hand, the ASR output does not provide notations, locations of several hidden speech events such as breathing, self-corrections if it is necessary and duration of little stops. Therefore, it is almost impossible to clarify either sentence or topic boundaries using the raw ASR output.

Prosodic features provide timing, duration, pitch patterns and energy patterns for each word. Since prosody of a spoken language is highly correlated with structure and semantics of the corresponding language, prosodic features provide important clues based on locations of the hidden speech events. Furthermore prosodic information is also used in several applications such as, dialog act tagging

[52, 53, 54, 55], emotion segmentation [56, 57, 58], speaker identification [59, 60, 61, 62], language identification, and analyzing characteristics of a new language without any available lexicon.

In our previous works [30, 31, 32], we have shown that a special subset of prosodic features, which designed by [4], and shown in Table 4.4, are sufficient to train strong models rather than using all of the prosodic features. Therefore, in this work, this subset of prosodic features which includes baseline measurements such as pause durations between two consecutive words and various measures of the range, movement and slope information of voiced regions, energy and voice of the speaker, have been used. In addition, several derived features such as pitch, energy before and after the current word, speaker normalized versions of them in either a certain time window or complete word, and finally maximum, minimum, average values in this range have been extracted and used.

Several types of information such as syntactic boundaries, resolving ambiguity, speaker identity, emotion, and lexical stress, that prosody is used to convey presented in [63]. First, a higher pitch at the beginning of next unit (called pitch reset in [64]), pauses, and pre-boundary lengthening are some important cues to detect syntactic boundaries. Second, meaning ambiguity in a given sentence can be resolved using duration pauses and stress. For instance the sentence "Tap the frog with the flower" in [65] may have two different meanings about the flower whether the flower is an instrument to tap frog or an object that indicates the frog. Third, speaker based pitch and energy distributions are important cues on speaker identification [60]. Fourth, activity in energy, pitch and speaking rate provide several cues on emotional state of the speaker [66]. Fifth, in some of spoken languages, the realization of word stress such as pitch accents, pitch excursions, or accents marked by changes in segmental duration or loudness may assign part of speech tag of the pronunced word.

In addition there are several well-known prosodic cues related to sentence boundaries such as longer pauses between two consecutive words [67], drop in pitch before the boundary followed by starting the next unit at a higher pitch [63].

The prosodic feature subset that used in this work, were extracted using the SRI-International Algemy prosodic feature extraction tool [68] and PRAAT[1] based Purdue prosodic feature extraction tool [29].

4.1.1 SRI-International Algemy Prosodic Feature Extraction Tool

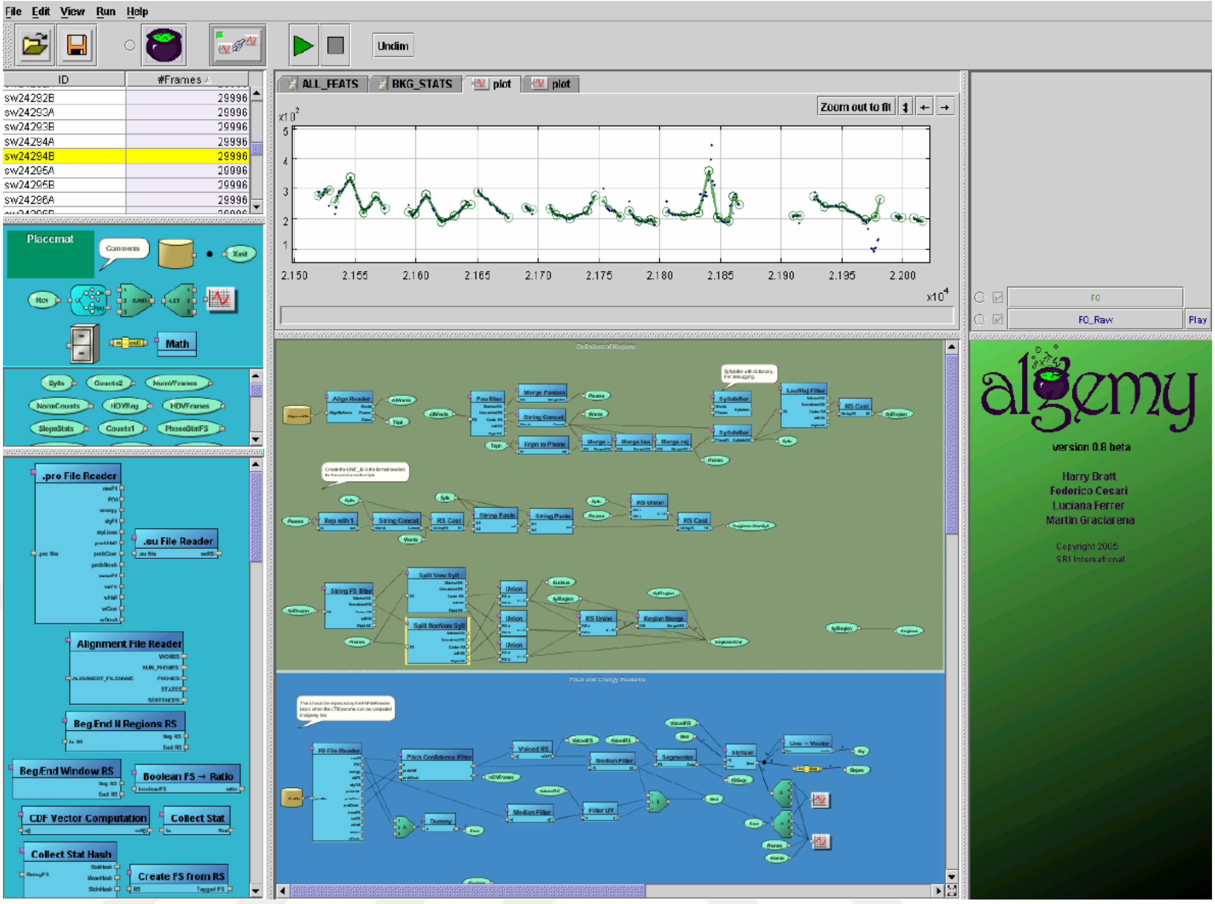
SRI-International Algemy (Algemy), a Java based commercial prosodic feature extraction tool, which was developed by Harry Bratt [26, 68, 69] at SRI-International[2], was used to extract a set of 34 prosodic features used in [3, 6, 7]. The user interface of the Algemy prosodic feature extraction tool has shown in Figure 4.1. This graphical user interface (GUI) allows researcher to modify prosodic feature extraction algorithms. Since one of the major objectives of this work and previous works [30, 31, 32] is to develop a new system using open source tools as much as possible, equivalent prosodic features of Algemy were also extracted by using PRAAT based Purdue Prosodic Feature Extraction Tool which was developed by [29]. The simplest approach of determining prosodic features is to model distributions of pitch and energy variations. Logarithm of pitch, logarithm of energy and delta features of each voiced region determine prosodic features of each frame. Those features are modeled by using Universal Background Model-Gaussian Mixture Model (UBM-GMM). Another approach is to determine pitch, energy and corresponding duration labels.

Fundamental prosodic feature groups are described below:

Pitch and Energy Features: Pitch features are classified into three groups such as range, reset and slope features. First formant frequencies are evaluated using 10ms frames over pitch tracker. Second, lognormal tied mixture (LTM) models eliminates halving and doubling errors. Third expectation maximization (EM)

[1] http://www.fon.hum.uva.nl/praat/

[2] https://www.sri.com/



ID	#Frames
sw24292B	29996
sw24293A	29996
sw24293B	29996
sw24294A	29996
sw24294B	29996
sw24295A	29996
sw24295B	29996
sw24296A	29996

Figure 4.1: User Interface of the SRI-International Algemy Prosodic Feature Extraction Tool

used to obtain speaker based statistical parameters. Extraction of energy features are similar with those steps.

Reset Features: Those features determine locations of drop in pitch followed by a pitch reset, which is an important cue on the locations of sentence boundaries. Stylized pitch contours over words are used to extract those features.

Slope Features: Those features describe pitch trajectory around the segmental boundaries. Those features are also used in speaker identification problems in [60, 20, 62].

Segmental Duration: Those features measure the length of the last vowel and rhyme preceding the boundary.

Pause Duration: Longer pauses between two consecutive words are important cues on sentence or topic boundaries.

Several fundamental prosodic features are described below:

PAUSE_DUR: The duration of pause between the current word and the next word.

PATTERN BOUNDARY: The boundary between last pattern of the word (PATTERN_WORD) in terms of “f” (fall), “r” (rise) , “u” (unvoiced region) and first pattern of the next word (PATTERN_NEXT_WORD).

SLOPE_DIFF: The difference between last non-zero slope (duration of this term is bigger than minimum frame length) of the current word and the first non-zero slope of the next word. The length of slopes which exceed minimum frame length are labeled as ×.

PAU_DUR_PREV: The pause duration between the current word and the previous word.

Several derived fundamental frequency (f0) features:

FOK_WRD_DIFF_HIHI_N: The normalized difference of maximum formant frequencies between the interested word and the next word. This feature represents log ratio between maximum piecewise linear fitted formant frequency of the current word and maximum piecewise linear fitted formant frequency of the next word.

FOK_WRD_DIFF_HILO_N: The normalized difference between maximum formant frequency of the current word and minimum formant frequency of the next word. This feature represents log ratio between maximum piecewise linear fitted formant frequency of the current word and minimum piecewise linear fitted formant frequency of the next word.

FOK_WRD_DIFF_LOHI_N: The normalized difference between minimum formant frequency of the current word and maximum formant frequency of the next word. This feature represents log ratio between minimum piecewise linear fitted formant frequency of the current word and maximum piecewise linear fitted formant frequency of the next word.

FOK_WRD_DIFF_LOLO_N: The normalized difference of minimum formant frequencies between the interested word and the next word. This feature represents log ratio between minimum piecewise linear fitted formant frequency of the current word and minimum piecewise linear fitted formant frequency of the next word.

FOK_WRD_DIFF_MNMN_N: The normalized difference of average formant frequencies between the interested word and the next word. This feature represents log ratio between average piecewise linear fitted formant frequency of the current word and average piecewise linear fitted formant frequency of the next word.

The formant frequency features related at edges of the current word:

FOK_WRD_DIFF_BEGBEG: The log ratio between the first piecewise linear fitted formant frequency (begin) of the current word and the first piecewise linear fitted formant frequency (begin) of the next word.

FOK_WRD_DIFF_ENDBEG: The log ratio between the last piecewise linear fitted formant frequency (end) of the current word and the first piecewise linear fitted formant frequency (begin) of the next word.

FOK_INWRD_DIFF: The log ratio between the first and last piecewise linear fitted formant frequency of the current word.

Features based on normalized slopes:

SLOPE_DIFF_N: This is the ratio between the measured slope difference in formant frequency and speaker related average formant frequency.

LAST_SLOPE_N: This is the ratio between the last slope of the formant frequency and last piecewise linear fitted formant frequency.

4.1.2 Purdue Prosodic Feature Extraction Tool

Prosodic feature extraction process of Purdue prosodic feature extraction tool[3] is presented in Figure 4.2 and Table 4.1 [29]. Figure 4.2 illustrates computation of statistics such as raw energy, stylized energy, energy slopes, raw pitch, voiced and unvoiced frames, stylized pitch, pitch slopes, vowel and rhyme duration, and Table 4.1 presents classes of prosodic features that extracted using those statistics.

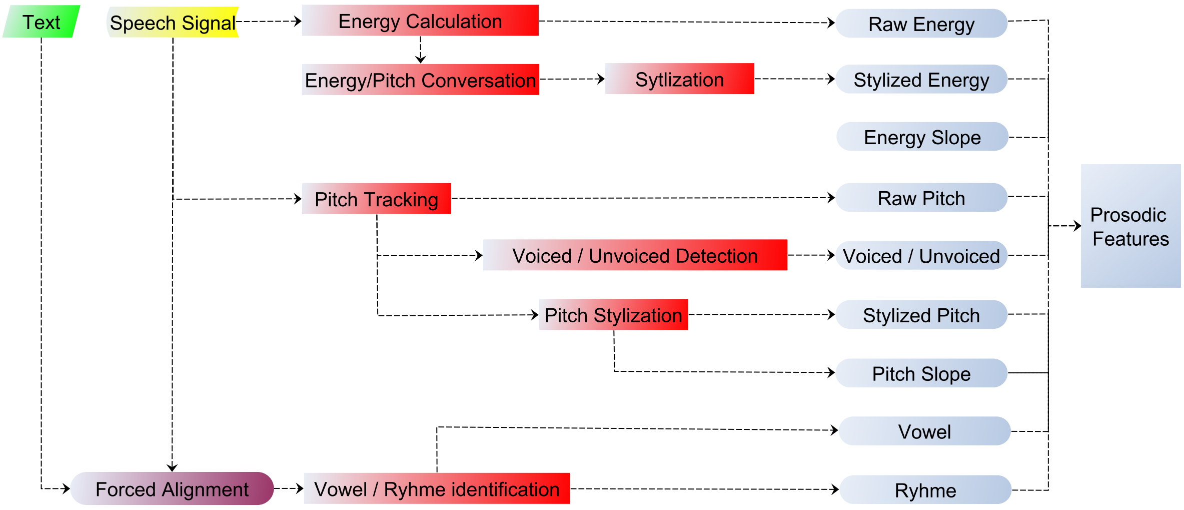


Figure 4.2: Block Diagram Representation of Purdue Prosodic Feature Extraction Tool

	Duration Features	F_O Features	Energy Features
Word	+	+	+
Phone	+	-	-
Vowel	+	-	-
Rhyme	+	-	-
VUV	-	+	-
Raw Pitch	-	+	-
Stylized Pitch	-	+	-
Pitch Slope	-	+	-
Raw Energy	-	-	+
Stylized Energy	-	-	+
Energy Slope	-	-	+

Table 4.1: The Use of Raw Files for Extracting Various Prosodic Features

Purdue prosodic feature extraction tool requires speaker based speech waveforms with corresponding word and phoneme durations (forced alignments) in the input. Extracting force alignment process is as follows: At the first step, MFCC of

[3] ftp://ftp.ecn.purdue.edu/harper/praat-prosody.tar.gz

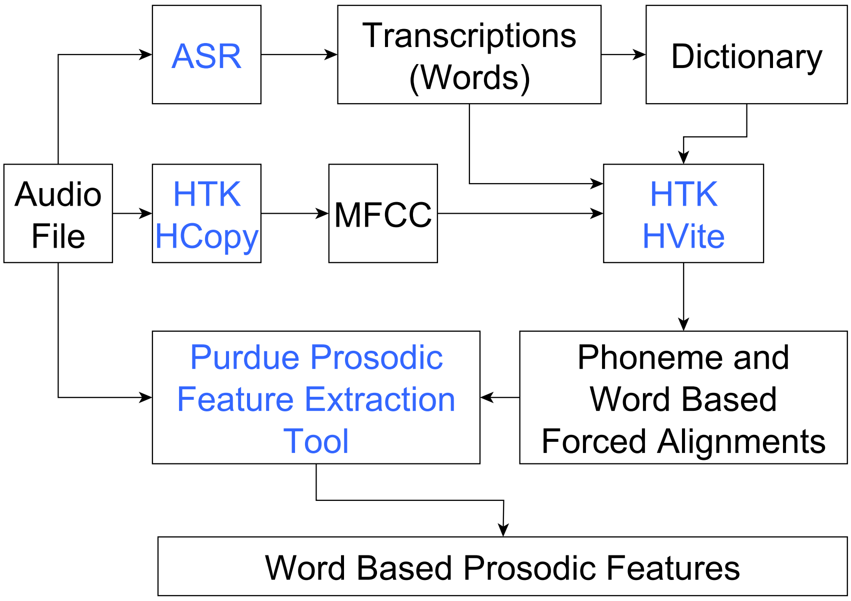
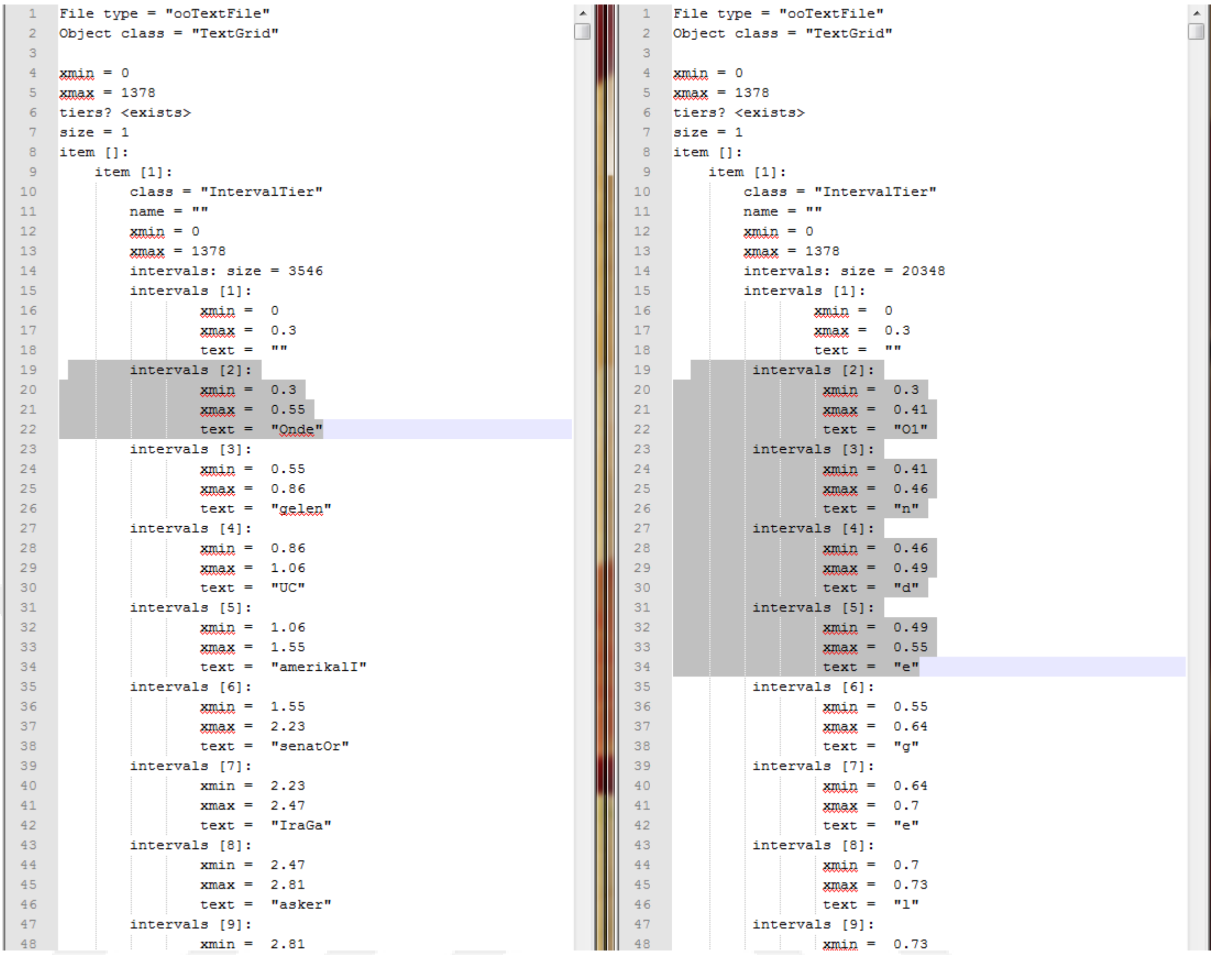

ASR
Transcriptions (Words)
Dictionary
Audio File
HTK HCopy
MFCC
HTK HVite
Purdue Prosodic Feature Extraction Tool
Phoneme and Word Based Forced Alignments
Word Based Prosodic Features

Figure 4.3: Block Diagram Representation of Forced Alignment and Prosodic Feature Extraction Process

corresponding waveforms are extracted using the HCopy tool of the HTK. Then, forced alignments of those voice records are extracted by HVite tool of the HTK, which requires corresponding trancsriptions, MFCC files and a dictionary which indicates spelling of each word in terms of phonemes. Figure 4.3 presents overall prosodic feature extraction process that has been used in this work.

Before extracting prosodic features using the Purdue prosodic feature extraction tool, the forced alignment outputs of HTK HVite tool should be converted into required "TextGrid" formats of the prosodic feature extraction tool by using a simple script. Figure 4.4 illustrates the required format of the required "TextGrid" file formats, where left side and right side presents "word.TextGrid" and "phone.TextGrid" file formats such that the former include timing information of each word and the latter include timing information of each phoneme. In addition, default phoneme list in the directory "../code/routine.praat" include a phoneme list (40 phonetic units of the English ARPAbet) based on English spoken language. Those phonemes should be replaced with the phoneme list of the Turkish spoken language for our problem, as shown in the Figure 4.5. One may also use different phoneme lists for different spoken languages. This tool requires



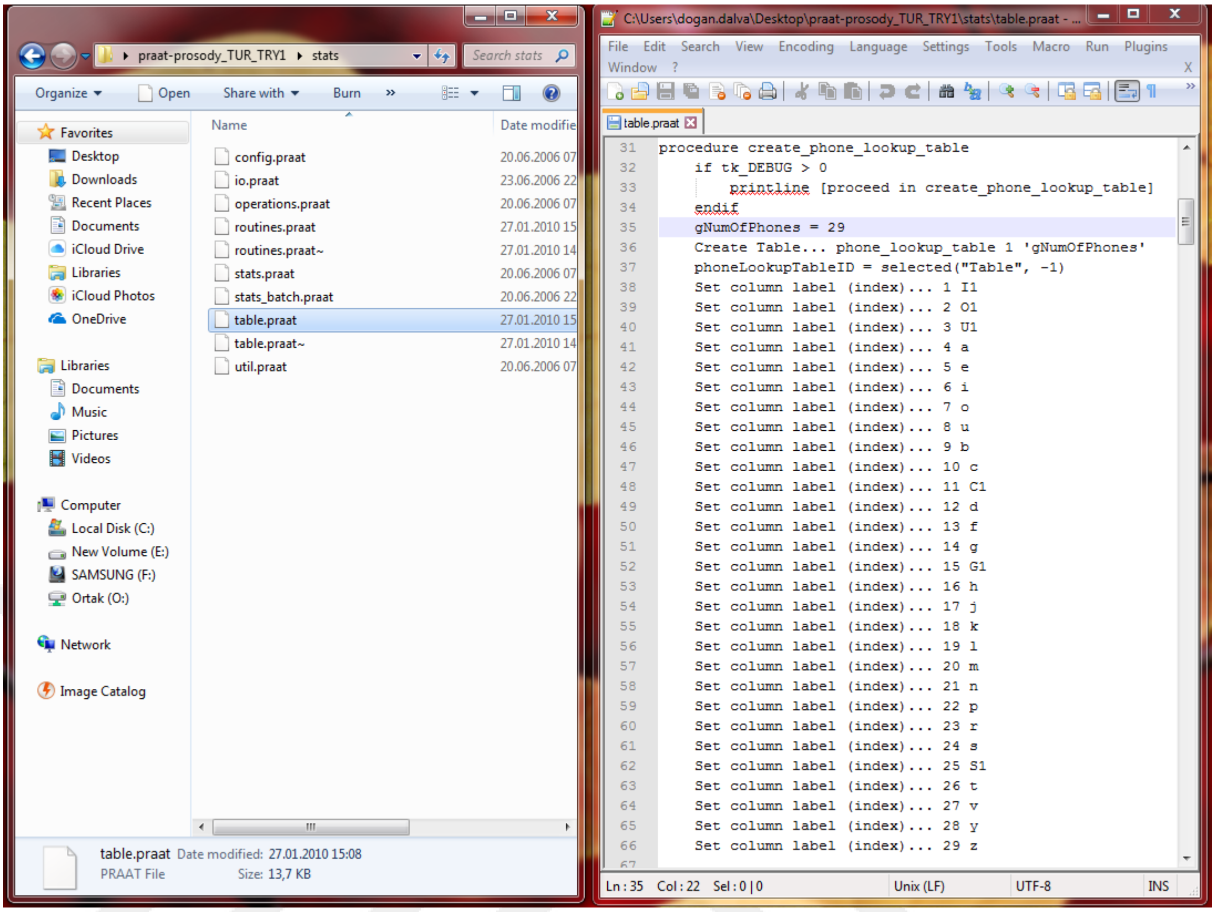
```
File type = "ooTextFile"
Object class = "TextGrid"

xmin = 0
xmax = 1378
tiers? <exists>
size = 1
item []:
    item [1]:
        class = "IntervalTier"
        name = ""
        xmin = 0
        xmax = 1378
        intervals: size = 3546
        intervals [1]:
                xmin =  0
                xmax =  0.3
                text =  ""
        intervals [2]:
                xmin =  0.3
                xmax =  0.55
                text =  "Onde"
        intervals [3]:
                xmin =  0.55
                xmax =  0.86
                text =  "gelen"
        intervals [4]:
                xmin =  0.86
                xmax =  1.06
                text =  "UC"
        intervals [5]:
                xmin =  1.06
                xmax =  1.55
                text =  "amerikalI"
        intervals [6]:
                xmin =  1.55
                xmax =  2.23
                text =  "senatOr"
        intervals [7]:
                xmin =  2.23
                xmax =  2.47
                text =  "IraGa"
        intervals [8]:
                xmin =  2.47
                xmax =  2.81
                text =  "asker"
        intervals [9]:
                xmin =  2.81
```

```
File type = "ooTextFile"
Object class = "TextGrid"

xmin = 0
xmax = 1378
tiers? <exists>
size = 1
item []:
    item [1]:
        class = "IntervalTier"
        name = ""
        xmin = 0
        xmax = 1378
        intervals: size = 20348
        intervals [1]:
                xmin =  0
                xmax =  0.3
                text =  ""
        intervals [2]:
                xmin =  0.3
                xmax =  0.41
                text =  "O1"
        intervals [3]:
                xmin =  0.41
                xmax =  0.46
                text =  "n"
        intervals [4]:
                xmin =  0.46
                xmax =  0.49
                text =  "d"
        intervals [5]:
                xmin =  0.49
                xmax =  0.55
                text =  "e"
        intervals [6]:
                xmin =  0.55
                xmax =  0.64
                text =  "g"
        intervals [7]:
                xmin =  0.64
                xmax =  0.7
                text =  "e"
        intervals [8]:
                xmin =  0.7
                xmax =  0.73
                text =  "l"
        intervals [9]:
                xmin =  0.73
```

Figure 4.4: TextGrid File Format

the following files at the input. The waveform file (16 bit 16 kHz Wav files has been used), corresponding word and phoneme alignments in “TextGrid” format and a session list which includes speaker ID, speaker gender, session name and corresponding audio waveform (wav) file.

Purdue prosodic feature extraction tool has two main operational steps which are called “Global Statistics Computation” and “Feature Extraction”. The former calculates global statistics such as speaker dependent and independent statistics, specific phone duration statistics, pitch and energy related statistics and the global phone duration statistics. The latter computes session dependent local statistics such as means and variances of last rhyme duration, the last rhyme phone duration, normalized last rhyme duration, and pause duration. Moreover “Feature Extraction” step computes derived features using the statistical values



Name	Date modifie
config.praat	20.06.2006 07
io.praat	23.06.2006 22
operations.praat	20.06.2006 07
routines.praat	27.01.2010 15
routines.praat~	27.01.2010 14
stats.praat	20.06.2006 07
stats_batch.praat	20.06.2006 22
table.praat	27.01.2010 15
table.praat~	27.01.2010 14
util.praat	20.06.2006 07

table.praat Date modified: 27.01.2010 15:08
PRAAT File Size: 13,7 KB

```
procedure create_phone_lookup_table
    if tk_DEBUG > 0
        printline [proceed in create_phone_lookup_table]
    endif
    gNumOfPhones = 29
    Create Table... phone_lookup_table 1 'gNumOfPhones'
    phoneLookupTableID = selected("Table", -1)
    Set column label (index)... 1 I1
    Set column label (index)... 2 O1
    Set column label (index)... 3 U1
    Set column label (index)... 4 a
    Set column label (index)... 5 e
    Set column label (index)... 6 i
    Set column label (index)... 7 o
    Set column label (index)... 8 u
    Set column label (index)... 9 b
    Set column label (index)... 10 c
    Set column label (index)... 11 C1
    Set column label (index)... 12 d
    Set column label (index)... 13 f
    Set column label (index)... 14 g
    Set column label (index)... 15 G1
    Set column label (index)... 16 h
    Set column label (index)... 17 j
    Set column label (index)... 18 k
    Set column label (index)... 19 l
    Set column label (index)... 20 m
    Set column label (index)... 21 n
    Set column label (index)... 22 p
    Set column label (index)... 23 r
    Set column label (index)... 24 s
    Set column label (index)... 25 S1
    Set column label (index)... 26 t
    Set column label (index)... 27 v
    Set column label (index)... 28 y
    Set column label (index)... 29 z
```

Ln : 35 Col : 22 Sel : 0 | 0 Unix (LF) UTF-8 INS

Figure 4.5: Modifying Phoneme List of the Purdue Prosodic Feature Extraction Tool

and baseline features. Figure 4.6 illustrates pitch slopes with blue lines, intensity slopes with yellow lines, formant frequencies with red dots on the spectogram in bottom, and blue lines on the time-domain speech waveform shown in top presents pulses, for a pronunced Turkish spoken segment “*tasariyi* [*short pause*] *dun*”.

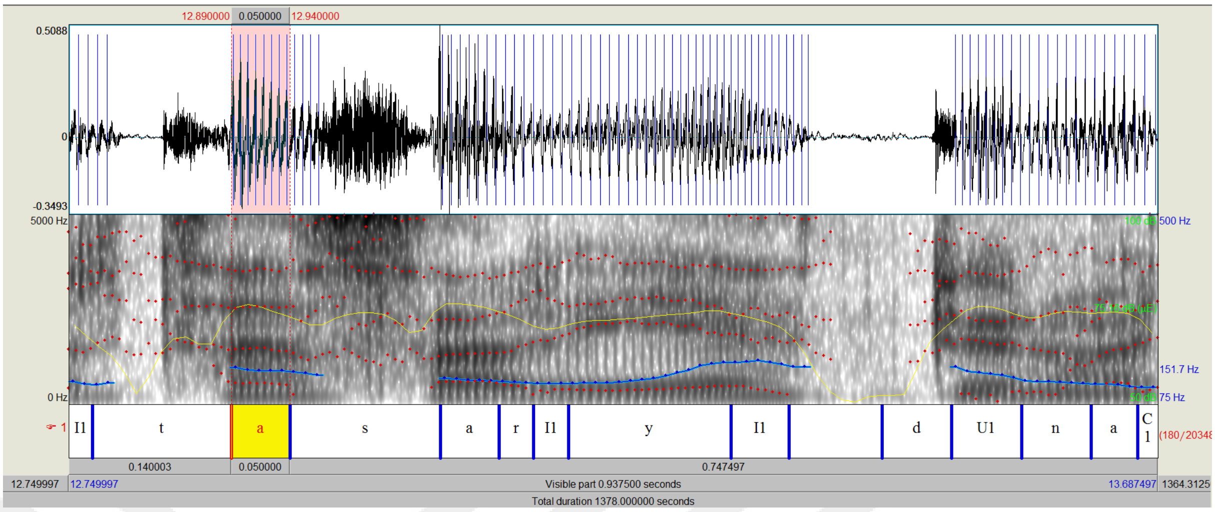
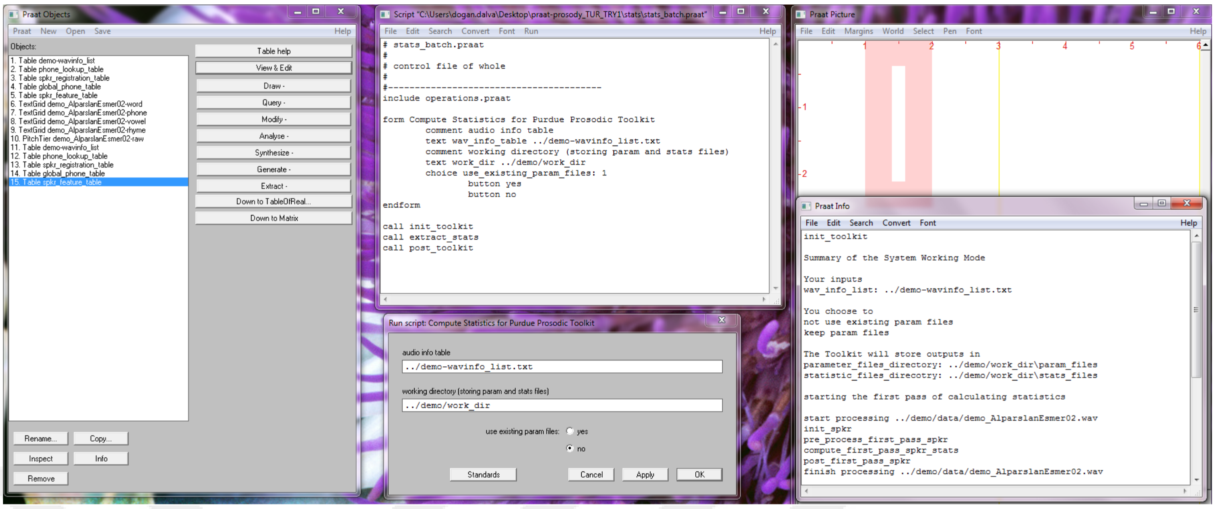


Figure 4.6: Graphical Representation of Pitch, Intensity Slopes, Formant Frequencies on Spectogram, and Pulses on the Time Domain Speech Waveform

Step 1: Global Statistics Computation
Using code: *praat stats_batch.praat../demo − wavinfo_list.txt../demo/work_dir yes*
Using user interface shown in Figure 4.7: 1. Run Praat 2. Praat Objects / Read / Read from file / *select_stats_batch.praat* 3. Click the "Run" button on the Script Editor 4. Type ../*demo − wavinfo_list.txt* and ../*demo/work_dir* into the boxes and choose the "yes option to use existing parameter files, or choose the "no" option to generate parameter files from the beginning. 5. Click "OK" 6. Process will be displayed in the "Praat Info Window".
Step 2: Prosodic Feature Extraction
Using Code: *praat main_batch.praat../demo − wavinfo_list.txt* *user_pf_name_table.Tab* ../*demo/work_dir/stats_files* ../*demo/work_dir yes*
Using user interface shown in Figure 4.8: 1. Run Praat 2. Praat Objects / Read / Read from file / *select_main_batch.praat* 3. Click the "Run" button on the Script Editor 4. Type ../*demo − wavinfo_list.txt* and ../*demo/work_dir* into the boxes and choose the "yes" option to use existing parameter files, or choose the "no" option to generate parameter files from the beginning. 5. Click "OK" 6. Process will be displayed in the "Praat Info Window".

Table 4.2: Using the Praat Based Purdue Prosodic Feature Extraction Tool



Objects:

1. Table demo-wavinfo_list
2. Table phone_lookup_table
3. Table spkr_registration_table
4. Table global_phone_table
5. Table spkr_feature_table
6. TextGrid demo_AlparslanEsmer02-word
7. TextGrid demo_AlparslanEsmer02-phone
8. TextGrid demo_AlparslanEsmer02-vowel
9. TextGrid demo_AlparslanEsmer02-rhyme
10. PitchTier demo_AlparslanEsmer02-raw
11. Table demo-wavinfo_list
12. Table phone_lookup_table
13. Table spkr_registration_table
14. Table global_phone_table
15. Table spkr_feature_table

Script "C:\Users\dogan.dalva\Desktop\praat-prosody_TUR_TRY1\stats\stats_batch.praat"

```
# stats_batch.praat
#
# control file of whole
#
#----------------------------------------
include operations.praat

form Compute Statistics for Purdue Prosodic Toolkit
        comment audio info table
        text wav_info_table ../demo-wavinfo_list.txt
        comment working directory (storing param and stats files)
        text work_dir ../demo/work_dir
        choice use_existing_param_files: 1
                button yes
                button no
endform

call init_toolkit
call extract_stats
call post_toolkit
```

Run script: Compute Statistics for Purdue Prosodic Toolkit

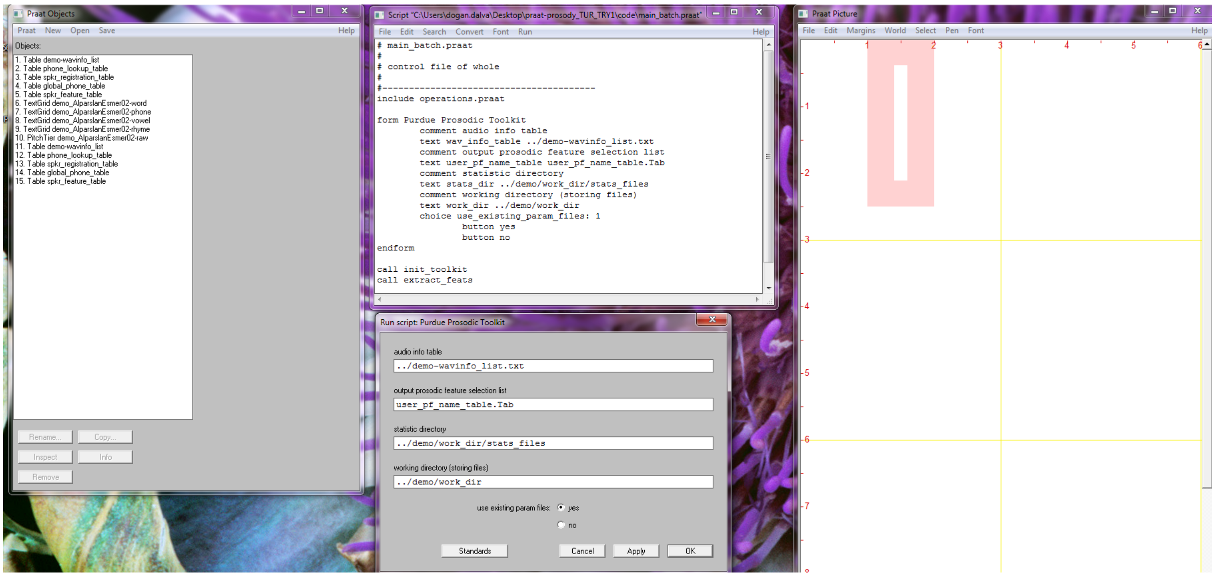
audio info table: ../demo-wavinfo_list.txt

working directory (storing param and stats files): ../demo/work_dir

use existing param files: ○ yes ● no

```
init_toolkit

Summary of the System Working Mode

Your inputs
wav_info_list: ../demo-wavinfo_list.txt

You choose to
not use existing param files
keep param files

The Toolkit will store outputs in
parameter_files_directory: ../demo/work_dir\param_files
statistic_files_direcotry: ../demo/work_dir\stats_files

starting the first pass of calculating statistics

start processing ../demo/data/demo_AlparslanEsmer02.wav
init_spkr
pre_process_first_pass_spkr
compute_first_pass_spkr_stats
post_first_pass_spkr
finish processing ../demo/data/demo_AlparslanEsmer02.wav
```

Figure 4.7: Global Statistics Computation Process of Purdue Prosodic Feature Extraction Tool



Objects:

1. Table demo-wavinfo_list
2. Table phone_lookup_table
3. Table spkr_registration_table
4. Table global_phone_table
5. Table spkr_feature_table
6. TextGrid demo_AlparslanEsmer02-word
7. TextGrid demo_AlparslanEsmer02-phone
8. TextGrid demo_AlparslanEsmer02-vowel
9. TextGrid demo_AlparslanEsmer02-rhyme
10. PitchTier demo_AlparslanEsmer02-raw
11. Table demo-wavinfo_list
12. Table phone_lookup_table
13. Table spkr_registration_table
14. Table global_phone_table
15. Table spkr_feature_table

Script "C:\Users\dogan.dalva\Desktop\praat-prosody_TUR_TRY1\code\main_batch.praat"

```
# main_batch.praat
#
# control file of whole
#
#----------------------------------------
include operations.praat

form Purdue Prosodic Toolkit
        comment audio info table
        text wav_info_table ../demo-wavinfo_list.txt
        comment output prosodic feature selection list
        text user_pf_name_table user_pf_name_table.Tab
        comment statistic directory
        text stats_dir ../demo/work_dir/stats_files
        comment working directory (storing files)
        text work_dir ../demo/work_dir
        choice use_existing_param_files: 1
                button yes
                button no
endform

call init_toolkit
call extract_feats
```

Run script: Purdue Prosodic Toolkit

audio info table: ../demo-wavinfo_list.txt

output prosodic feature selection list: user_pf_name_table.Tab

statistic directory: ../demo/work_dir/stats_files

working directory (storing files): ../demo/work_dir

use existing param files: ● yes ○ no

Figure 4.8: Prosodic Feature Extraction Process of Purdue Prosodic Feature Extraction Tool

4.1.3 Comparison of SRI Algemy and Purdue Prosodic Feature Extraction Tools

Table 4.4 presents the prosodic features that are used in this work and extracted using the Purdue prosodic feature extraction tool, and equivalent prosodic features are extracted using the Algemy prosodic feature extraction tool. Table 4.3 presents experimental results based on speaker based data sets. In these experiments, a special subset of the data presented in Section 3.2 has been used. Experimental results that presented in Table 4.3 show that sophisticated algorithms of Algemy prosodic feature extraction tool outperforms performance of the open-source PRAAT based Purdue prosodic feature extraction tool. However performance of Purdue prosodic feature extraction tool is still effective to build classifiers. Since one of the goals of this work is to build classifiers using open-source tools, prosodic features provided by Purdue prosodic feature extraction tool have been used. The experimental results presented in Table 4.3 are in terms of F-measure score and NIST error rate, where the former is harmonic mean of precision and recall, and the former is the ratio of misclassified examples over all actual sentence boundaries. Definitions of F-measure score and NIST error rate are presented in section 6.1.

Man. Labeled Data	**Prosodic Feature Extraction Tool**	**F (%)**	**NIST (%)**
1K Words	Purdue	76.9478	41.8317
1K Words	SRI Algemy	80.0791	36.9224
3K Words	Purdue	76.7717	41.5017
3K Words	SRI Algemy	79.3052	38.0863
6K Words	Purdue	76.6567	42.3267
6K Words	SRI Algemy	81.9690	33.5396
Average	Purdue	76.7920	41.8867
Average	SRI Algemy	80.4511	36.1661

Table 4.3: Average Performance Comparison of SRI-International Algemy and Purdue Prosodic Feature Extraction Tools Using Speaker Based Data Sets

Prosodic Feature Names in Purdue Tool	Corresponding Feature Names in SRI Algemy Tool	Feature Values
PAUSE_DUR	PAU_DUR	continuous.
PATTERN_BOUNDARY	F0K_PATTERN_BOUNDARY	X, f+f, f+r, r+f, r+r.
ENERGY_PATTERN_BOUNDARY	ENERGY_PATTERN_BOUNDARY	X, f+f, f+r, r+f, r+r.
SLOPE_DIFF	F0K_SLOPE_DIFF	continuous.
ENERGY_SLOPE_DIFF	ENERGY_SLOPE_DIFF	continuous.
LAST_SLOPE	F0K_LAST_SLOPE	continuous.
ENERGY_LAST_SLOPE	ENERGY_LAST_SLOPE	continuous.
LAST_SLOPE_N	F0K_LAST_SLOPE_N	continuous.
ENERGY_LAST_SLOPE_N	ENERGY_LAST_SLOPE_N	continuous.
F0K_WORD_DIFF_HIHI_N	F0K_PREVWRD1_NEXTWRD1_HIHI_N	continuous.
ENERGY_WORD_DIFF_HIHI_N	ENERGY_PREVWRD1_NEXTWRD1_HIHI_N	continuous.
F0K_WORD_DIFF_HILO_N	F0K_PREVWRD1_NEXTWRD1_HILO_N	continuous.
ENERGY_WORD_DIFF_HILO_N	ENERGY_PREVWRD1_NEXTWRD1_HILO_N	continuous.
F0K_WORD_DIFF_LOLO_N	F0K_PREVWRD1_NEXTWRD1_LOLO_N	continuous.
ENERGY_WORD_DIFF_LOLO_N	ENERGY_PREVWRD1_NEXTWRD1_LOLO_N	continuous.
F0K_WORD_DIFF_LOHI_N	F0K_PREVWRD1_NEXTWRD1_LOHI_N	continuous.
ENERGY_WORD_DIFF_LOHI_N	ENERGY_PREVWRD1_NEXTWRD1_LOHI_N	continuous.
F0K_WIN_DIFF_HIHI_N	F0K_PREVWIN20_NEXTWIN20_HIHI_N	continuous.
ENERGY_WIN_DIFF_HIHI_N	ENERGY_PREVWIN20_NEXTWIN20_HIHI_N	continuous.
F0K_WIN_DIFF_HILO_N	F0K_PREVWIN20_NEXTWIN20_HILO_N	continuous.
ENERGY_WIN_DIFF_HILO_N	ENERGY_PREVWIN20_NEXTWIN20_HILO_N	continuous.
F0K_WIN_DIFF_LOLO_N	F0K_PREVWIN20_NEXTWIN20_LOLO_N	continuous.
ENERGY_WIN_DIFF_LOLO_N	ENERGY_PREVWIN20_NEXTWIN20_LOLO_N	continuous.
F0K_WIN_DIFF_LOHI_N	F0K_PREVWIN20_NEXTWIN20_LOHI_N	continuous.
ENERGY_WIN_DIFF_LOHI_N	ENERGY_PREVWIN20_NEXTWIN20_LOHI_N	continuous.
F0K_WORD_DIFF_MNMN_N	F0K_PREVWRD1_NEXTWRD1_MNMN_N	continuous.
ENERGY_WORD_DIFF_MNMN_N	ENERGY_PREVWRD1_NEXTWRD1_MNMN_N	continuous.
F0K_WORD_DIFF_BEGBEG	F0K_PREVWRD1_NEXTWRD1_BEGBEG	continuous.
ENERGY_WORD_DIFF_BEGBEG	ENERGY_PREVWRD1_NEXTWRD1_BEGBEG	continuous.
F0K_WORD_DIFF_ENDBEG	F0K_PREVWRD1_NEXTWRD1_ENDBEG	continuous.
ENERGY_WORD_DIFF_ENDBEG	ENERGY_PREVWRD1_NEXTWRD1_ENDBEG	continuous.
F0K_INWORD_DIFF	F0K_PREVWRD1_NEXTWRD1_INWRD_DIFF	continuous.
ENERGY_INWORD_DIFF	ENERGY_PREVWRD1_NEXTWRD1_INWRD_DIFF	continuous.
PAUSE_DUR_PREV	PAU_DUR_PREV	continuous.

Table 4.4: Useful Prosodic Features for Sentence Segmentation

4.2 Morphological Features

The expression “Morphology” means word formation of a language based on patterns such as inflections, derivations and compositions. In linguistics, morphology is the study of description of the behavior and combination of morphemes, where morpheme is the smallest unit that has a meaning. Morphological features provide linguistic information about the current word.

The morphological features used in [3], [33] and this work are obtained using a morphological analyzer for Turkish developed by [27]. This tool can tag boundaries related to morphology, morphemes and part of speech (POS) tags. Those tags are useful in morphological feature extraction process. The initialization and finalization times of morphemes are evaluated using initialization and finalization times of phonemes that are provided by ASR system. Homonymic words may have different morphological analysis (uncertainity) for each different meaning. For instance the word “*bak + an*” (noun) has two meanings in Turkish: minister and “*bak + an*” where the word “*bak*” (verb) means to look and “*bak + an* (adjective) means someone looking. Different morphological analyzes for different meanings of each homonymic words are kept in the database. Determination of actual morphological analysis of a homonymic word is possible with usage of prosodic information of the corresponding word.

Despite typical constituent order of Turkish is Subject + Object + Verb, constituents can also be used in different orders. Table 4.5 presents useful morphological features for sentence segmentation and 4.6 illustrates those morphological features of a simple sentence [3] “çocuk yemek yedi” in Turkish, which means, “the child ate the meal” in English. The Morphological analysis of this sentence is as follows.

çocuk: Noun+A3sg+Pnon+Nom (the child);
yemek: Noun+A3sg+Pnon+Nom (the meal);
yedi:Verb+Pos (+dH)+Past+A3sg (ate).

Turkish has an agglutinative morphology with productive inflectional and derivational suffixations [27]. Morphological information in Turkish can be represented in general form as $root + IG_1 + DB + IG_2 + DB + ... + DB + IG_n$. In this representation, the inflectional groups (IGs) denote the derivational boundaries which are marked with "DB".

+Adj: adjective

+Noun: noun

+Verb: verb

+A3sg:3rd person singular agreement

+P1sg: 1st person singular possessive agreement

+Pnon: no possessive agreement

+Nom: nominative case

+Pos: positive polarity

+Past:past tense

+Fut: future tense

+FutPart: future participle

For instance the word "yapabilecegim" has three different morphological analysis as shown below.

1. (yap)yap+Verb+Pos(+yAbil)DB+Verb
+Able(+yAcak)+Fut(+yHm)+A1sg
I will be able to do it

2. (yap)yap+Verb+Pos(+yAbil)DB+Verb
+Able(+yAcak)DB+Adj+FutPart(+Hm)+P1sg
The (thing that) I will be able to do

3. (yap)yap+Verb+Pos(+yAbil)DB+Verb
+Able(+yAcak)DB+Noun+FutPart+A3sg(+Hm)+P1sg+Nom
The one I will be able to do

lastMarkerA3sg: 1,0.
lastMarkerNom: 1,0.
lastIGhasVerb: 1,0.
lastPOS: Adj, Adverb, Conj, Det, Dup, Interj, Noun, Num, Postp, Pron, Ques, Verb.
PrevLast3: text.
CurrentLast3: text.
NextLast3: text.
PrevCurrentLast3: text.
CurrentNextLast3: text.
PrevCurrentNextLast3: text.

Table 4.5: Useful Morphological Features for Sentence Segmentation

Word	A3SG	Nom	Verb	POS	wp	w	wn	wp-w	w-wn	wp-w-wn
Çocuk	0	1	0	Nom	?	cuk	mek	?-cuk	cuk-mek	?-cuk-mek
Yemek	0	1	0	Nom	mek	cuk	edi	cuk-mek	mek-edi	cuk-mek-edi
Yedi	1	0	1	Verb	edi	mek	?	mek-edi	edi-?	mek-edi-?

Table 4.6: Morphological Features of a Simple Sentence

This example illustrates a word which has a Verb - Adjective - Noun inflectional group (IG) order. Words in Turkish are formed by lots of part of speech (POS) tags in various orders because of productive and reproductive nature of this language.

In this work, 10 morphological features have been used. Three of them are binary features which state whether or not the current word is a verb or noun and person singular agreement (A3sg). One feature determines part of speech of the current word such as Adverb, Noun, Pronoun, Question or Verb. Since typical order of constituents in Turkish, especially in BN is Subject + Object + Verb (SOV), these four features provide strong information on the locations of the sentence boundaries. In addition to these four features, there are six pseudo-morphological features, which are formed by the last three phonemes (also letters for Turkish spoken language) combined like n-grams. Pseudo-morphological features provide the inflectional group of the interested word if it is a verb. For instance words ending with letters "dti", "ti", "di" may correspond to past tense like "ed" in English. Therefore this features indicate locations of sentence boundaries with a probabilistic view.

4.3 Lexical Features

The expression "lexical item" means the word in linguistics terminology. Lexical features consists of N-grams, where N-grams can be considered as combination of N words within a left-right HMM with a probability of occurence $P(W_N)P(W_N|W_{N-1})P(W_{N-1}|W_{N-2})...P(W_2|W_1)$ based on vocabulary size of the data. Therefore in language processing applications, lexical information provides probabilistic models (i.e. language model) based on either letter or word orders of the language [6]. In [3, 6, 7, 33] and in this work we use 6 N-gram features which are shown in Table 4.7 for each word boundary such as three unigrams (previous, current, next), two bigrams (previous-current and current-next) and one trigram (previous-current-next). Lexical features of a simple complete sentence [3] "çocuk yemek yedi" in Turkish, which means, "the child ate the meal" in English, illustrated in Table 4.8. Lexical features of a written text could be extracted by using a simple script.

Unigrams, $P(W_N)$: Previous (wp),Current (w),Next (wn)
Bigrams, $P(W_N|W_{N-1})$: Previous-Current (wp-w),Current-Next (w-wn)
Trigrams, $P(W_N|W_{N-1}, W_{N-2})$: Previous-Current-Next (wp-w-wn)

Table 4.7: Useful Lexical Features for Sentence Segmentation

Word	wp	w	wn	wp-w	w-wn	wp-w-wn
Çocuk	?	Çocuk	yemek	?-Çocuk	Çocuk-yemek	?-Çocuk-yemek
Yemek	Çocuk	yemek	yedi	Çocuk-yemek	yemek-yedi	Çocuk-yemek-yedi
Yedi	yemek	yedi	?	yemek-yedi	yedi-?	yemek-yedi-?

Table 4.8: Lexical Features of a Simple Sentence

Chapter 5

Proposed Method

This chapter presents the sentence segmentation approach of this study, multi-view semi-supervised methods, and proposed three-view co-training and committee-based learning strategies for sentence segmentation task.

5.1 Sentence Segmentation

Finding locations of each punctuation sign of an ASR output task is considered as a multi-class classification problem, where each notation sign preceded by current word is a class. However, the aim of sentence segmentation problem is to divide long output of ASR into sentences. In other words, the aim of sentence segmentation problem is to decide whether or not the current word is followed by a sentence boundary or not. Therefore, sentence segmentation task is considered as a binary sequence classification problem. Figure 5.1 illustrates sentence segmentation problem graphically.

In this problem, the aim is to estimate the posterior probability $P(y_i = H(x_i)|o_i)$ of existence a sentence boundary $y_i = (s)$ or non-sentence boundary $y_i = (n)$ between two consecutive words w_i and w_{i+1}, for a given word sequence $\{w_i, ..., w_N\}$ with feature observations o_i. The sentence boundaries are hypothesized by a binary classifier with a posterior probability, where the former and latter is represented by $H(x_i) \in \{+1, -1\}$ and $P(y_i = +1|o_i)$ respectively. When $P(y_i = +1|o_i)$

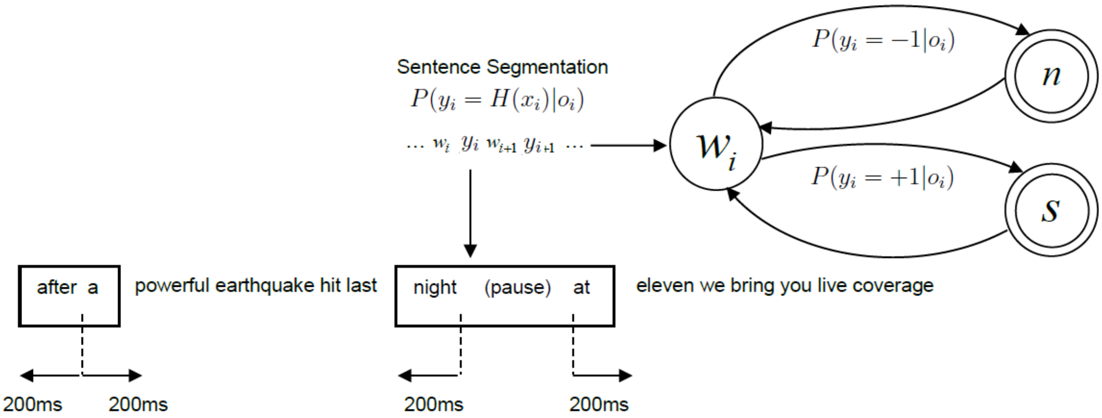


Figure 5.1: Graphical Representation of the Sentence Segmentation Problem

exceeds a certain threshold, the interested word is hypothesized as a sentence boundary [70].

5.1.1 Boosting

Boosting is one of the most well-known supervised model training method. In this work, Icsiboost[1] [71], which is an AdaBoost (Adaptive Boosting) algorithm [72] based tool, has been used.

Algorithm 1 presents the Adaboost Algorithm. This algorithm starts with an initial uniform probability distribution D_1 over all instances in the training set S. In each iteration, first a weak learner h_t is trained by using the instant distribution D_t, then error probability ϵ_t and weights α_t are evaluated, the distribution over the training set is updated based on weights and whole process is iterated until error decreases or converges at iteration T. In other words, individually weak classifiers are generated in each iteration. Finally, the hypothesized label of the interested instance $H(x)$ is determined by weighted votes of weak classifiers $h_t(x)$, rather than estimating once [73].

Final classifier provides a boosting score $f(x_i = +1)$ whose magnitude and sign represents confidence and hypothesized label for an interested instance respectively.

[1] https://github.com/benob/icsiboost

Algorithm 1 AdaBoost

```
Initialization:
1. Given training data
$S = (x_1, y_1), ..., (x_m, y_m)$ where $x_i \in X$ and
$y_i \in Y = \{-1, +1\}$
2. Initialize the distribution $D_1(i) = \frac{1}{m}$
Algorithm:
  for $t$ = 1,...,$T$ do
      Train weak learner $h_t : X$, using distribution $D_t$
      Calculate the error probability
      $\epsilon_t = P_{DT}\big(h_t(x_i) \neq y_i\big)$
      Determine weight $\alpha_t$
      $\alpha_t = \frac{1}{2} ln(\frac{1-\epsilon_t}{\epsilon_t})$
      Update the distribution over the training set:
      $D_{t+1} = \frac{D_t(i) \exp\big(-\alpha_t y_i h_t(x_i)\big)}{Z_t}$, where
      $Z_t$ is the normalization factor chosen so that $D_{t+1}$ will be a distribution.
  end for
  Final strong classifier:
  $H(X) = sign(f(x))$ where $f(x) = \sum_{t=0}^{T} \alpha_t h_t(x)$
```

5.1.2 Calibrated Confidence Scores

In self-training and co-training algorithms, which were described in section 5.2, boosting score is considered as a confidence score of the training models, in order to separate easily classified examples from harder examples. On the other hand, one may also use logistic regression methods described in [74] to obtain calibrated posterior probabilities shown in Equation 5.1 where parameters A and B are estimated by maximum likelihood estimation. Figure 5.2 illustrates plot of Equation 5.1 for different values of A and B, when it has been assumed that $f(x_i) > 0$ for sentence boundary hypothesis.

$$P(h_t(x_i) = n|x_i) = \frac{1}{1 + \exp(Af(x_i) + B)} \tag{5.1}$$

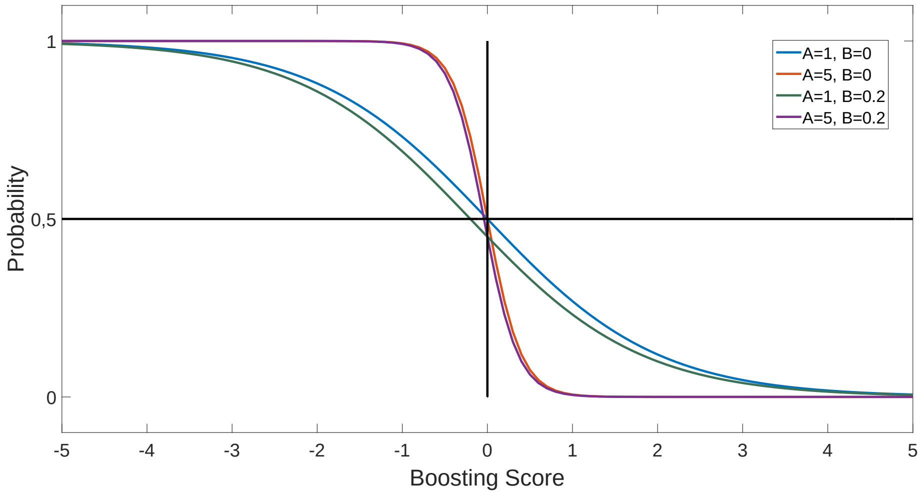


Figure 5.2: Effects of Coefficients A and B to the Posterior Probability Function

5.2 Semi-Supervised Learning

In this section, several semi-supervised learning methods such as self-training, co-training with agreement, disagreement and self-combined strategies are described. Then the proposed three-view co-training and committee-based learning strategies are described.

5.2.1 Self-Training

Self-training method is an iterative process, which is presented in Figure 5.3 and Algorithm 2. This process starts with a set of manually labeled data and a set of unlabeled data, where the former includes small amount of examples relative to the latter. In this algorithm, the classes of unlabeled examples are hypothesized with single-view models. First an initial model is built using the labeled data. Afterwards, unlabeled examples, with a boosting score exceeding a certain threshold θ, are moved to the labeled set with their hypothesized labels and the whole process is iterated.

Three different random orderings of the training set are used, then average performance is reported in order to get different feature distributions and remove

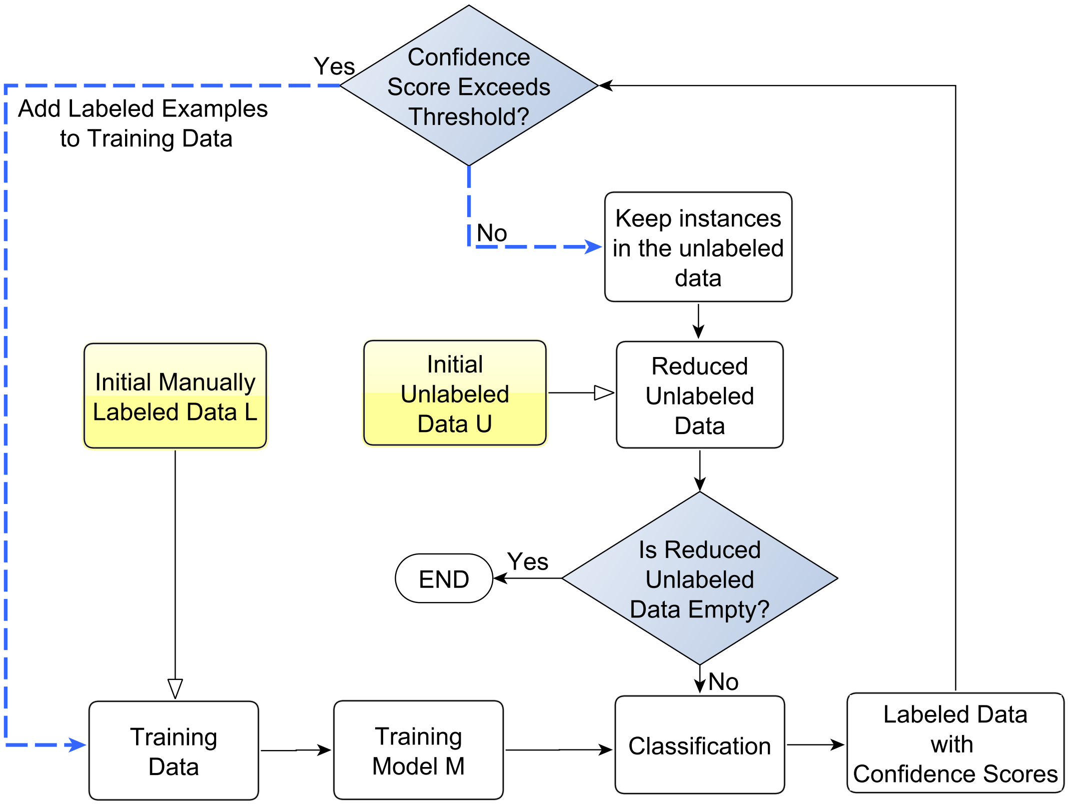

Yes
Confidence Score Exceeds Threshold?
Add Labeled Examples to Training Data
No
Keep instances in the unlabeled data
Initial Manually Labeled Data L
Initial Unlabeled Data U
Reduced Unlabeled Data
END
Yes
Is Reduced Unlabeled Data Empty?
No
Training Data
Training Model M
Classification
Labeled Data with Confidence Scores

Figure 5.3: Self-Training Scheme

biasing effect. On the other hand, the development and test sets are kept same for all the experiments. In those experiments, development and test sets are used for model optimization and performance evaluation, respectively.

Self-training algorithm is applied with the following options:

- Lexical, Prosodic, Morphological, Lexical + Morphological, Lexical + Prosodic and Prosodic + Morphological feature sets
- Different initially labeled data sizes, 1000, 3000 and 6000 words
- Different increment sizes of N, 100, 250, 500, 1000, 1500 words
- 25 iterations

Algorithm 2 Pseudo-code of the self-training algorithm

```
Initialization:
1.Given a small set of manually labeled examples,
$L = \{(x_1, y_1), ..., (x_{|L|}, y_{|L|})\}$
2.Given a large set of unlabeled examples,
$U = \{(x_1), ..., (x_{|U|})\}$
where $x_i = (x_{i,features})$ and $y_i = Y \in \{s, n\}$
Algorithm:
  while $U \neq \emptyset$ and development set error rate does not
  converge/increase do
  Train classifier $M$ from $L$
      for each $x_i \in U$ do
          if $|f_M(x_i)| > \theta$ then
              $U = U - \{x_i\}$
              $L = L \cup \{(x_i, H_M(x_i))\}$
          end if
      end for
  end while
```

Detailed algorithmic steps are described below:

Step 1: Divide the training set into the labeled and unlabeled data sets.

Divide the training set into two parts, called labeled set (L) and unlabeled set (U). Three different initial labeled data sizes which are 1000, 3000 and 6000 words, are used in order to estimate the learning curve of the self-training algorithm as well as various feature sets. Each instance x_i either in the labeled set or in the unlabeled set contains words, features and the original label. So, $x_i = (word_i, x_{i,features}, y_i)$

Step 2: Train the baseline (initial) model. Train the baseline model using the initial labeled data set. In this stage we use development set in order to decide optimum number of boosting iterations with respect to minimum error. Maximum 2000 boosting iterations are used.

Step 3: Self-training strategy.

The following steps are iterated for 25 times. Iteratively, N samples with the highest boosting scores are moved from L to U with hypothesized label, and removed from U. The value of N is set to 100, 250, 500, 1000 and 1500 words for various runs. N samples selected according to the instructions below.

Step 3.1: Hypothesize the classes $H_M(x_i)$ of each example x_i with a boosting score $f_M(x_i)$

Hypothesize the classes $H_M(x_i)$ of each example with a corresponding boosting score $f_M(x_i)$ in U using the current model. In the first iteration, the current model is the same as the baseline model.

Step 3.2: Sort instances x_i in U according to their boosting score $f_M(x_i)$

Sort the instances in U according to their boosting scores in decreased order with their corresponding hypothesized labels.

Step 3.3: Update labeled and unlabeled data sets.

Select N instances with highest boosting scores in U. Move them to L with their corresponding hypothesized labels and remove them from U. After this step new data sizes of L and U will be $|L_{ins}| + N$ and $|U_{ins}| - N$, respectively. Note that, the expression θ seen in the Algorithm 2 corresponds to the boosting score of the N$^{\text{th}}$ example.

Step 3.4: Retrain the model.

Retrain the model using updated L according to the instructions described in the Step 2.

Step 3.5: Performance evaluation of the retrained model

F-measure scores and NIST error rates of the retrained model are evaluated by using the development set and the test set. These results are recorded into a log file for each size of L. At the end of 25 iterations, optimum data size of L is selected according to the maximum F-measure score on the development set, and corresponding F-measure score and NIST error rate pair on the test set are reported as the final performance of the algorithm.

5.2.2 Co-Training

The general structure of multi-view co-training process which is shown in Figure 5.4 and Algorithm 3 is very similar to self-training process. In two-view co-training experiments, binary combinations of prosodic, lexical and morphological information are used as different view pairs. The co-training approach consists of three steps. First, two individual single-view models (M_{view1} and M_{view2}) are trained using split versions of initial manually labeled data L according to associated feature sets. For instance, for the (Lexical-Prosodic) case; $M_{Lexical}$ and $M_{Prosodic}$ models are trained by using only lexical features (L_{lex}) and only prosodic features (L_{pros}) of L, respectively. Second, the confidence scores and hypothesized labels of all examples in the unlabeled set, U, are estimated and recorded using separate M_{view1} and M_{view2} models. Third, examples which will be moved from unlabeled set to the labeled set are selected according to different example selection mechanisms such as co-training with agreement, disagreement and self-combined strategies [7].

5.2.2.1 Co-Training Agreement Strategy

In the Co-Training Agreement Strategy (Agreement), examples from the unlabeled data, which are hypothesized as same class (sentence boundary or not) by both views with high confidence score, are selected. In other words, agreement occurs when models M_{view1} and M_{view2} classifies the interested example certainly with the same decision. Agreed examples are selected in two steps. First, agreed examples are sorted according to the sum of confidence scores, then examples which has a corresponding confidence score exceeds a certain threshold are selected. The example selection mechanism of the co-training with agreement strategy is shown in Algorithm 4.

For the following example, morphological and lexical models are hypothesized for the boundary preceding the word “belirtti” with a sentence boundary (s)

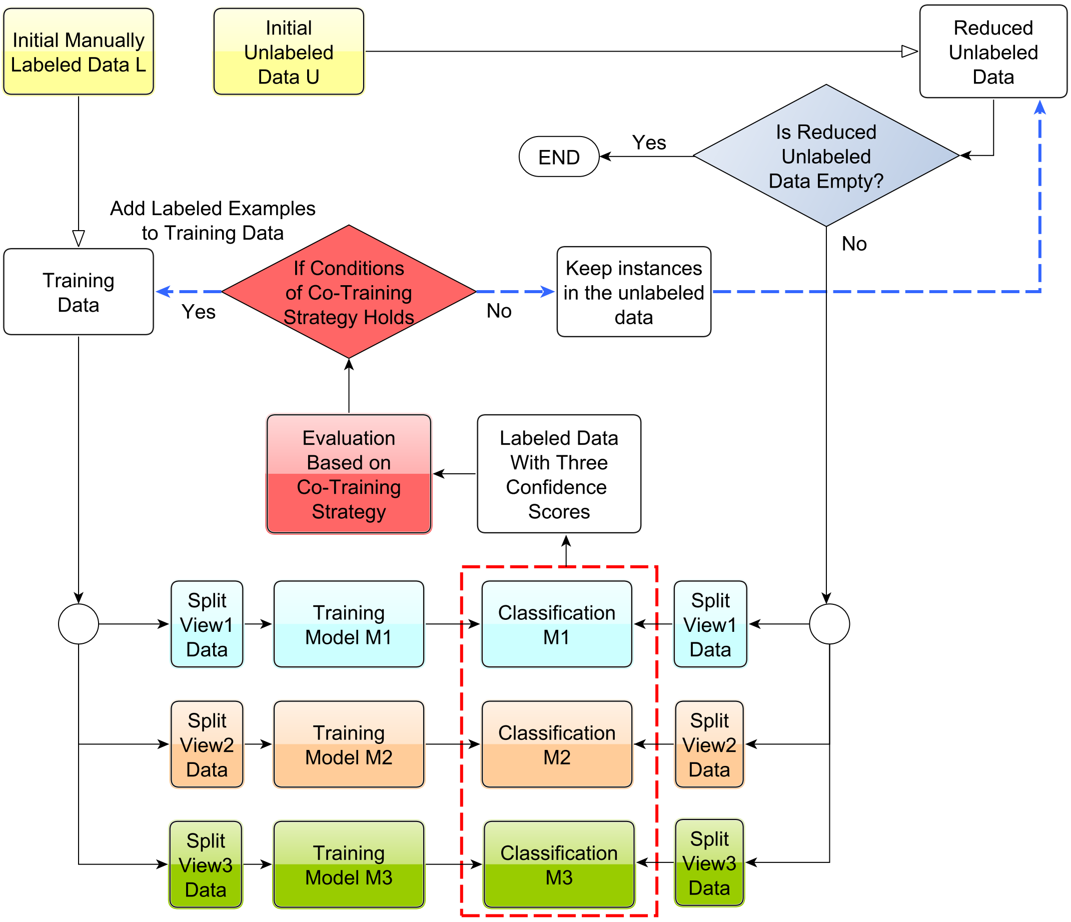


Figure 5.4: Proposed Multi-View Co-Training Scheme

with high confidence scores. Therefore, this word is transferred to labeled part of the data with hypothesized label (s) since sum of confidence scores from different views exceeded a certain threshold. From the morphological view, the word "belirtti" is a third-person singular verb, therefore morphological model is hypothesized as sentence boundary with a high confidence score. On the other hand from the lexical view, the bi-grams "istediğini-**belirtti**" and "**belirtti**-Öte" indicates high confidence to sentence boundary in probabilistic view.

Example: "...aralarındaki görüş ayrılıklarını çözme yönünde temel atmak istediğini **belirtti**. $\{SB\}$ Öte yandan..." which means "...**remarked** that he intended to start settling the disagreement between two sides. on the other hand..." where the word "remarked" represents the word "belirtti" in this translation.

Algorithm 3 Proposed Multi-View Co-Training Algorithm

```
Initialization:
1.Given a small set of manually labeled examples,
$L = \{(x_1, y_1), ..., (x_{|L|}, y_{|L|})\}$
2.Given a large set of unlabeled examples,
$U = \{(x_1), ..., (x_{|U|})\}$
where $x_i = (x_{i,features})$ and $y_i = Y \in \{s, n\}$
Algorithm:
  while $U \neq \emptyset$ and development set error rate does not
  converge/increase do
      Obtain three sets from $L$.
      $L_{view1} = \{(x_{1,view1}, y_1), ..., (x_{|L|,view1}, y_{|L|})\}$
      $L_{view2} = \{(x_{1,view2}, y_1), ..., (x_{|L|,view2}, y_{|L|})\}$
      $L_{view3} = \{(x_{1,view3}, y_1), ..., (x_{|L|,view3}, y_{|L|})\}$
      Train classifier $M_{view1}$ using $L_{view1}$.
      Train classifier $M_{view2}$ using $L_{view2}$.
      Train classifier $M_{view3}$ using $L_{view3}$.
      for each $x_i \in U$ do
          Apply example selection strategies:
          Two-view Co-Training Strategies in Algorithms 4-6 or
          Proposed Three-View Co-Training Strategies 1-7 in Algorithms 7-13 or
          Proposed Committee-Based Learning Strategies 8-9 in Algorithms 14-15
          Update data sets $L$ and $U$
      end for
  end while
```

Algorithm 4 Co-training Agreement Strategy

```
  for each $x_i \in U$ do
      if $H_{M_{view1}}(x_{i,view1}) = H_{M_{view2}}(x_{i,view2})$ and
          $|f_{M_{view1}}(x_{i,view1})| + |f_{M_{view2}}(x_{i,view2})| > \theta$ then
          $U = U - \{x_i\}$
          $L = L \cup \{(x_i, H_{M_{view1}}(x_{i,view1})\}$
      end if
  end for
```

Co-training algorithm with agreement strategy shown in Algorithm 3 is applied with the following options:

- View1=Lexical, View2=Morphological
- View1=Prosodic, View2=Lexical
- View1=Prosodic, View2=Morphological
- Different initially labeled data (L) sizes: 1000, 3000 and 6000 words
- Different increment sizes (N): 100, 250, 500, 1000 and 1500 words
- 25 iterations

Detailed algorithmic steps are described below:

Initialization: Split the Development and test sets into different views.

Split development and test sets into four different subsets according to associated feature sets of the interested views. $Dev_{view1} = (x_{i,view1}, y_i)$, $Dev_{view2} = (x_{i,view2}, y_i)$, $Test_{view1} = (x_{j,view1}, y_j)$, $Test_{view2} = (x_{j,view2}, y_j)$, where $i = 1, ..., |Dev|$ and $j = 1, ..., |Test|$ Keep these subsets same in each run.

Step 1: Divide the training set into the labeled and unlabeled data sets.

Similar with self-training algorithm, at the beginning divide the training set into two subsets that called labeled L and unlabeled U data sets. Then, split labeled and unlabeled subsets into different views such as prosodic, lexical and morphological views. Three different initial labeled data size which are 1000, 3000 and 6000 words, are used in order to estimate the learning curve of the self-training algorithm as well for various feature sets. $L_{view1} = (x_{k,view1}, y_k)$, $L_{view2} = (x_{k,view2}, y_k)$, $U_{view1} = (x_{l,view1}, y_l)$, $U_{view2} = (x_{l,view2}, y_l)$, where $k = 1, ..., |L|$ and $l = 1, ..., |U|$

Step 2: Train current models.

Train current models M_{view1}, M_{view2} and $M_{view1,view2}$ using subsets of the initial labeled data L_{view1}, L_{view2} and $L_{view1,view2}$, respectively. Note that current model will be baseline model in the first iteration. In this step, the model training procedure is similar with self-training process. Baseline performances are evaluated using $Test_{view1}$, $Test_{view2}$ and $Test_{view1,view2}$, respectively in terms of F-measure score and NIST error rate.

Step 3: Example selection mechanism.

Hypothesize examples in U_{view1} and U_{view2} with corresponding boosting scores $f_{M,view1}(x_{i,view1})$ and $f_{M,view2}(x_{i,view2})$, using the current models M_{view1} and M_{view2}, respectively. Then, sort the agreed examples in U according to a corresponding confidence score $|f_{M,view1}(x_{i,view1})| + |f_{M,view2}(x_{i,view2})|$ in decreasing order. Finally select top N examples. Move them to L with their hypothesized labels and remove them from the U. Note that the expression θ seen in the Algorithm 4 corresponds to the boosting score of the N$^{\text{th}}$ example.

5.2.2.2 Co-Training Disagreement Strategy

The Co-Training Disagreement Strategy (Disagreement) aims to classify the examples, which are hypothesized with high confidence scores by one model and low confidence scores by the other model. Thus, harder examples where one model is hypothesized certainly while the other is indecisive can be hypothesized. In this strategy, instances in the unlabeled set are sorted according to the absolute difference of absolute confidence scores, then the examples, which exceed a certain threshold are picked with corresponding hypothesis of the confident model. The example selection mechanism of the co-training with disagreement strategy is shown in Algorithm 5.

For the following example, prosodic and morphological models hypothesized the boundary that follows the word “oldugunu” differently. From the prosodic view, the word preceded by a relatively long silence is an important cue for the location of a sentence boundary however the energy and pitch characteristics of the spelling does not support that decision. Therefore prosodic model hypothesized this example as a sentence boundary with a low confidence score. On the other hand, from the morphological view, since the word “oldugunu” is not a verb and this word is a noun, morphological model hypothesized this word as a non-sentence boundary with a high confidence score. Hence, disagreement strategy truly classified this example as a non-sentence boundary.

Example: “...Amerika’nin **denetiminde** ***oldugunu n [silence]*** **iddia** etti. s Washington ise...” which means “...Claimed that this **was under** control **of** America. However Washington...”

Algorithm 5 Co-training Disagreement Strategy

```
for each $x_i \in U$ do
    if $\big||f_{M_{view1}}(x_{i,view1})| - |f_{M_{view2}}(x_{i,view2})|\big| > \theta$ then
        $U = U - \{x_i\}$
        if $|f_{M_{view1}}(x_{i,view1})| > |f_{M_{view2}}(x_{i,view2})|$ then
            $L = L \cup \{(x_i, H_{M_{view1}}(x_{i,view1})\}$
        else
            $L = L \cup \{(x_i, H_{M_{view2}}(x_{i,view2})\}$
        end if
    end if
end for
```

Co-training algorithm with disagreement strategy shown in Algorithm 5 is applied with the following options:

- View1=Lexical, View2=Morphological
- View1=Prosodic, View2=Lexical
- View1=Prosodic, View2=Morphological
- Different initially labeled data (L) sizes: 1000, 3000 and 6000 words
- Different increment sizes (N): 100, 250, 500, 1000 and 1500 words
- 25 iterations

Detailed explanation of algorithmic steps are described below.

Step 1: Divide the training set into the labeled and unlabeled data sets.

This step is the same with as first step of co-training with agreement strategy.

Step 2: Train current models.

This step is the same as the first step of co-training with agreement strategy.

Step 3: Example selection mechanism.

Hypothesize examples in U_{view1} and U_{view2} with corresponding boosting scores $f_{M,view1}(x_{i,view1})$ and $f_{M,view2}(x_{i,view2})$, using the current models M_{view1} and M_{view2}, respectively. Then assign hypothesized labels of each example in U according to decision of more confident model with corresponding confidence score $\big||f_{M_{view1}}(x_{i,view1})| - |f_{M_{view2}}(x_{i,view2})|\big|$. Sort examples in U in decreasing order. Finally select top N examples. Move them to L with their hypothesized labels and remove them from the U.

5.2.2.3 Self-Combined Strategy

Self-combined strategy is considered as the combination of self-training algorithm and co-training with the agreement strategy. At the beginning, labeled L and unlabeled U data separated into two parts such as L_{view1}, L_{view2} and U_{view1}, U_{view2} respectively. Then two individual models, M_{view1} and M_{view2} are trained by using the corresponding separated parts of L and estimated boosting scores $f_{M_{view1}(x_{i,view1})}$ and $f_{M_{view2}(x_{i,view2})}$ of the instances in U_{view1} and U_{view2} are obtained. Afterwards examples are to be moved to L are picked in two steps.

First, the examples in U_{view1} with corresponding confidence scores that exceed a certain threshold, θ_1, are selected if the estimated classes of both models are agreed for the interested example. Second, the dual of the first step is repeated for the examples in U_{view2}. The example selection mechanism of the co-training with self-combined strategy is shown in Algorithm 6.

Algorithm 6 Self-Combined Strategy

```
for each $x_i \in U_{view1}$ do
    if $|f_{M_{view1}}(x_{i,view1})| > \theta_1$ and
        $H_{M_{view1}}(x_{i,view1}) = H_{M_{view2}}(x_{i,view2})$ then
        $U = U - \{x_i\}$
        $L = L \cup \{(x_i, H_{M_{view1}}(x_{i,view1})\}$
    end if
end for
for each $x_i \in U_{view2}$ do
    if $|f_{M_{view2}}(x_{i,view2})| > \theta_2$ and
        $H_{M_{view1}}(x_{i,view1}) = H_{M_{view2}}(x_{i,view2})$ then
        $U = U - \{x_i\}$
        $L = L \cup \{(x_i, H_{M_{view2}}(x_{i,view2})\}$
    end if
end for
```

Self-combined algorithm shown in Algorithm 6 is applied with the following options:

- View1=Lexical, View2=Morphological
- View1=Prosodic, View2=Lexical
- View1=Prosodic, View2=Morphological
- Different initially labeled data (L) sizes: 1000, 3000 and 6000 words
- Different increment sizes (N): 100, 250, 500, 1000 and 1500 words
- 25 iterations

Detailed explanation of algorithmic steps are described below.

Step 1: Divide the training set into the labeled and unlabeled data sets.

This step is the same as the first step of co-training with agreement and disagreement strategies.

Step 2: Train current models.

This step is the same as the first step of co-training with agreement and disagreement strategies.

Step 3: Example selection mechanism.

The following method have been applied in order to move N examples from U to L iteratively. At the beginning, the classes $H_{M,view1}(x_{i,view1})$ and $H_{M,view2}(x_{i,view2})$ of all examples in U_{view1} and U_{view2} are predicted by the models M_{view1} and M_{view2} with corresponding boosting scores $f_{M,view1}(x_{i,view1})$ and $f_{M,view2}(x_{i,view2})$ separately. Then the examples are sorted according to their absolute boosting scores, $f_{M,view1}(x_{i,view1})$ and first $N/2$ examples are picked if predicted class of the first view $H_{M,view2}(x_{i,view2})$. Dual of this process was repeated for the examples in U_{view2}. The picked examples have been moved to the L with predicted labels and removed from the U.

Hypothesize examples in U_{view1} and U_{view2} with corresponding boosting scores $f_{M,view1}(x_{i,view1})$ and $f_{M,view2}(x_{i,view2})$, using the current models M_{view1} and M_{view2}, respectively. Then sort examples in U according to $f_{M,view1}(x_{i,view1})$ in decreased order. Pick agreed examples that exceed a certain threshold and repeat the dual of this step. Move selected examples to L with their hypothesized labels and remove them from the U.

5.2.3 Proposed Three-view Co-Training and Committee-Based Learning Strategies

In three-view approach, several algorithms has been developed which use prosodic, lexical and morphological information together. The three-view co-training approach consists of three steps. First initial manually labeled data L is split into three different views (L_{view1}, L_{view2}, L_{view3}) and three individual models (M_{view1}, M_{view2}, M_{view3}) are trained separately. Second, classes of all examples in the U are hypothesized with a corresponding confidence scores $f_{M,view1}(x_{i,view1})$, $f_{M,view2}(x_{i,view2})$, $f_{M,view3}(x_{i,view3})$ by using (M_{view1}, M_{view2} and M_{view3}). Finally, selected examples are moved from the unlabeled set to the labeled set according to nine different example selection strategies, which are shown in Table 5.1.

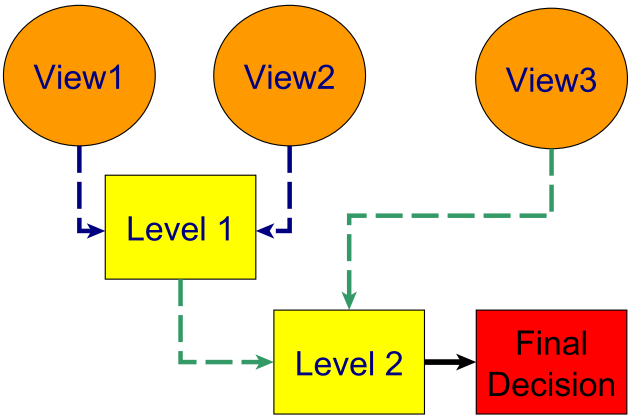


Figure 5.5: Three-View Learning Structure 1

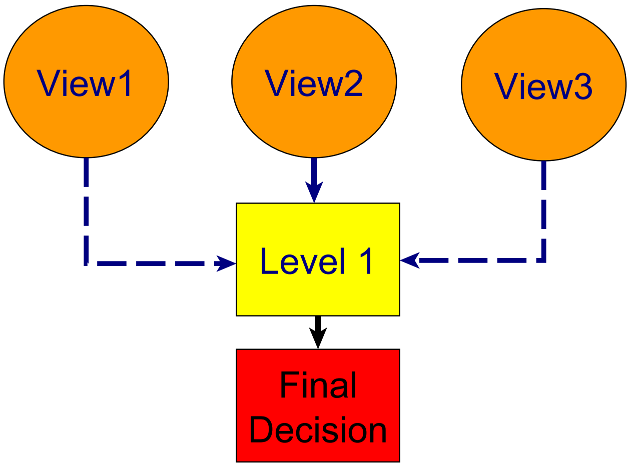


Figure 5.6: Three-View Learning Structure 2

As shown in Table 5.1 and Figures 5.5 - 5.6, three-view co-training strategies could be considered as either combined (Structure 1 in Figure 5.5) or extended (Structure 2 in Figure 5.6) versions of two-view co-training strategies. Structure 1 returns different models based on different view orders which are denoted as (View1-View2-View3). On the other hand, Structure 2 returns same models based on different view orders which are denoted as (View1+View2+View3).

In Table 5.1, there is a duality between Strategy 4 and Strategy 7 since these strategies are combination of agreement and self-combined strategies in different levels of the Structure 1, which shown in Figure 5.5. Similar duality is also holds between Strategy 5 and Strategy 6, since these strategies are combination

Strategy	Learning Structure	Level 1	Level 2
Strategy 1	Structure 2 (View1+View2+View3)	Agreement	-
Strategy 2	Structure 1 (View1-View2-View3)	Agreement	Disagreement
Strategy 3	Structure 2 (View1+View2+View3)	Self-Combined	-
Strategy 4	Structure 1 (View1-View2-View3)	Agreement	Self-Combined
Strategy 5	Structure 1 (View1-View2-View3)	Self-Combined	Disagreement
Strategy 6	Structure 1 (View1-View2-View3)	Disagreement	Self-Combined
Strategy 7	Structure 1 (View1-View2-View3)	Self-Combined	Agreement
Strategy 8	Structure 2 (View1+View2+View3)	Committee-Based Learning	
Strategy 9	Committee-Based Learning over Strategies 2,3,5,6,7,8 (View1+View2+View3)		

Table 5.1: Proposed Three-View Co-Training Strategies (Strategy 1 to 7), Committee-Based Learning Strategies (Strategy 8 and 9) and their Structures

of disagreement and self-combined strategies in different levels of the Structure 1. On the other hand, Strategy 2 is combination of agreement and disagreement strategies in the first and second levels of the Structure 1 but we do not use another strategy such that combination of disagreement and agreement strategies in the first and second levels of the Structure 1. Because this combination converges to self-training when a certain example is hypothesized by View 1 and View 2 indecisively. On the other hand, this combination converges to Strategy 1 when a certain example is hypothesized with a high confidence score by one of the views of the level 1, while the other view of level 1 is indecisive.

Algorithms 7-15 presents pseudo-codes of example selection mechanisms of three-view strategies, which are presented in Table 5.1, inserted into “apply two-view or three-view example selection mechanisms” line of Algorithm 3 in order to obtain full pseudo-codes.

Algorithms 7-15 has been applied with the following options:

- View 1=Lexical, View 2=Morphological, View 3=Prosodic
  (for cases View1+View2+View3 and View1-View2-View3)

- View 1=Lexical, View 2=Prosodic, View 3=Morphological
  (for cases View1-View2-View3)

- View 1=Morphological, View 2=Prosodic, View 3=Lexical
  (for cases View1-View2-View3)

- Different initially labeled data (L) sizes: 1000, 3000 and 6000 words
  (for cases View1+View2+View3 and View1-View2-View3)

- Different increment sizes (N): 100, 250, 500, 1000, 1500 words
  (for cases View1+View2+View3 and View1-View2-View3)

- 25 iterations
  (for cases View1+View2+View3 and View1-View2-View3)

5.2.3.1 Three-View Co-Training Strategy 1

Three-View Co-Training Strategy 1 (Strategy 1) is an extended version of two-view co-training strategy according to Structure 2 shown in Figure 5.6. First, for each agreed example, which are hypothesized as same class by different views, we calculate absolute sum of boosting scores from different views. Then we sort examples in decreasing order based on absolute sum of boosting scores. We move top N examples with their hypothesized labels from unlabeled data set U to labeled data set L and we remove them from the unlabeled data set iteratively. The example selection mechanism is shown in Algorithm 7 and Table 5.1.

The detailed explanation of Strategy 1 is described below.

Step 1: Train different models from different views. Hypothesize labels of each example in the unlabeled data set individually. Record hypothesized labels,

Algorithm 7 Three-View Co-Training Strategy 1

```
for each $x_i \in U$ do
    if $H_{M_{view1}}(x_{i,view1}) = H_{M_{view2}}(x_{i,view2}) = H_{M_{view3}}(x_{i,view3})$ and
        $|f_{M_{view1}}(x_{i,view1})| + |f_{M_{view2}}(x_{i,view2})| + |f_{M_{view3}}(x_{i,view3})| > \theta$ then
        $U = U - \{x_i\}$
        $L = L \cup \{(x_i, H_{M_{view1}}(x_{i,view1})\}$
    end if
end for
```

$H_{M_{view_j}}(x_{i,view_j})$ with corresponding boosting scores, $|f_{M_{view1}}(x_{i,view1})|$, where $j \in \{1, 2, 3\}$. Do Step 2 for each example.

Step 2: Check hypothesized labels of individual models. If three models are in agreement, assign hypothesized label with a confidence score which is the sum of absolute boosting scores of individual models, otherwise assign confidence score to zero.

Step 3: Move most confident examples from unlabeled set to labeled set with hypothesized labels.

5.2.3.2 Three-View Co-Training Strategy 2

Three-View Co-Training Strategy 2 (Strategy 2) is a combination of agreement and disagreement strategies in the first and second levels of Structure 1 which shown in Figure 5.5 and Table 5.1. Algorithm 8 presents pseudo-code of this strategy. This strategy has two levels, where first level is agreement strategy between View 1 and View 2, and level 2 is disagreement strategy between result of level 1 and View 3.

As shown in Algorithm 8 the exterior if-condition applies co-training agreement strategy between View 1 and View 2 in the first stage. In this level, for agreed examples based on View 1 and View 2 are picked and absolute sum of their boosting scores based on View 1 and View 2 is recorded as boosting score of level 1, θ_1. Afterwards for each example that selected in level 1, the score of level 1, θ_1, compared with boosting score of the third view and disagreement strategy is

Algorithm 8 Three-View Co-Training Strategy 2

```
for each $x_i \in U$ do
    if $H_{M_{view1}}(x_{i,view1}) = H_{M_{view2}}(x_{i,view2})$
        $|f_{M_{view1}}(x_{i,view1})| + |f_{M_{view2}}(x_{i,view2})| = \theta_1$ then
        if $\left|\theta_1 - |f_{M_{view3}}(x_{i,view3})|\right| > \theta_2$ then
            $U = U - \{x_i\}$
            if $\theta_1 > |f_{M_{view3}}(x_{i,view3})|$ then
                $L = L \cup \{(x_i, H_{M_{view1}}(x_{i,view1})\}$
            else
                $L = L \cup \{(x_i, H_{M_{view3}}(x_{i,view3})\}$
            end if
        end if
    end if
end for
```

applied in order to assign final hypothesized label with a corresponding confidence score.

The detailed explanation of Strategy 2 is described below.

Step 1: Train different models from different views. Hypothesize labels of each example in the unlabeled data set individually. Record hypothesized labels, $H_{M_{view_j}}(x_{i,view_j})$ with corresponding boosting scores, $|f_{M_{view1}}(x_{i,view1})|$, where $j \in \{1, 2, 3\}$. Do Step 2 for each example.

Step 2: Check hypothesized labels of View 1 and View 2. If they are agree, assign hypothesis of level1 with sum of absolute boosting scores of View 1 and View 2, as confidence score of level 1. Then apply co-training disagreement between level 1 and View 3 to decide final hypothesis with final confidence score.

Step 3: Move most confident examples from unlabeled set to labeled set with hypothesized labels.

5.2.3.3 Three-View Co-Training Strategy 3

Three-View Co-Training Strategy 3 (Strategy 3) is an extended version of two-view self-combined strategy which shown in Table 5.1. As shown in Algorithm

Algorithm 9 Three-View Co-Training Strategy 3

```
for each $x_i \in U_{view1}$ do
    if $|f_{M_{view1}}(x_{i,view1})| > \theta_1$ and
        $H_{M_{view1}}(x_{i,view1}) = H_{M_{view2}}(x_{i,view2}) = H_{M_{view3}}(x_{i,view3})$ then
        $U = U - \{x_i\}$
        $L = L \cup \{(x_i, H_{M_{view1}}(x_{i,view1})\}$
    end if
end for
for each $x_i \in U_{view2}$ do
    if $|f_{M_{view2}}(x_{i,view2})| > \theta_2$ and
        $H_{M_{view1}}(x_{i,view1}) = H_{M_{view2}}(x_{i,view2}) = H_{M_{view3}}(x_{i,view3})$ then
        $U = U - \{x_i\}$
        $L = L \cup \{(x_i, H_{M_{view2}}(x_{i,view2})\}$
    end if
end for
for each $x_i \in U_{view3}$ do
    if $|f_{M_{view3}}(x_{i,view3})| > \theta_3$ and
        $H_{M_{view1}}(x_{i,view1}) = H_{M_{view2}}(x_{i,view2}) = H_{M_{view3}}(x_{i,view3})$ then
        $U = U - \{x_i\}$
        $L = L \cup \{(x_i, H_{M_{view3}}(x_{i,view3})\}$
    end if
end for
```

9, at the beginning, the classes $H_{M,view_j}(x_{i,view_j})$,for $j \in \{1, 2, 3\}$ of all examples in U_{view1}, U_{view2} and U_{view3} are predicted by the models M_{view1}, M_{view2} and M_{view3} with corresponding boosting scores $f_{M,view_j}(x_{i,view_j})$ separately. Afterwards the examples in U_{view1} are sorted according to their absolute boosting scores, $\left|f_{M_{view1}}(x_{i,view1})\right|$ and first $N/3$ examples are picked if predicted class of the first view $H_{M_{view1}}(x_{i,view1})$ is same with the other views $H_{M_{view2}}(x_{i,view2})$ and $H_{M_{view3}}(x_{i,view3})$. Dual of this process is repeated for the examples in U_{view2} and U_{view3}. The picked instances are moved to the L with predicted labels and removed from the U. Structure 2, shown in Figure 5.6, presents the learning structure of this algorithm.

The detailed explanation of Strategy 3 is described below.

Step 1: Train different models from different views. Hypothesize labels of each example in the unlabeled data set individually. Record hypothesized labels, $H_{M_{view_j}}(x_{i,view_j})$ with corresponding boosting scores, $|f_{M_{view1}}(x_{i,view1})|$, where $j \in \{1, 2, 3\}$. Do Step 2 for each example.

Step 2: If all of views are agreed, determine biggest absolute boosting score as confidence score of the interested example with agreed hypothesized label. Otherwise assign zero to confidence score of the interested example.

Step 3: Move most confident examples from unlabeled set to labeled set with hypothesized labels.

5.2.3.4 Three-View Co-Training Strategy 4

Three-View Co-Training Strategy 4 (Strategy 4) is a combination of agreement and self-combined strategies in the first and second levels of Structure 1, which shown in Figure 5.5 and Table 5.1. Algorithm 10 presents pseudo-code of this strategy. As shown in Table 5.1, this strategy has two levels, where first level is agreement strategy between View 1 and View 2, and level 2 is self-combined strategy between result of level 1 and View 3. As shown in Algorithm 10 there are two exterior for loops which applies self-combined algorithm between decision of level 1 (agreement strategy) and View 3. Each for loop selects $N/2$ examples without repeated examples between each other.

Algorithm 10 Three-View Co-Training Strategy 4

```
for each $x_i \in U$ do
    if $|f_{M_{view1}}(x_{i,view1})| + |f_{M_{view2}}(x_{i,view2})| > \theta_1$ and
      $H_{M_{view1}}(x_{i,view1}) = H_{M_{view2}}(x_{i,view2}) = H_{M_{view3}}(x_{i,view3})$ then
      $U = U - \{x_i\}$
      $L = L \cup \{(x_i, H_{M_{view1}}(x_{i,view1}))\}$
    end if
end for
for each $x_i \in U$ do
    if $|f_{M_{view3}}(x_{i,view3})| > \theta_2$ and
      $H_{M_{view1}}(x_{i,view1}) = H_{M_{view2}}(x_{i,view2}) = H_{M_{view3}}(x_{i,view3})$ then
      $U = U - \{x_i\}$
      $L = L \cup \{(x_i, H_{M_{view3}}(x_{i,view3}))\}$
    end if
end for
```

The detailed explanation of Strategy 4 is described below.

Step 1: Train different models from different views. Hypothesize labels of each example in the unlabeled data set individually. Record hypothesized labels, $H_{M_{view_j}}(x_{i,view_j})$ with corresponding boosting scores, $|f_{M_{view1}}(x_{i,view1})|$, where $j \in \{1, 2, 3\}$. Do Step 2 for each example.

Step 2: If all views are in agreement, first apply co-training with agreement between View 1 and View 2 to assign hypothesis label and confidence score of the level 1. Then for each agreed example assign maximum of confidence score of level 1 and absolute boosting score of View 3 to final confidence score with agreed hypothesized label. If disagreement between the views occur, assign zero to the confidence score of the interested example.

Step 3: Move most confident examples from unlabeled set to labeled set with hypothesized labels.

5.2.3.5 Three-View Co-Training Strategy 5

Three-View Co-Training Strategy 5 (Strategy 5) is a combination of self-combined and disagreement strategies. Algorithm 11 and Figure 5.5 presents pseudo-code and structure of this strategy respectively. As shown in Table 5.1, this strategy has two levels, where first level is self-combined strategy between View 1 and View 2, and level 2 is disagreement strategy between result of level 1 and View 3. In pseudo-code of this algorithm which is presented in Algorithm 11 there are two exterior for loops. First for loop can be considered as a confidence score adjustment based on self-combined strategy. In this for loop, the boosting score of each agreed example based on View 1 and View 2 re-organized according to stronger model. Then in the second for loop, disagreement strategy between level 1 and View 3 is applied.

Algorithm 11 Three-View Co-Training Strategy 5

```
for each $x_i \in U$ do
    if $H_{M_{view1}}(x_{i,view1}) = H_{M_{view2}}(x_{i,view2})$ then
        $H_1 = H_{M_{view1}}(x_{i,view1})$
        if $|f_{M_{view1}}(x_{i,view1})| > |f_{M_{view2}}(x_{i,view2})|$ then
            $\theta_1 = |f_{M_{view1}}(x_{i,view1})|$
        else
            $\theta_1 = |f_{M_{view2}}(x_{i,view2})|$
        end if
    end if
end for
for each $x_i \in U$ do
    if $\big||f_{M_{view3}}(x_{i,view3})| - \theta_1\big| > \theta_2$ then
        if $|f_{M_{view3}}(x_{i,view3})| > \theta_1$ then
            $U = U - \{x_i\}$
            $L = L \cup \{(x_i, H_{M_{view3}}(x_{i,view3}))\}$
        else
            $U = U - \{x_i\}$
            $L = L \cup \{(x_i, H_1)\}$
        end if
    end if
end for
```

The detailed explanation of Strategy 5 is described below.

Step 1: Train different models from different views. Hypothesize labels of each example in the unlabeled data set individually. Record hypothesized labels, $H_{M_{view_j}}(x_{i,view_j})$ with corresponding boosting scores, $|f_{M_{view1}}(x_{i,view1})|$, where $j \in \{1, 2, 3\}$. Do Step 2 for each example.

Step 2: In the Level 1, for each example if hypotheses of View 1 and View 2 are agreed, assign hypothesis of level 1 to the agreed hypothesis with a corresponding boosting score as maximum absolute boosting scores of View 1 and View 2. Then apply two-view co-training disagreement strategy between level 1 and View 3 to determine final hypothesis with a corresponding confidence score for each agreed example in the level 1.

Step 3: Move most confident examples from labeled set to unlabeled set with hypothesized labels.

5.2.3.6 Three-View Co-Training Strategy 6

Three-View Co-Training Strategy 6 (Strategy 6) is a combination of self-combined and disagreement strategies. Algorithm 12 and Figure 5.5 presents pseudo-code and structure of this strategy respectively. As shown in Table 5.1, this strategy has two levels, where first level is disagreement strategy between View 1 and View 2, and level 2 is self-combined strategy between result of level 1 and View 3. In pseudo-code of this algorithm which is presented in Algorithm 12 there are three exterior for loops. First for loop can be considered as a disagreement between View 1 and View 2. Then, second and third loops can be considered as self-combined between result of first loop (level 1) and View 3.

Algorithm 12 Three-View Co-Training Strategy 6

```
for each $x_i \in U$ do
    $\theta_1 = \big||f_{M_{view1}}(x_{i,view1})| - |f_{M_{view2}}(x_{i,view2})|\big|$
    if $|f_{M_{view1}}(x_{i,view1})| > |f_{M_{view2}}(x_{i,view2})|$ then
        $H_1 = H_{M_{view1}}(x_{i,view1})$
    else
        $H_1 = H_{M_{view2}}(x_{i,view2})$
    end if
end for
for each $x_i \in U$ do
    if $\theta_1 > \theta_2$ and
        $H_1 = H_{M_{view3}}(x_{i,view3})$ then
        $U = U - \{x_i\}$
        $L = L \cup \{(x_i, H_{M_{view3}}(x_{i,view3})\}$
    end if
end for
for each $x_i \in U$ do
    if $|f_{M_{view3}}(x_{i,view3})| > \theta_3$ and
        $H_1 = H_{M_{view3}}(x_{i,view3})$ then
        $U = U - \{x_i\}$
        $L = L \cup \{(x_i, H_{M_{view3}}(x_{i,view3})\}$
    end if
end for
```

The detailed explanation of Strategy 6 is described below.

Step 1: Train different models from different views. Hypothesize labels of each example in the unlabeled data set individually. Record hypothesized labels,

$H_{M_{view_j}}(x_{i,view_j})$ with corresponding boosting scores, $|f_{M_{view1}}(x_{i,view1})|$, where $j \in \{1, 2, 3\}$. Do Step 2 for each example.

Step 2: In the level 1, apply two-view co-training disagreement strategy between View 1 and View 2 to assign hypothesized label of level 1 with a corresponding confidence score. Then for each agreed example assign maximum of confidence score of level 1 and absolute boosting score of View 3 to final confidence score with agreed hypothesized label. If disagreement between the views occur, assign zero to the confidence score of the interested example.

Step 3: Move most confident examples from unlabeled set to labeled set with hypothesized labels.

5.2.3.7 Three-View Co-Training Strategy 7

Three-View Co-Training Strategy 7 (Strategy 7) is a combination of self-combined and agreement strategies. Algorithm 13 and Figure 5.5 presents pseudo-code and structure of this strategy respectively. As shown in Table 5.1, this strategy has two levels, where first level is self-combined strategy between View 1 and View 2, and level 2 is agreement strategy between result of level 1 and View 3. In pseudo-code of this algorithm which presented in Algorithm 13 there are three exterior for loops. Similar to Strategy 5, first for loop can be considered as a confidence score adjustment based on self-combined strategy. In this for loop, the boosting score of each agreed example based on View 1 and View 2 are re-organized according to stronger model. Then in the second for loop, agreement strategy between level1 and View 3 is applied.

The detailed explanation of Strategy 7 is described below.

Step 1: Train different models from different views. Hypothesize labels of each example in the unlabeled data set individually. Record hypothesized labels, $H_{M_{view_j}}(x_{i,view_j})$ with corresponding boosting scores, $|f_{M_{view1}}(x_{i,view1})|$, where $j \in \{1, 2, 3\}$. Do Step 2 for each example.

Algorithm 13 Three-View Co-Training Strategy 7

```
for each $x_i \in U$ do
    if $|f_{M_{view1}}(x_{i,view1})| > |f_{M_{view2}}(x_{i,view2})|$ then
        $\theta_1 = |f_{M_{view1}}(x_{i,view1})|$
    else
        $\theta_1 = |f_{M_{view2}}(x_{i,view2})|$
    end if
end for
for each $x_i \in U$ do
    if $|f_{M_{view3}}(x_{i,view3})| + \theta_1 > \theta_2$ and
        $H_{M_{view1}}(x_{i,view1}) = H_{M_{view2}}(x_{i,view2}) = H_{M_{view3}}(x_{i,view3})$ then
        $U = U - \{x_i\}$
        $L = L \cup \{(x_i, H_{M_{view1}}(x_{i,view1})\}$
    end if
end for
```

Step 2: For each agreed examples by all of the views, assign confidence score of level 1 to maximum absolute boosting scores of View 1 and View 2, then add absolute boosting score of View 3 to determine final confidence score, assign hypothesized label to the agreed hypothesis. If disagreement occur, assign zero to the final confidence score.

Step 3: Move most confident examples from unlabeled set to labeled set with hypothesized labels.

5.2.3.8 Committee-Based Learning Strategy 8

Committee-Based Learning Strategy 8 (Strategy 8) is a kind of weighted vote approach over three view as shown in Table 5.1. Figure 5.6 presents the structure of this strategy. Since the classes of examples estimated with a boosting score, we consider hypothesized class and absolute boosting score as vote of the view and weight of the vote respectively. According to this strategy, if all of the three-view models are agreed in a certain label, sum of absolute boosting scores determines a confidence score for this example. On the other hand, if a disagreement occurs, the absolute boosting scores of majority views added and absolute boosting score of minority view subtracted in order to find the final confidence score of the interested example in the Level 1 of the Structure 2.

Algorithm 14 Committee-Based Learning Strategy 8

```
for each $x_i \in U$ do
    if $H_{M_{view1}}(x_{i,view1}) = H_{M_{view2}}(x_{i,view2}) = H_{M_{view3}}(x_{i,view3})$ then
        $H_1 = H_{M_{view1}}(x_{i,view1})$
        $\theta_1 = |f_{M_{view1}}(x_{i,view1})| + |f_{M_{view2}}(x_{i,view2})| + |f_{M_{view3}}(x_{i,view3})|$
    end if
    if $H_{M_{view1}}(x_{i,view1}) = H_{M_{view2}}(x_{i,view2}) \neq H_{M_{view3}}(x_{i,view3})$ then
        $H_1 = H_{M_{view1}}(x_{i,view1})$
        $\theta_1 = \big||f_{M_{view1}}(x_{i,view1})| + |f_{M_{view2}}(x_{i,view2})| - |f_{M_{view3}}(x_{i,view3})|\big|$
    end if
    if $H_{M_{view1}}(x_{i,view1}) = H_{M_{view3}}(x_{i,view3}) \neq H_{M_{view2}}(x_{i,view2})$ then
        $H_1 = H_{M_{view1}}(x_{i,view1})$
        $\theta_1 = \big||f_{M_{view1}}(x_{i,view1})| + |f_{M_{view3}}(x_{i,view3})| - |f_{M_{view2}}(x_{i,view2})|\big|$
    end if
    if $H_{M_{view2}}(x_{i,view2}) = H_{M_{view3}}(x_{i,view3}) H_1 \neq H_{M_{view1}}(x_{i,view1})$ then
        $H_1 = H_{M_{view2}}(x_{i,view2})$
        $\theta_1 = \big||f_{M_{view2}}(x_{i,view2})| + |f_{M_{view3}}(x_{i,view3})| - |f_{M_{view1}}(x_{i,view1})|\big|$
    end if
end for
for each $x_i \in U$ do
    if $\theta_1 > \theta_2$ then
        $U = U - \{x_i\}$
        $L = L \cup \{(x_i, H_1)\}$
    end if
end for
```

The detailed explanation of Strategy 8 is described below.

Step 1: Train different models from different views. Hypothesize labels of each example in the unlabeled data set individually. Record hypothesized labels, $H_{M_{view_j}}(x_{i,view_j})$ with corresponding boosting scores, $|f_{M_{view1}}(x_{i,view1})|$, where $j \in \{1, 2, 3\}$. Do Step 2 for each example.

Step 2: If all of models based on different views are in agreement, assign confidence score and hypothesized label as absolute sum of boosting scores and agreed label, respectively. If a disagreement occurs, assign confidence score and hypothesized label as the sum of absolute boosting scores of majority views minus absolute boosting score of the minority view, and hypothesized label by majority views, respectively.

Step 3: Move most confident examples from unlabeled set to labeled set with hypothesized labels.

5.2.3.9 Committee-Based Learning Strategy 9

Committee-Based Learning Strategy 9 (Strategy 9) is one another approach based on weighted vote over hypothesizes of Strategies 2,3,5,6,7 and 8. Algorithm 15 presents the pseudo-code of this strategy. This strategy starts with hypothesizing unlabeled data using Strategies 2,3,5,6,7 and 8 with the strongest view order of Lexical (l), Morphological (m) and Prosodic (p) views. When ideal view order of l, m and p are used, Strategy 4 and Strategy 7 provides the same hypothesis with the same confidence score for each example. In addition, Strategy 1 provides a subset of Strategy 8 outputs. Therefore, Strategy 1 and Strategy 4 are not used in Strategy 9. Different hypothesis with corresponding confidence scores are normalized over each strategy, then a committee-based approach is used to obtain final hypothesis with a corresponding confidence score. In this approach the confidence scores of examples which are hypothesized as sentence boundaries by corresponding strategies are multiplied by "-1" and confidence scores of other examples are kept the same. Then final confidence score is determined by sum of confidence scores come from different strategies, and sign of this sum determines final hypothesis. Based on the given data distribution, in each iteration the coefficients "α" and "β" are estimated by normalizing examples with respect to final confidence scores of hypothesized two classes individually in each iteration, to select sufficient amounts of examples from both hypothesized classes.

The detailed explanation of Strategy 9 is described below.

Step 1: Train different models from different views. Hypothesize labels of each example in the unlabeled data set individually. Record hypothesized labels, $H_{M_{view_j}}(x_{i,view_j})$ with corresponding boosting scores, $|f_{M_{view1}}(x_{i,view1})|$, where $j \in \{1, 2, 3\}$. Do Step 2 for each example.

Step 2: This step consists of several sub steps.

Step 2a: Obtain different confidence scores from Strategies 2,3,5,6,7,8 for each View1-View2-View3 ordering of lexical, morphological and prosodic views. For

Algorithm 15 Committee-Based Learning Strategy 9

```
for Three-View Strategies $n = 2, 3, 5, 6, 7, 8$ do
    for view orders j = 1: l-m-p, 2: l-p-m and 3: m-p-l do
        for each $x_i \in U$ do
            $H_{i,Strategy_n,order_j} = H_{M,Strategy_n,order_j}(x_i)$ and
            $f_{i,Strategy_n,order_j} = |f_{M,Strategy_n,order_j}(x_i)|$
            if $H_{i,Strategy_n,order_j}$ is undefined then
                $H_{i,Strategy_n,order_j} = N$ and
                $f_{i,Strategy_n,order_j} = 0$
            end if
        end for
    end for
    Assign $H_{i,Strategy_n}$
    with respect to $f_{i,Strategy_n} = argmax_j(f_{i,Strategy_n,order_j})$.
    Normalize confidence scores within each strategy
    $||f_{i,Strategy_n}|| = \frac{f_{i,Strategy_n}}{argmax_i(f_{i,Strategy_n})}$
    if $H_{i,Strategy_n} = S$ then
        $\hat{f}_{i,Strategy_n} = -1 * ||f_{i,Strategy_n}||$
    else
        $\hat{f}_{i,Strategy_n} = 1 * ||f_{i,Strategy_n}||$
    end if
end for
$f_M(x_i) = \sum_n(\hat{f}_{i,Strategy_n})$
if $f_M(x_i) < 0$ then
    $|f_M(x_i)| = -\alpha * f_M(x_i)$
    $H_i = S$
else
    $|f_M(x_i)| = \beta * f_M(x_i)$
    $H_i = N$
end if
for each $x_i \in U$ do
    if $|f_M(x_i)| > \theta$ then
        $U = U - \{x_i\}$
        $L = L \cup \{(x_i, H_i)\}$
    end if
end for
```

each strategy, assign the decision of most confident (highest confidence score) ordering combination of View1-View2-View3 as decision of the associated strategy. In the case of uncertainty, assign 0 to confidence score and assign non-sentence boundary (N) decision to hypothesized label of the associated strategy.

Step 2b: Normalize confidence scores within each strategy.

Step 2c: Assign a minus and plus signs for sentence boundary and non-sentence boundary labeled examples with associated magnitude of strategy based confidence scores, respectively. Then evaluate sum of those result. If this sum is smaller than zero, assign sentence boundary label (S) to final hypothesized label, then multiply the resulting sum with coefficient α and assign it to final confidence score. On the other hand if this sum is bigger than zero, assign sentence boundary label (N) to final hypothesized label, then multiply the resulting sum with coefficient β and assign it to final confidence score.

Step 3: Move most confident examples from unlabeled set to labeled set with hypothesized labels.

Chapter 6

Experiments and Results

In this work, experiments are performed by using different sizes of initial manually labeled data and newly proposed three-view co-training and committee-based learning strategies compared to two-view co-training with agreement disagreement and self-combined strategies, self-training and baseline. In these experiments, icsiboost were used as a boosting classifier.

6.1 Evaluation Metrics

Since sentence boundaries are hypothesized by a binary classifier, the description of confusion matrix follows for each example.

- True Positives (TP) represent correctly labeled sentence boundaries such that $H(X_i) = 1|y_i = 1$.
- True Negatives (TN) represent correctly labeled non-sentence boundaries such that $H(X_i) = 0|y_i = 0$.
- False Positives (FP) represent unexpected sentence boundaries, i.e. the interested example is a non-sentence boundary but hypothesized as a sentence boundary by the classifier such that $H(X_i) = 1|y_i = 0$.

- False Negatives (FN) represent missing sentence boundaries, i.e. the interested example is a sentence boundary but hypothesized as a non-sentence boundary by the classifier such that $H(X_i) = 0|y_i = 1$.

F-measure score (F_1 score when $\beta = 1$), shown in Equation 6.1, represents harmonic mean of precision and recall, where the former measures the ratio of correctly labeled sentence boundaries over all of sentence boundary decisions made by the classifier, and the latter measures the ratio of correctly labeled sentence boundaries over all actual sentence boundaries, as shown in Equations 6.2 and 6.3, respectively.

$$F-measure(\%) = 100 \times \frac{(\beta^2 + 1) \times Precision \times Recall}{\beta^2 \times Precision + Recall} \tag{6.1}$$

$$Precision = \frac{TP}{TP + FP} \tag{6.2}$$

$$Recall = \frac{TP}{TP + FN} \tag{6.3}$$

NIST (National Institute Standards and Technology) error rate is one of the most common used and well-known performance evaluation measures. The NIST error rate is the ratio of total misclassified examples over total actual sentence boundaries as shown in Equation 6.4.

$$NIST(\%) = 100 \times \frac{FN + FP}{TP + FN} \tag{6.4}$$

6.2 Baseline Results of Different Feature Sets

Before presenting experimental results of semi-supervised algorithms, baseline results of the lexical, prosodic, morphological and binary combination models are

Data Size	F-measure (%)						NIST (%)					
(Words)	l	m	p	l+m	l+p	m+p	l	m	p	l+m	l+p	m+p
1K	26.31	75.22	72.19	75.37	73.20	81.52	91.38	49.81	52.00	48.86	50.38	35.19
3K	42.90	75.51	71.75	76.41	72.65	82.31	92.14	48.74	55.00	48.55	49.33	33.43
6K	54.41	76.30	70.31	77.14	75.21	83.25	73.02	47.43	55.79	45.48	45.00	32.21
10K	59.81	78.05	71.90	78.35	77.41	83.55	68.86	43.47	51.78	42.88	40.86	31.78
30K	66.97	78.81	73.39	80.65	79.89	85.01	56.81	42.90	49.31	38.14	37.09	29.17
60K	69.50	79.36	74.18	81.96	79.64	86.47	52.95	41.55	48.64	36.19	37.79	26.26

Table 6.1: Baseline Results of Different Feature Sets With Different Data Sizes

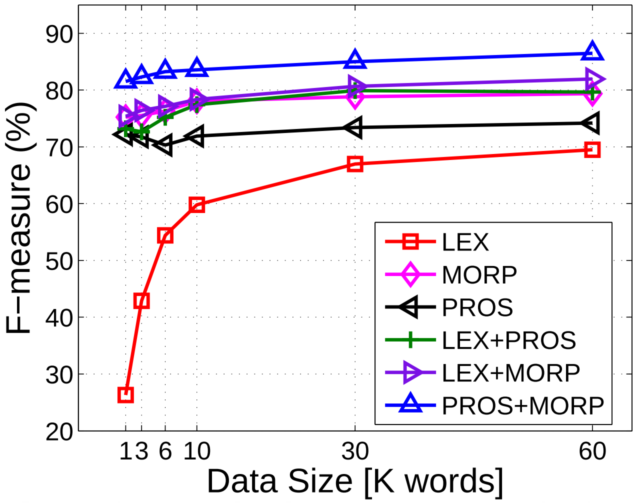


Figure 6.1: F-measure Scores of Different Feature Sets With Different Data Sizes

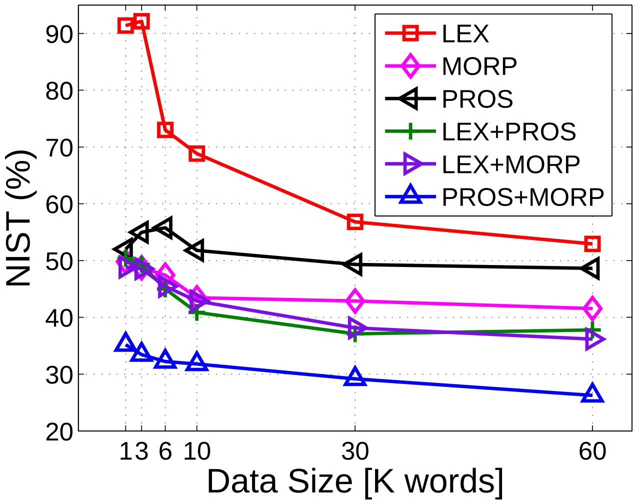


Figure 6.2: NIST Error Rates of Different Feature Sets With Different Data Sizes

presented for different training data sizes such as 1K, 3K, 6K, 10K, 30K and 60K words in Figures 6.1, 6.2 and Table 6.1. The curves in Figures 6.1 and 6.2 shows that lexical (l), prosodic (p) and morphological (m) information are sufficient to train binary classifiers for sentence detection problem. Moreover, since total training data size is 61375 words, baseline results of various models with 60K words initial data size could considered as a kind of boundary for each model.

Ternary combination of lexical, prosodic, and morphological features has not been used because comparative results of baseline, self-training, two-view co-training algorithms (agreement, disagreement, and self-combined), and proposed three-view co-training and committee-based algorithms has been presented.

6.3 Experimental Setup of the Self-Training and Co-Training Methods

At the beginning, the data sets are divided into a training set, a development set, and a test set, which are identically distributed over various speakers and acoustical conditions, without any overlap. In other words, the training, development and test sets has been balanced in terms of speaker variability and acoustical conditions. This approach helps to prevent the biasing effect because of the balanced distribution of the data into the training, development and test sets while keeping non-overlapping rather than choosing the bunch of examples randomly. The details of the experimental setup and evaluation process are illustrated in Figure 6.3.

Three different orderings of the training set, which shown in the following Figure 6.4 has been used. The training set has divided into a labeled L and unlabeled U data set, by assigning first L samples with their original labels (i.e., words with their corresponding features and labels). For Turkish BN data, three different sizes of L which are 1K, 3K, and 6K has been used. Using three different orderings of the training set provides three different initial manually labeled data sets. For instance, when orderings shown in the Fig 6.4 are considered, there are three

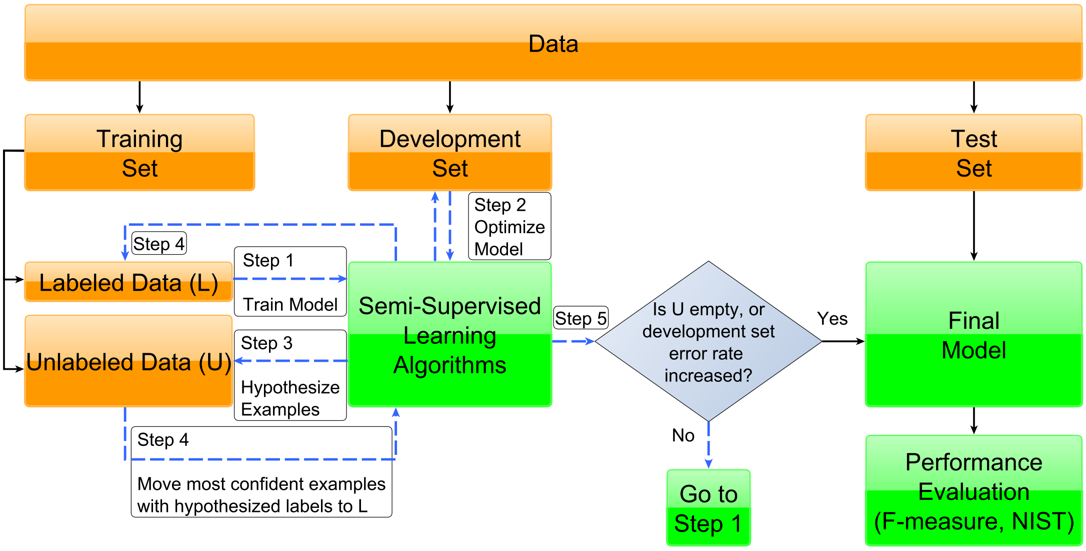
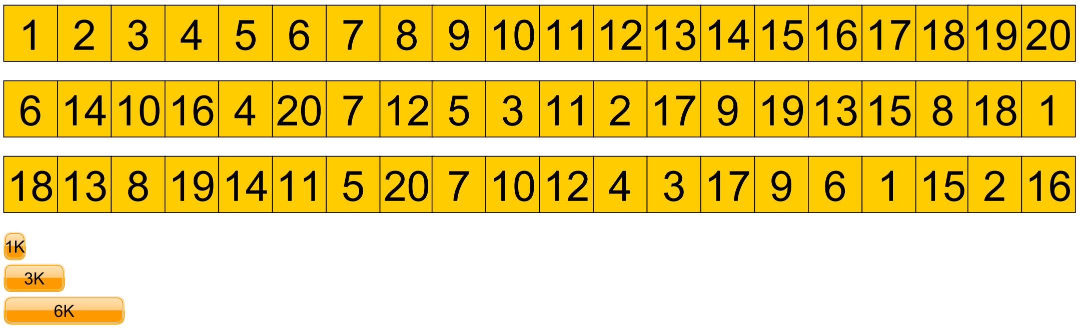

Data
Training Set
Development Set
Test Set
Step 4
Step 1
Train Model
Step 2 Optimize Model
Labeled Data (L)
Semi-Supervised Learning Algorithms
Step 5
Is U empty, or development set error rate increased?
Yes
Final Model
Unlabeled Data (U)
Step 3
Hypothesize Examples
No
Step 4
Move most confident examples with hypothesized labels to L
Go to Step 1
Performance Evaluation (F-measure, NIST)

Figure 6.3: Block Diagram Representation of the Experimental Setup



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
6	14	10	16	4	20	7	12	5	3	11	2	17	9	19	13	15	8	18	1
18	13	8	19	14	11	5	20	7	10	12	4	3	17	9	6	1	15	2	16

Figure 6.4: Block Diagram Representation of Three Different Random Orderings of the Training Set

different 1K initial manually labeled data for each ordering, which consists of a subset of record 1 in the first ordering, a subset of record 6 in the second ordering, and a subset of record 18 in the third ordering. Different initial manually labeled data sets for each ordering provides different baseline initial models at the beginning for each ordering. Therefore, in each ordering, different samples will be automatically selected from the unlabeled data set U and moved to labeled data set L, within the processing of semi-supervised learning algorithms. Hence, different experimental results will be obtained from three different orderings. In order to evaluate an average performance within three different orderings, development and test sets are kept same for consistency.

6.4 Experimental Results of the Self-Training and Co-Training Methods Based on Different Feature Sets

Figures 6.5 - 6.28 present average and extremum results using various semi-supervised strategies against the baseline based on various feature sets. The curves in Figures 6.5 - 6.16 illustrate average performance variation on using individual features (Lexical-only, Morphological-only and Prosodic-only) when only a small amount of labeled data is available. Moreover, Figures 6.17 - 6.28 presents average performance variation on using binary combinations of the lexical, morphological and prosodic features together. These figures show that not only semi-supervised methods improved the results of the baseline, but also those methods improved the results of the self-training.

6.4.1 Experimental Results Based on the Lexical Features

Tables 6.2, 6.3 and Figures 6.5 - 6.8 present improvement of various semi-supervised methods against the baseline. While analyzing those results, it is assumed that all features are available for the training set, but only lexical features are available on the test set. This means, while training multi-view models, morphological and prosodic views can still contribute in order to improve final binary classifier (M_{lex}).

Strategy	**F (%)**	**NIST (%)**	**Opt. Increment**	**Opt. Iteration**	**Opt. Data Size**
Manually Labeled Data Size = 1K					
Baseline(l)	26.31	91.38	0	0	0
Self-Training (l)	35.41	96.62	1166.67	21.33	26333.33
Agreement (l+m)	37.17	82.86	1500.00	22.00	34000.00
Agreement (l+p)	34.83	93.57	1333.33	12.00	16666.67
Self-Combined (l+m)	34.85	94.07	1333.33	15.67	14591.00
Self-Combined (l+p)	34.41	96.29	1000.00	15.33	15003.33
Disagreement (l+m)	43.81	81.21	1333.33	18.33	26833.33
Disagreement (l+p)	54.60	72.52	1333.33	25.00	34333.33

Continued on next page

Table 6.2 – *Continued from previous page*

Strategy	**F (%)**	**NIST (%)**	**Opt. Increment**	**Opt. Iteration**	**Opt. Data Size**
Strategy 1	34.71	95.78	1333.33	15.33	20833.33
Strategy 2	36.30	91.60	1333.33	19.34	26611.11
Strategy 3	35.91	97.93	1166.67	20.00	27835.33
Strategy 4	34.81	94.22	1277.78	21.11	25389.33
Strategy 5	37.84	93.09	1277.78	17.56	23055.56
Strategy 6	37.26	90.29	1027.78	16.67	17614.33
Strategy 7	33.77	92.94	777.78	18.11	15055.55
Strategy 8	35.88	88.21	1333.33	19.33	27166.67
Strategy 9	**59.30**	**64.97**	1000.00	25.00	26000
Manually Labeled Data Size = 3K					
Baseline (l)	42.90	92.14	0	0	0
Self-Training (l)	45.78	89.88	1000.00	17.33	20333.33
Agreement (l+m)	48.95	74.55	1000.00	24.33	27333.33
Agreement (l+p)	46.52	78.12	1333.33	22.33	32666.67
Self-Combined (l+m)	47.29	76.50	1333.33	21.00	27129.00
Self-Combined (l+p)	47.12	77.36	1500.00	20.33	31383.67
Disagreement (l+m)	56.12	68.26	1500.00	23.67	38500.00
Disagreement (l+p)	57.92	70.98	1500.00	24.33	39500.00
Strategy 1	46.93	77.47	1500.00	19.67	32500.00
Strategy 2	49.52	78.59	1333.33	22.22	33055.56
Strategy 3	47.18	77.10	1333.33	20.00	36190.33
Strategy 4	47.16	78.84	1333.33	20.67	29380.33
Strategy 5	53.35	73.58	1277.78	22.89	32666.67
Strategy 6	49.93	75.57	1388.89	21.00	30733.89
Strategy 7	46.74	78.42	1333.33	20.44	29833.33
Strategy 8	49.94	73.98	1500.00	25.00	40500.00
Strategy 9	**64.57**	**58.91**	1500.00	25.00	40500
Manually Labeled Data Size = 6K					
Baseline (l)	54.41	73.02	0	0	0
Self-Training (l)	55.09	74.09	1166.67	17.33	25166.67
Agreement (l+m)	56.83	68.40	1333.33	21.33	34166.67
Agreement (l+p)	56.86	70.36	1166.67	24.00	33666.67
Self-Combined (l+m)	56.83	68.38	1500.00	22.33	35647.33

Continued on next page

Table 6.2 – *Continued from previous page*

Strategy	**F (%)**	**NIST (%)**	**Opt. Increment**	**Opt. Iteration**	**Opt. Data Size**
Self-Combined (l+p)	55.63	73.31	1083.33	22.33	27898.00
Disagreement (l+m)	62.45	61.52	1500.00	21.00	37500.00
Disagreement (l+p)	64.08	62.86	1500.00	24.00	42000.00
Strategy 1	56.93	70.14	1500.00	18.33	33500.00
Strategy 2	58.96	66.55	1444.44	22.56	38833.33
Strategy 3	57.12	69.24	1166.67	23.33	40452.67
Strategy 4	56.72	68.82	1277.78	20.56	31091.78
Strategy 5	59.77	64.61	1388.89	24.00	39555.56
Strategy 6	58.91	65.23	1500.00	23.67	39609.56
Strategy 7	56.84	69.05	1444.44	20.00	34611.11
Strategy 8	57.92	68.12	1500.00	22.33	39500.00
Strategy 9	**66.15**	**57.12**	1500.00	24.00	42000

TABLE 6.2: Average Results of the Different Strategies for the Lexical Features Only

Man. Labeled Data = 1K	**Strategy**	**F (%)**	**NIST (%)**
Baseline	-	26.31	91.38
Self-Training	-	35.41	96.62
Co-Training (2-View)	Disagreement (l+m)	43.81	81.21
Co-Training (2-View)	Disagreement (l+p)	54.60	72.52
Committee-Based (3-View)	Strategy 9 (l+m+p)	**59.30**	**64.97**
Man. Labeled Data = 3K	**Strategy**	**F (%)**	**NIST (%)**
Baseline	-	42.90	92.14
Self-Training	-	45.78	89.88
Co-Training (2-View)	Disagreement (l+m)	56.12	68.26
Co-Training (2-View)	Disagreement (l+p)	57.92	70.98
Committee-Based (3-View)	Strategy 9 (l+m+p)	**64.57**	**58.91**
Man. Labeled Data = 6K	**Strategy**	**F (%)**	**NIST (%)**
Baseline	-	54.41	73.02
Self-Training	-	55.09	74.09
Co-Training (2-View)	Disagreement (l+m)	62.45	61.52
Co-Training (2-View)	Disagreement (l+p)	64.08	62.86
Committee-Based (3-View)	Strategy 9 (l+m+p)	**66.15**	**57.12**

Table 6.3: Maximum F-measure Scores and Minimum NIST Error Rates of the Different Strategies for the Lexical Features Only

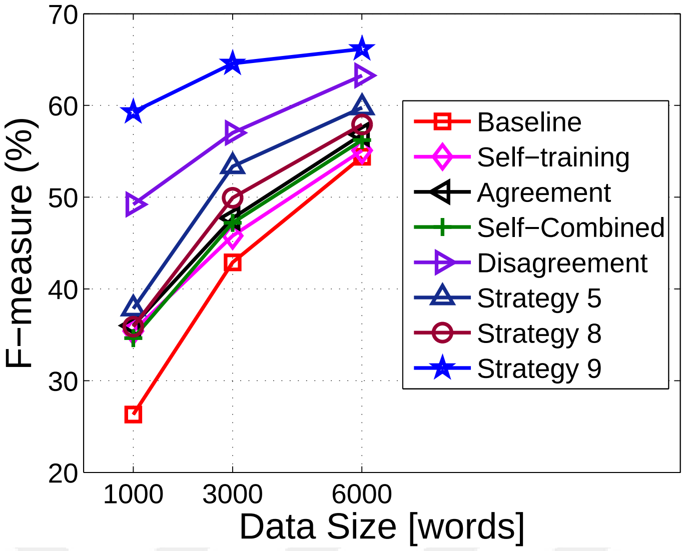


Figure 6.5: Average F-measure Scores of Different Strategies for the Lexical Features Only

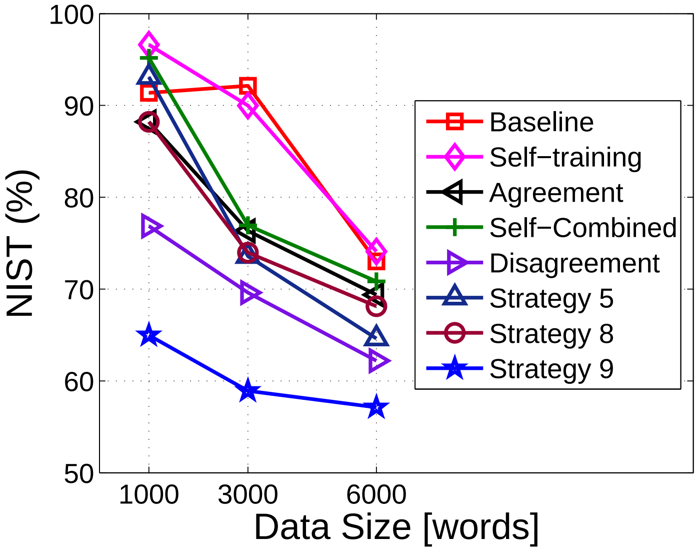


Figure 6.6: Average NIST Error Rates of Different Strategies for the Lexical Features Only

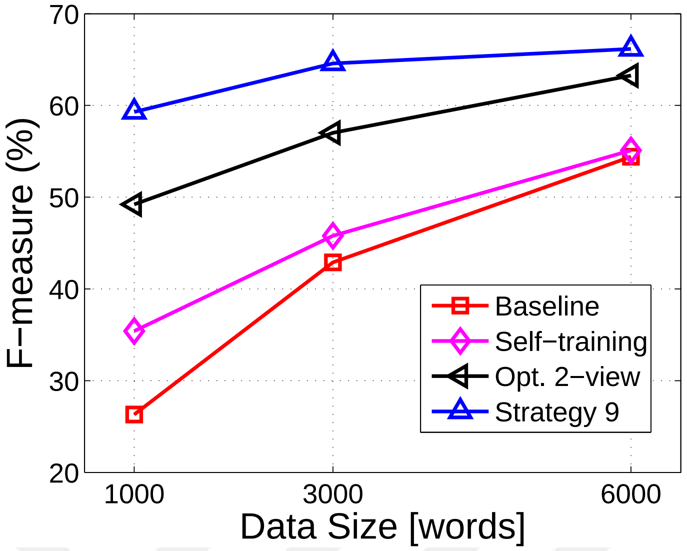


Figure 6.7: Maximum F-measure Scores of Different Strategies for the Lexical Features Only

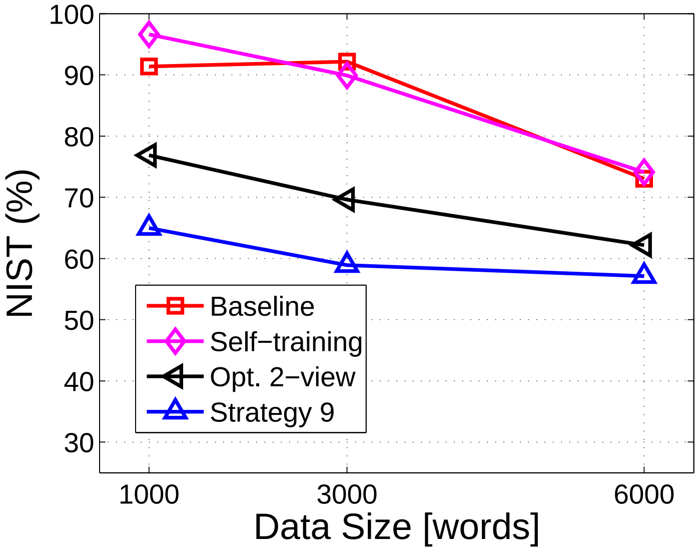


Figure 6.8: Minimum NIST Error Rates of Different Strategies for the Lexical Features Only

Figures 6.5 - 6.6 and Table 6.2 presents experimental results of Baseline, Self-training, different strategies of two-view and three-view co-training strategies. Figures 6.7 - 6.8 and Table 6.3 compares Baseline and Self-training results to best 2-view results and Strategy 9 (Best 3-view strategy for this case) when different sizes of manually labeled data are available. According to these results, when only 1000 manually labeled examples are available, Strategy 9 improved F-measure scores of Baseline, Self-training, Co-Training (l+m), Co-Training (l+p) with a percentage improvement of 125.3896%, 67.4668%, 35.3572%, 8.6081% respectively. In addition, when only 3000 manually labeled examples are available, Strategy 9 improved F-measure scores of Baseline, Self-training, Co-Training (l+m), Co-Training (l+p) with a percentage improvement of 50.5128%, 41.0441%, 15.0570%, 11.4814% respectively, and finally when only 6000 manually labeled examples are available, Strategy 9 improved F-measure scores of Baseline, Self-training, Co-Training (l+m), Co-Training (l+p) with a percentage improvement of 21.5769%, 20.0762%, 5.9247%, 3.2303% respectively.

6.4.2 Experiment Results Based on the Morphological Features

Tables 6.4, 6.5 and Figures 6.9 - 6.12 present improvement of various semi-supervised methods against the baseline. While analyzing those results, it is assumed that all features are available for the training set, but only morphological features are available on the test set. This means, while training multi-view models, lexical and prosodic views can still contribute in order to improve final binary classifier (M_{morp}).

Strategy	**F (%)**	**NIST (%)**	**Opt. Increment**	**Opt. Iteration**	**Opt. Data Size**
Manually Labeled Data Size = 1K					
Baseline (m)	75.22	49.81	0	0	0
Self-Training (m)	75.52	48.69	1033.33	18.00	22866.67
Agreement (m+l)	75.69	48.76	1000.00	19.67	21666.67
Agreement (m+p)	75.74	48.79	533.33	18.00	11500.00

Continued on next page

Table 6.4 – *Continued from previous page*

Strategy	**F (%)**	**NIST (%)**	**Opt. Increment**	**Opt. Iteration**	**Opt. Data Size**
Self-Combined (m+l)	75.10	49.60	233.33	16.33	4358.00
Self-Combined (m+p)	**75.90**	**48.34**	700.00	23.33	16838.00
Disagreement (m+l)	75.47	49.10	1083.33	19.67	21333.33
Disagreement (m+p)	75.59	48.81	450.00	8.00	7050.00
Strategy 1	75.77	48.55	1500.00	16.67	26000.00
Strategy 2	75.52	49.27	716.67	7.11	8250.00
Strategy 3	75.88	48.67	916.67	11.33	11131.33
Strategy 4	75.69	49.04	966.67	13.78	13044.78
Strategy 5	75.37	49.40	488.89	9.56	7050.00
Strategy 6	75.52	49.34	483.33	5.11	3337.11
Strategy 7	75.70	48.92	1166.67	18.44	22722.22
Strategy 8	75.73	48.90	916.66	9.00	8916.66
Strategy 9	75.77	48.62	700.42	19.22	12767
Manually Labeled Data Size = 3K					
Baseline (m)	75.51	48.74	0	0	0
Self-Training (m)	75.75	49.05	533.33	13.00	13000.00
Agreement (m+l)	75.85	49.19	1033.33	8.67	14133.33
Agreement (m+p)	76.05	47.90	866.67	11.00	14533.33
Self-Combined (m+l)	75.64	49.71	450.00	10.67	5167.67
Self-Combined (m+p)	76.18	47.52	1083.33	18.33	21906.00
Disagreement (m+l)	75.62	49.67	333.33	4.67	5083.33
Disagreement (m+p)	76.45	47.07	1083.33	13.33	19250.00
Strategy 1	76.20	47.67	866.67	13.67	19933.33
Strategy 2	76.21	47.07	811.11	16.67	14366.67
Strategy 3	75.88	48.43	616.67	16.00	11680.00
Strategy 4	76.25	48.03	1011.11	12.67	13751.56
Strategy 5	76.37	46.83	666.67	11.34	13027.78
Strategy 6	76.21	47.66	983.33	12.11	14541.00
Strategy 7	76.11	48.10	800.00	15.78	16488.89
Strategy 8	76.26	48.12	916.66	19.00	18250.00
Strategy 9	**78.14**	**43.76**	1333.33	23.33	33833
Manually Labeled Data Size = 6K					
Baseline (m)	76.30	47.43	0	0	0

Continued on next page

Table 6.4 – *Continued from previous page*

Strategy	**F (%)**	**NIST (%)**	**Opt. Increment**	**Opt. Iteration**	**Opt. Data Size**
Self-Training (m)	76.76	47.05	700.00	13.00	19700.00
Agreement (m+l)	76.70	47.14	1033.33	10.67	19666.67
Agreement (m+p)	77.39	44.31	1166.67	20.67	30833.33
Self-Combined (m+l)	76.92	46.86	616.67	6.33	8419.67
Self-Combined (m+p)	77.26	45.69	1083.33	17.00	25915.33
Disagreement (m+l)	76.69	47.62	283.33	16.00	11500.00
Disagreement (m+p)	77.55	45.02	1000.00	22.00	28000.00
Strategy 1	77.10	45.36	1333.33	22.00	35333.33
Strategy 2	77.70	44.61	1222.22	18.22	29666.66
Strategy 3	77.12	45.69	1166.67	17.67	36104.33
Strategy 4	77.00	46.12	1388.89	15.22	25639.44
Strategy 5	77.66	44.38	1166.66	21.11	31666.67
Strategy 6	77.14	45.59	844.44	18.22	20920.11
Strategy 7	77.41	44.75	1444.44	22.11	37833.33
Strategy 8	77.27	44.78	1333.33	16.00	26333.33
Strategy 9	**78.44**	**42.90**	1333.33	22.11	36833

Table 6.4: Average Results of the Different Strategies for the Morphological Features Only

Figures 6.9 - 6.10 and Table 6.4 presents experimental results of Baseline, Self-training, different strategies of two-view and three-view co-training strategies. Figures 6.11 - 6.12 and Table 6.5 compares Baseline and Self-training results to best 2-view and 3-view results included with Strategy 9, when different sizes of manually labeled data are available. According to these results, when only 1000 manually labeled examples are available, Strategy 4 with l-m-p view order improved F-measure scores of Baseline, Self-training, 2-view Agreement (l+m), 2-view Self-Combined (m+p) with a percentage improvement of 0.9173%, 0.5164%, 0.2907%, 0.0132% respectively. In addition, when only 3000 manually labeled examples are available, Strategy 9 improved F-measure scores of those strategies with a percentage improvement of 3.4830%, 3.1551%, 3.0191%, 2.2106% respectively, and finally when only 6000 manually labeled examples are available, Strategy 9 improved F-measure scores of those strategies with a percentage improvement of 2.8047%, 2.1886%, 1.9761%, 1.1476% respectively.

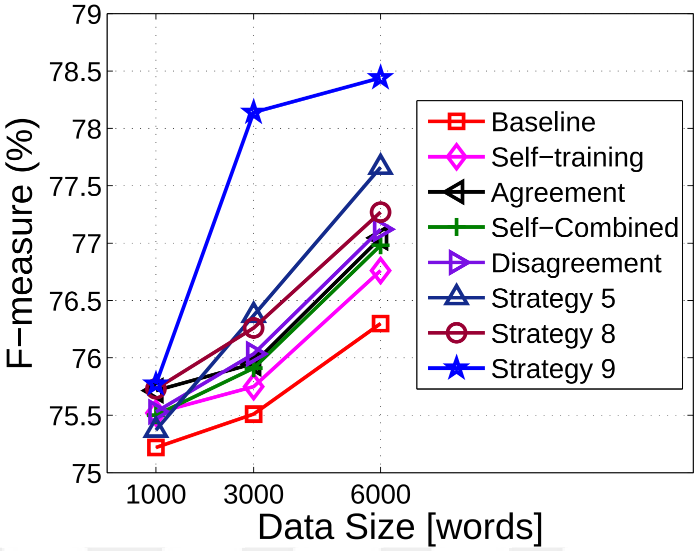


Figure 6.9: Average F-measure Scores of Different Strategies for the Morphological Features Only

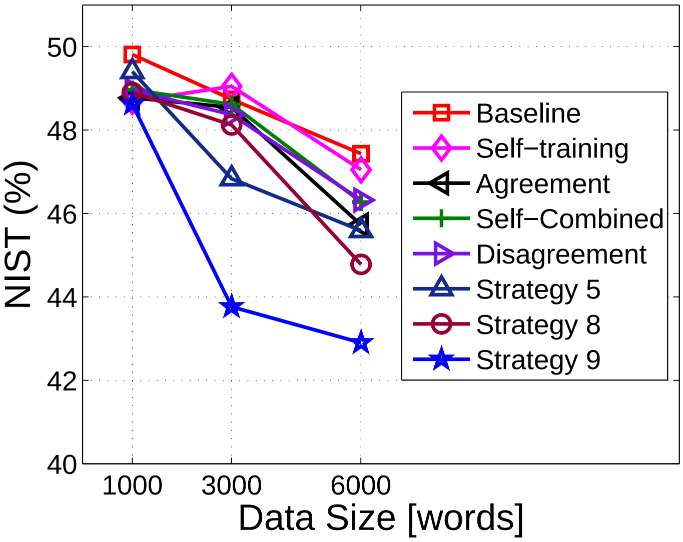


Figure 6.10: Average NIST Error Rates of Different Strategies for the Morphological Features Only

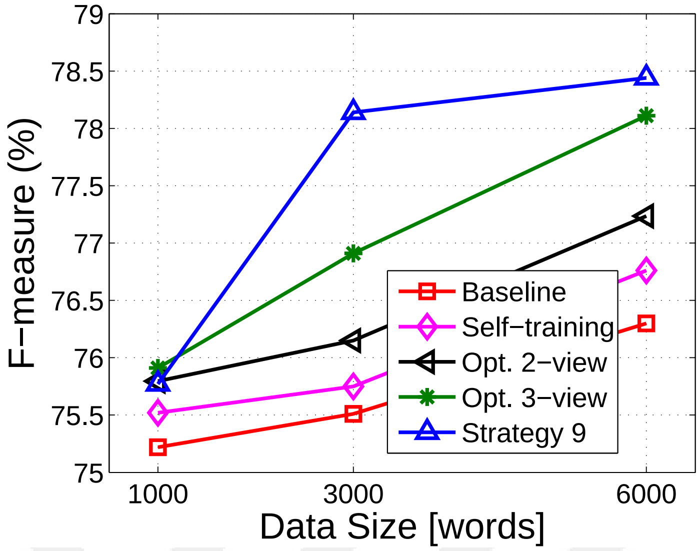


Figure 6.11: Maximum F-measure Scores of Different Strategies for the Morphological Features Only

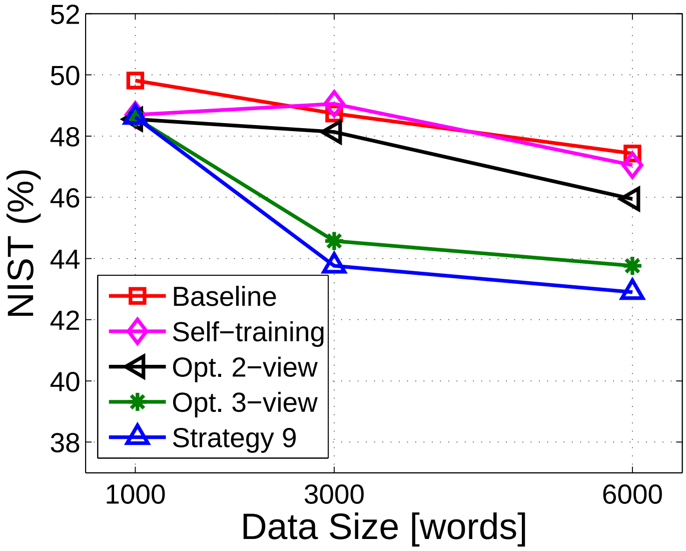


Figure 6.12: Minimum NIST Error Rates of Different Strategies for the Morphological Features Only

Man. Labeled Data = 1K	**Strategy**	**F (%)**	**NIST (%)**
Baseline	-	75.22	49.81
Self-Training	-	75.52	48.69
Co-Training (2-View)	Agreement (m+l)	75.69	48.76
Co-Training (2-View)	Self-Combined (m+p)	75.90	**48.34**
Co-Training (3-View)	Strategy 4 (l-m-p)	**75.91**	48.60
Committee-Based (3-View)	Strategy 9 (l+m+p)	75.77	48.62
Man. Labeled Data = 3K	**Strategy**	**F (%)**	**NIST (%)**
Baseline	-	75.51	48.74
Self-Training	-	75.75	49.05
Co-Training (2-View)	Agreement (m+l)	75.85	49.19
Co-Training (2-View)	Disagreement (m+p)	76.45	47.07
Co-Training (3-View)	Strategy 5 (l-p-m)	76.91	44.57
Committee-Based (3-View)	Strategy 9 (l+m+p)	**78.14**	**43.76**
Man. Labeled Data = 6K	**Strategy**	**F (%)**	**NIST (%)**
Baseline	-	76.30	47.43
Self-Training	-	76.76	47.05
Co-Training (2-View)	Self-Combined (m+l)	76.92	46.86
Co-Training (2-View)	Disagreement (m+p)	77.55	45.02
Co-Training (3-View)	Strategy 2 (l-p-m)	78.11	43.76
Co-Training (3-View)	Strategy 5 (l-p-m)	77.91	43.69
Committee-Based (3-View)	Strategy 9 (l+m+p)	**78.44**	**42.90**

Table 6.5: Maximum F-measure Scores and Minimum NIST Error Rates of the Different Strategies for the Morphological Features Only

6.4.3 Experimental Results Based on the Prosodic Features

Tables 6.6, 6.7 and Figures 6.13 - 6.16 present improvement of various semi-supervised methods against the baseline. While analyzing those results, it is assumed that all features are available for the training set, but only prosodic features are available on the test set. This means, while training multi-view models, lexical and morphological views can still contribute in order to improve final binary classifier (M_{pros}).

Strategy	**F (%)**	**NIST (%)**	**Opt. Increment**	**Opt. Iteration**	**Opt. Data Size**
Manually Labeled Data Size = 1K					
Baseline (p)	72.19	52.00	0	0	0
Self-Training (p)	72.85	50.33	616.67	13.33	13450.00
Agreement (p+l)	72.61	51.28	150.00	8.33	2083.33
Agreement (p+m)	73.32	49.55	1166.67	13.67	20166.67

Continued on next page

Table 6.6 – *Continued from previous page*

Strategy	**F (%)**	**NIST (%)**	**Opt. Increment**	**Opt. Iteration**	**Opt. Data Size**
Self-Combined (p+l)	72.96	50.90	200.00	14.33	3628.00
Self-Combined (p+m)	72.51	50.83	583.33	16.33	9943.33
Disagreement (p+l)	72.16	50.78	400.00	9.00	4600.00
Disagreement (p+m)	71.67	52.41	583.33	4.67	3583.33
Strategy 1	72.82	51.21	400.00	11.00	7200.00
Strategy 2	72.11	51.96	611.11	7.11	5694.44
Strategy 3	72.78	50.26	400.00	10.33	5194.33
Strategy 4	72.93	50.40	505.56	9.45	3991.89
Strategy 5	72.31	51.78	616.67	11.56	10127.78
Strategy 6	72.77	50.74	377.78	16.00	6109.78
Strategy 7	72.66	50.65	638.89	12.22	8805.56
Strategy 8	72.80	50.73	100.00	6.33	1633.33
Strategy 9	**74.20**	**47.88**	1000.00	18.13	19667
Manually Labeled Data Size = 3K					
Baseline (p)	71.75	55.00	0	0	0
Self-Training (p)	72.63	52.05	700.00	8.67	8966.67
Agreement (p+l)	72.70	51.55	1166.67	8.67	11833.33
Agreement (p+m)	72.90	50.91	1083.33	8.33	10500.00
Self-Combined (p+l)	73.24	49.88	833.33	8.33	9131.00
Self-Combined (p+m)	**73.33**	**48.71**	700.00	15.33	15859.33
Disagreement (p+l)	72.72	51.29	616.67	6.67	4583.33
Disagreement (p+m)	72.85	51.41	283.33	16.33	8416.67
Strategy 1	72.65	51.43	666.67	19.67	17916.67
Strategy 2	72.90	51.19	461.11	8.56	7722.22
Strategy 3	72.90	52.05	1166.67	10.67	20998.00
Strategy 4	72.97	50.44	761.11	16.11	13499.67
Strategy 5	72.10	52.10	855.56	11.89	11388.89
Strategy 6	72.88	50.47	816.67	10.33	11649.11
Strategy 7	72.52	51.77	827.78	15.78	15888.89
Strategy 8	72.65	50.93	250.00	14.00	6500.00
Strategy 9	72.71	50.38	1803.02	14.33	21833
Manually Labeled Data Size = 6K					
Baseline (p)	70.31	55.79	0	0	0

Continued on next page

Table 6.6 – *Continued from previous page*

Strategy	**F (%)**	**NIST (%)**	**Opt. Increment**	**Opt. Iteration**	**Opt. Data Size**
Self-Training (p)	71.68	52.02	700.00	13.00	12100.00
Agreement (p+l)	71.84	51.50	1166.67	3.67	10166.67
Agreement (p+m)	71.86	51.95	866.67	9.00	10133.33
Self-Combined (p+l)	72.20	51.95	1083.33	10.33	16160.33
Self-Combined (p+m)	**72.59**	52.43	1166.67	16.00	21312.67
Disagreement (p+l)	71.54	52.88	1033.33	14.33	16766.67
Disagreement (p+m)	72.18	52.48	916.67	9.33	17416.67
Strategy 1	71.75	51.26	533.33	13.33	10766.67
Strategy 2	71.77	52.28	694.45	11.89	13416.67
Strategy 3	72.26	**51.02**	1333.33	14.67	33103.67
Strategy 4	72.58	51.52	455.55	16.00	12445.78
Strategy 5	71.93	52.97	1011.11	9.56	14555.56
Strategy 6	72.32	51.64	472.22	15.56	11390.11
Strategy 7	72.07	51.71	522.22	11.89	11400.00
Strategy 8	71.74	52.33	1083.33	8.33	16000.00
Strategy 9	72.29	51.21	1166.67	17.12	25333

Table 6.6: Average Results of the Different Strategies for the Prosodic Features Only

Figures 6.13 - 6.14 and Table 6.6 presents experimental results of Baseline, Self-training, different strategies of two-view and three-view co-training strategies. Figures 6.15 - 6.16 and Table 6.7 compares Baseline and Self-training results to best 2-view and 3-view results included with Strategy 9, when different sizes of manually labeled data are available. According to these results, when only 1000 manually labeled examples are available, Strategy 4 with l-m-p view order improved F-measure scores of Baseline, Self-training, 2-view Self-Combined (p+l), 2-view Agreement (m+p) with a percentage improvement of 2.7843%, 1.8531%, 1.6996%, 1.2002% respectively. When only 3000 manually labeled examples are available, 2-view Co-Training with Self-Combined Strategy (p+m) improved F-measure scores of Baseline and Self-Training with a percentage improvement of

2.2021%, 0.9638% respectively, and finally when only 6000 manually labeled examples are available, Strategy 4 (l-p-m) improved F-measure scores of Baseline, Self-Training and 2-view Co-Training with Self-Combined strategy (p+m) with a percentage improvement of 3.7264%, 1.7439%, 0.4684% respectively.

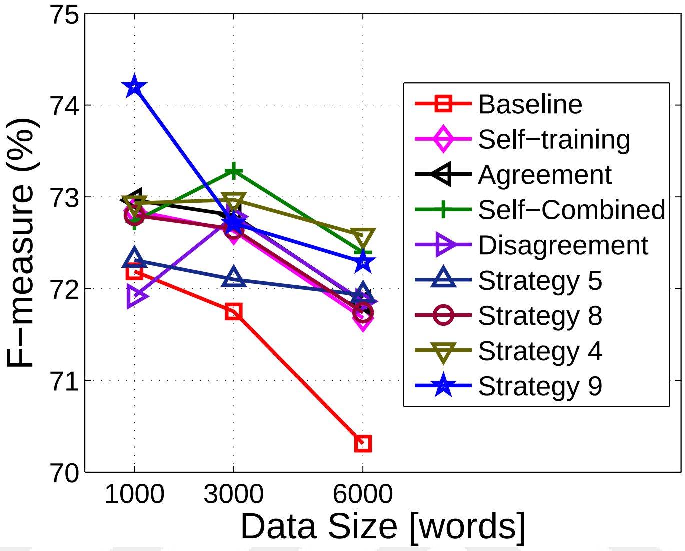


Figure 6.13: Average F-measure Scores of Different Strategies for the Prosodic Features Only

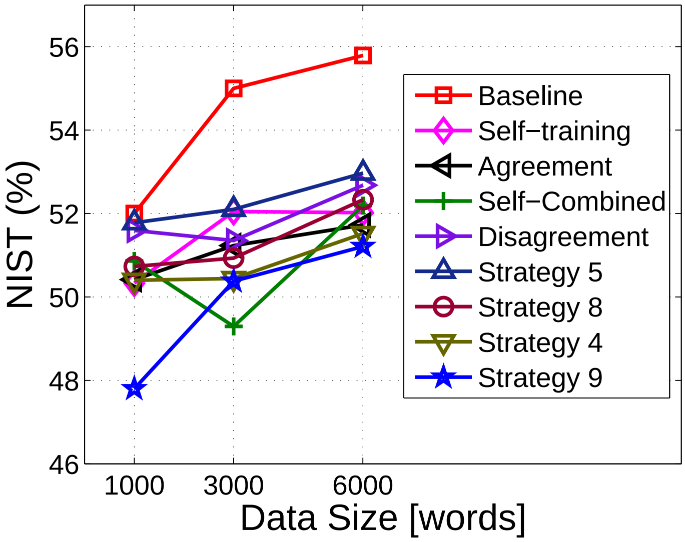


Figure 6.14: Average NIST Error Rates of Different Strategies for the Prosodic Features Only

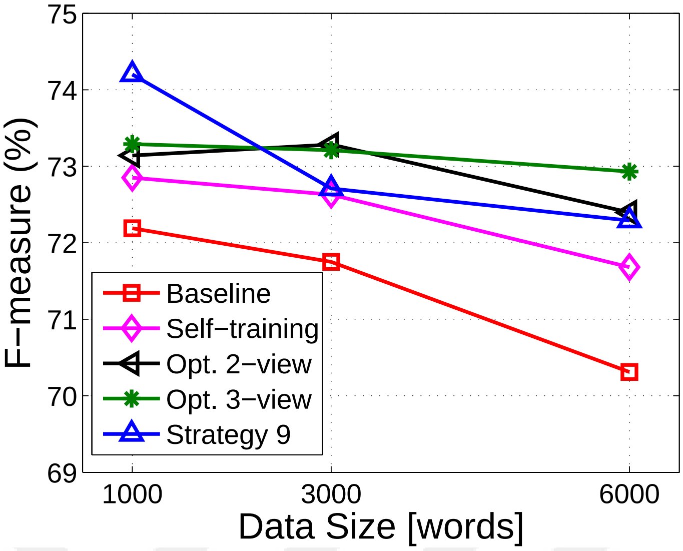


Figure 6.15: Maximum F-measure Scores of Different Strategies for the Prosodic Features Only

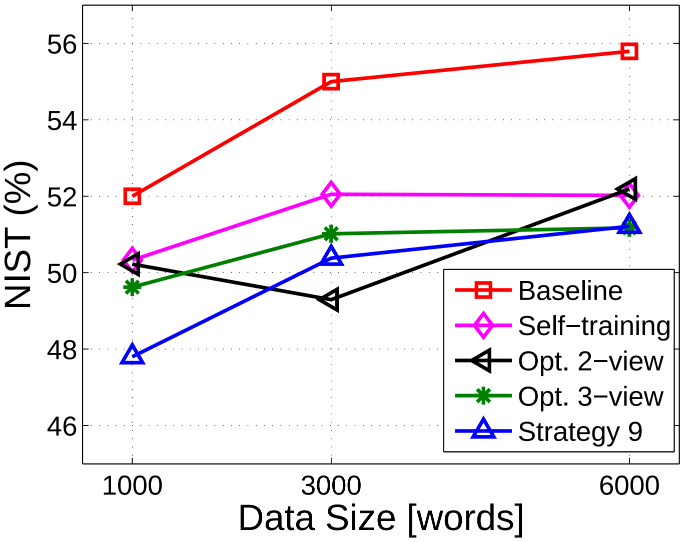


Figure 6.16: Minimum NIST Error Rates of Different Strategies for the Prosodic Features Only

Man. Labeled Data = 1K	**Strategy**	**F (%)**	**NIST (%)**
Baseline	-	72.19	52.00
Self-Training	-	72.85	50.33
Co-Training (2-View)	Self-Combined (p+l)	72.96	50.90
Co-Training (2-View)	Agreement (p+m)	73.32	49.55
Co-Training (3-View)	Strategy 6 (l-m-p)	73.29	49.62
Co-Training (3-View)	Strategy 4 (l-m-p)	73.19	49.09
Committee-Based (3-View)	Strategy 9 (l+m+p)	**74.20**	**47.80**
Man. Labeled Data = 3K	**Strategy**	**F (%)**	**NIST (%)**
Baseline	-	71.75	55.00
Self-Training	-	72.63	52.05
Co-Training (2-View)	Self-Combined (p+l)	73.24	49.88
Co-Training (2-View)	Self-Combined (p+m)	**73.33**	**48.71**
Co-Training (3-View)	Strategy 7 (p-m-l)	73.21	51.02
Co-Training (3-View)	Strategy 4 (l-m-p)	73.15	49.66
Committee-Based (3-View)	Strategy 9 (l+m+p)	72.71	50.38
Man. Labeled Data = 6K	**Strategy**	**F (%)**	**NIST (%)**
Baseline	-	70.31	55.79
Self-Training	-	71.68	52.02
Co-Training (2-View)	Self-Combined (p+m)	72.59	52.43
Co-Training (2-View)	Self-Combined (p+l)	72.20	51.95
Co-Training (3-View)	Strategy 4 (l-p-m)	**72.93**	51.17
Co-Training (3-View)	Strategy 7 (l-p-m)	72.44	**50.88**
Committee-Based (3-View)	Strategy 9 (l+m+p)	72.29	51.21

Table 6.7: Maximum F-measure Scores and Minimum NIST Error Rates of the Different Strategies for the Prosodic Features Only

6.4.4 Experimental Results Based on the Combination of Lexical and Morphological Features

Tables 6.8, 6.9 and Figures 6.17 - 6.20 present improvement of various semi-supervised methods against the baseline. While analyzing those results, it is assumed that all features are available for the training set, but only lexical and morphological features are available on the test set. This means, while training multi-view models, prosodic view can still contribute in order to improve final binary classifier ($M_{lex+morp}$).

Strategy	**F (%)**	**NIST (%)**	**Opt. Increment**	**Opt. Iteration**	**Opt. Data Size**
Manually Labeled Data Size = 1K					
Baseline (l+m)	75.37	48.86	0	0	0

Continued on next page

Table 6.8 – *Continued from previous page*

Strategy	**F (%)**	**NIST (%)**	**Opt. Increment**	**Opt. Iteration**	**Opt. Data Size**
Self-Training (l+m)	75.40	49.07	533.33	14.33	8233.33
Agreement (l+m)	75.58	49.09	1033.33	18.67	17800.00
Self-Combined (l+m)	75.59	48.95	700.00	14.67	4884.00
Disagreement (l+m)	75.47	48.86	833.33	10.00	6666.67
Strategy 1	75.57	48.74	1033.33	12.33	17166.67
Strategy 2	75.58	49.13	800.00	11.89	9888.89
Strategy 3	75.88	48.40	700.00	4.67	5242.33
Strategy 4	75.50	49.04	1166.67	17.55	19897.89
Strategy 5	75.67	49.18	861.11	15.33	15194.45
Strategy 6	75.33	49.28	511.11	15.11	9250.44
Strategy 7	75.59	48.89	1083.33	17.34	20361.11
Strategy 8	75.63	48.90	833.33	18.00	16166.67
Strategy 9	**76.03**	**48.00**	666.67	18.33	12000
Manually Labeled Data Size = 3K					
Baseline (l+m)	76.41	48.55	0	0	0
Self-Training (l+m)	76.38	47.98	866.67	8.33	11933.33
Agreement (l+m)	76.39	47.95	1166.67	9.67	13666.67
Self-Combined (l+m)	76.13	49.14	233.33	13.00	7015.00
Disagreement (l+m)	76.53	47.26	916.67	10.67	10083.33
Strategy 1	76.61	48.00	833.33	15.00	17833.33
Strategy 2	76.58	47.72	927.78	11.44	14055.55
Strategy 3	76.28	48.21	1500.00	11.33	25439.67
Strategy 4	76.64	47.67	1333.33	20.56	29269.11
Strategy 5	76.75	47.01	788.89	14.56	16166.66
Strategy 6	76.65	46.87	1083.33	17.11	22037.67
Strategy 7	76.52	47.76	1122.22	19.44	25755.56
Strategy 8	76.78	46.21	1333.33	13.66	20500.00
Strategy 9	**78.45**	**43.76**	1333.33	25.41	35833
Manually Labeled Data Size = 6K					
Baseline (l+m)	77.14	45.48	0	0	0
Self-Training (l+m)	77.27	46.88	1333.33	12.00	21666.67
Agreement (l+m)	77.59	45.07	1333.33	16.67	27666.67
Self-Combined (l+m)	77.10	47.40	700.00	19.33	15938.67

Continued on next page

Table 6.8 – *Continued from previous page*

Strategy	**F (%)**	**NIST (%)**	**Opt. Increment**	**Opt. Iteration**	**Opt. Data Size**
Disagreement (l+m)	77.37	46.07	1000.00	17.67	26000.00
Strategy 1	77.72	44.74	1333.33	21.00	34500.00
Strategy 2	78.32	43.48	1277.78	20.78	32777.78
Strategy 3	77.79	44.57	1166.67	19.00	33526.00
Strategy 4	77.58	46.19	1277.78	18.22	28932.89
Strategy 5	78.41	42.82	1388.89	20.11	33833.33
Strategy 6	78.30	43.59	1333.33	20.56	32923.67
Strategy 7	78.12	43.38	1500.00	23.55	41333.33
Strategy 8	78.38	42.78	1333.33	24.00	38000.00
Strategy 9	**79.36**	**41.09**	1333.33	23.33	36833

TABLE 6.8: Average Results of the Different Strategies for the Combination of Morphological and Lexical Features

Figures 6.17 - 6.18 and Table 6.8 presents experimental results of Baseline, Self-training, different strategies of two-view and three-view co-training strategies.

Man. Labeled Data = 1K	**Strategy**	**F (%)**	**NIST (%)**
Baseline	-	75.37	48.86
Self-Training	-	75.40	49.07
Co-Training (2-View)	Self-Combined (l+m)	75.59	48.95
Co-Training (2-View)	Disagreement (l+m)	75.47	48.86
Co-Training (3-View)	Strategy 5 (p-m-l)	75.97	48.14
Committee-Based (3-View)	Strategy 9 (l+m+p)	**76.03**	**48.00**
Man. Labeled Data = 3K	**Strategy**	**F (%)**	**NIST (%)**
Baseline	-	76.41	48.55
Self-Training	-	76.38	47.98
Co-Training (2-View)	Disagreement (l+m)	76.53	47.26
Co-Training (3-View)	Strategy 5 (l-m-p)	77.07	46.21
Committee-Based (3-View)	Strategy 9 (l+m+p)	**78.45**	**43.76**
Man. Labeled Data = 6K	**Strategy**	**F (%)**	**NIST (%)**
Baseline	-	77.14	45.48
Self-Training	-	77.27	46.88
Co-Training (2-View)	Agreement (l+m)	77.59	45.07
Co-Training (3-View)	Strategy 6 (l-m-p)	78.76	41.95
Committee-Based (3-View)	Strategy 9 (l+m+p)	**79.36**	**41.09**

Table 6.9: Maximum F-measure Scores and Minimum NIST Error Rates of the Different Strategies for the Combination of Morphological and Lexical Features

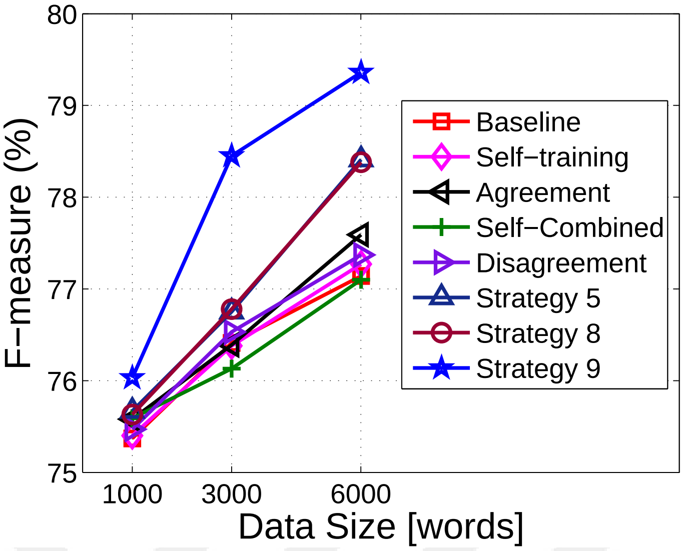


Figure 6.17: Average F-measure Scores of Different Strategies for the Combination of Morphological and Lexical Features

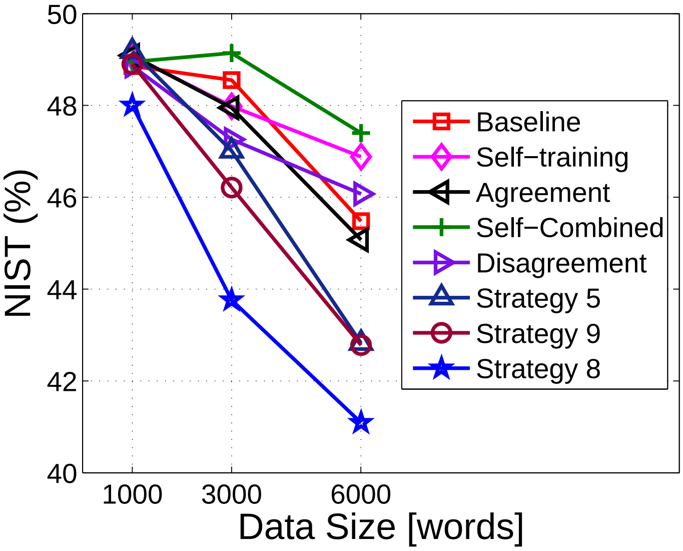


Figure 6.18: Average NIST Error Rates of Different Strategies for the Combination of Morphological and Lexical Features

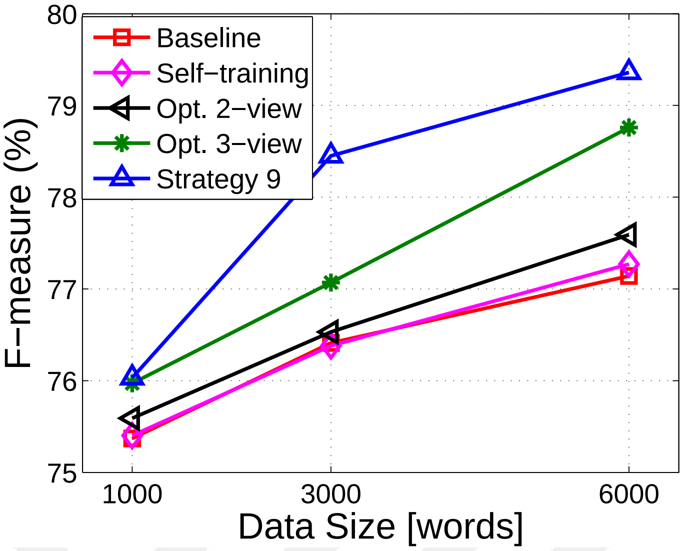


Figure 6.19: Maximum F-measure Scores of Different Strategies for the Combination of Morphological and Lexical Features

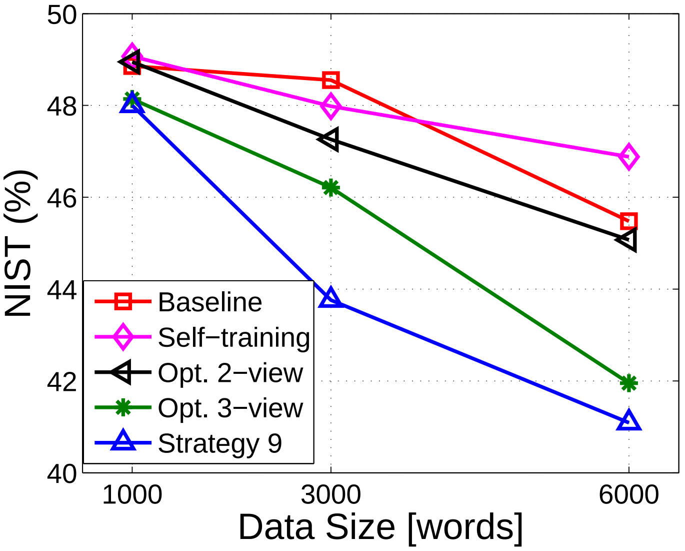


Figure 6.20: Minimum NIST Error Rates of Different Strategies for the Combination of Morphological and Lexical Features

Figures 6.19 - 6.20 and Table 6.9 compares Baseline and Self-training results to best 2-view and 3-view results included with Strategy 9, when different sizes of manually labeled data are available. According to these results, when only 1000 manually labeled examples are available, Strategy 9 improved F-measure scores of Baseline, Self-training, 2-view Self-Combined (m+l), 2-view Disagreement (m+l) and Strategy 5 (p-m-l) with a percentage improvement of 0.8757%, 0.8355%, 0.5821%, 0.7420%, 0.0790% respectively. When only 3000 manually labeled examples are available, Strategy 9 improved F-measure scores of Baseline, Self-Training, 2-view Disagreement (l+m) and Strategy 5 (l-m-p) with a percentage improvement of 2.6698%, 2.7101%, 2.5088%, 1.7906% respectively, and finally when only 6000 manually labeled examples are available, Strategy 9 improved F-measure scores of Baseline, Self-Training and 2-view Agreement (l-m) and Strategy 6 (l-m-p) with a percentage improvement of 2.8779%, 2.7048%, 2.2812%, 0.7618% respectively.

6.4.5 Experimental Results Based on the Combination of Lexical and Prosodic Features

Tables 6.10, 6.11 and Figures 6.21 - 6.24 present improvement of various semi-supervised methods against the baseline. While analyzing those results, it is assumed that all features are available for the training set, but only lexical and prosodic features are available on the test set. This means, while training multi-view models, morphological view can still contribute in order to improve final binary classifier ($M_{lex+pros}$).

Strategy	**F (%)**	**NIST (%)**	**Opt. Increment**	**Opt. Iteration**	**Opt. Data Size**
Manually Labeled Data Size = 1K					
Baseline (l+p)	73.20	50.38	0	0	0
Self-Training (l+p)	73.94	48.81	366.67	8.33	3833.33
Agreement (l+p)	73.54	49.17	1166.67	9.67	14166.67
Self-Combined (l+p)	74.44	49.19	500.00	13.00	7358.67

Continued on next page

Table 6.10 – *Continued from previous page*

Strategy	**F (%)**	**NIST (%)**	**Opt. Increment**	**Opt. Iteration**	**Opt. Data Size**
Disagreement (l+p)	73.38	50.74	1166.67	7.33	9000.00
Strategy 1	73.90	48.62	1333.33	10.00	15833.33
Strategy 2	73.89	48.59	744.45	13.22	10977.78
Strategy 3	74.22	48.16	366.67	15.00	6622.00
Strategy 4	74.17	48.41	327.78	18.78	6690.78
Strategy 5	73.44	49.30	605.56	7.78	5272.22
Strategy 6	74.25	48.64	666.67	10.22	4983.67
Strategy 7	74.01	48.50	1055.56	11.67	14222.22
Strategy 8	74.20	48.50	916.66	15.33	11583.33
Strategy 9	**75.57**	**46.50**	833.33	15.67	14167
Manually Labeled Data Size = 3K					
Baseline (l+p)	72.65	49.33	0	0	0
Self-Training (l+p)	74.40	47.14	700.00	9.33	12033.33
Agreement (l+p)	74.10	46.69	833.33	6.00	7500.00
Self-Combined (l+p)	74.62	47.38	700.00	14.00	14241.33
Disagreement (l+p)	74.23	48.45	450.00	4.67	6366.67
Strategy 1	73.86	48.26	616.67	9.67	14950.00
Strategy 2	74.02	47.47	644.45	8.89	9811.11
Strategy 3	**75.42**	**44.86**	1166.67	16.67	28852.33
Strategy 4	75.20	46.41	1122.22	19.78	25537.67
Strategy 5	74.38	47.03	855.56	12.11	15611.11
Strategy 6	74.42	47.72	877.78	17.33	17913.44
Strategy 7	74.17	48.00	522.22	7.34	5716.67
Strategy 8	73.57	48.88	450.00	7.66	9116.66
Strategy 9	74.71	46.74	1166.70	20.67	27667
Manually Labeled Data Size = 6K					
Baseline (l+p)	75.21	45.00	0	0	0
Self-Training (l+p)	76.54	43.57	450.00	13.33	15233.33
Agreement (l+p)	76.58	42.91	1033.33	1.67	8033.33
Self-Combined (l+p)	77.03	42.74	916.67	10.67	17564.00
Disagreement (l+p)	75.93	44.14	750.00	14.33	20416.67
Strategy 1	75.68	43.76	1033.33	17.00	22166.67
Strategy 2	76.52	43.77	577.78	13.55	14077.78

Continued on next page

Table 6.10 – *Continued from previous page*

Strategy	**F (%)**	**NIST (%)**	**Opt. Increment**	**Opt. Iteration**	**Opt. Data Size**
Strategy 3	**77.56**	**41.48**	1166.67	14.33	27433.33
Strategy 4	76.86	43.13	1138.89	14.44	21524.00
Strategy 5	76.42	43.56	650.00	16.89	16416.67
Strategy 6	77.31	42.27	1138.89	15.44	22439.45
Strategy 7	76.25	43.46	722.22	14.66	18833.33
Strategy 8	76.71	43.07	666.67	9.00	11000.00
Strategy 9	76.79	42.31	833.33	17.00	22500

Table 6.10: Average Results of the Different Strategies for the Combination of Prosodic and Lexical Features

Figures 6.21 - 6.22 and Table 6.10 presents experimental results of Baseline, Self-training, different strategies of two-view and three-view co-training strategies. Figures 6.23 - 6.24 and Table 6.11 compares Baseline and Self-training results to

Man. Labeled Data = 1K	**Strategy**	**F (%)**	**NIST (%)**
Baseline	-	73.20	50.38
Self-Training	-	73.94	48.81
Co-Training (2-View)	Agreement (l+p)	73.54	49.17
Co-Training (2-View)	Self-Combined (l+p)	74.44	49.19
Co-Training (3-View)	Strategy 4 (l-m-p)	74.61	47.86
Committee-Based (3-View)	Strategy 9 (l+m+p)	**75.57**	**46.50**
Man. Labeled Data = 3K	**Strategy**	**F (%)**	**NIST (%)**
Baseline	-	72.65	49.33
Self-Training	-	74.40	47.14
Co-Training (2-View)	Agreement (l+p)	74.10	46.69
Co-Training (2-View)	Self-Combined (l+p)	74.62	47.38
Co-Training (3-View)	Strategy 4 (m-p-l)	**75.51**	45.79
Co-Training (3-View)	Strategy 3 (AVG)	75.42	**44.86**
Committee-Based (3-View)	Strategy 9 (l+m+p)	74.71	46.74
Man. Labeled Data = 6K	**Strategy**	**F (%)**	**NIST (%)**
Baseline	-	75.21	45.00
Self-Training	-	76.54	43.57
Co-Training (2-View)	Self-Combined (l+p)	77.03	42.74
Co-Training (3-View)	Strategy 6 (l-m-p)	**77.63**	42.12
Co-Training (3-View)	Strategy 3 (AVG)	77.56	**41.48**
Committee-Based (3-View)	Strategy 9 (l+m+p)	76.79	42.31

Table 6.11: Maximum F-measure Scores and Minimum NIST Error Rates of the Different Strategies for the Combination of Prosodic and Lexical Features

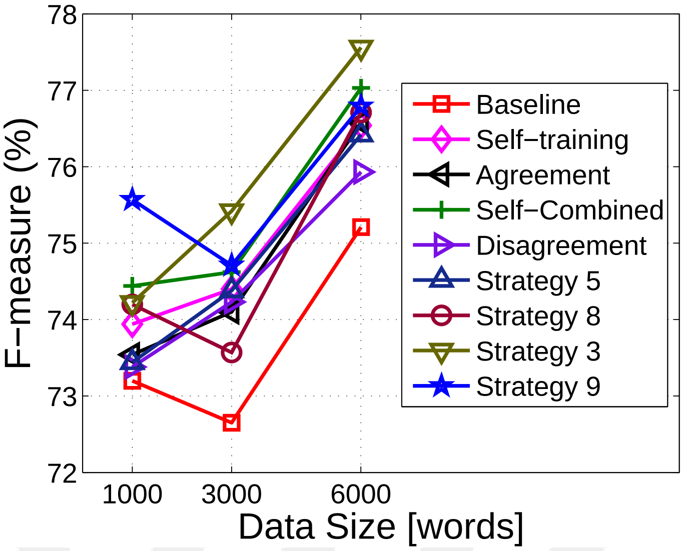


Figure 6.21: Average F-measure Scores of Different Strategies for the Combination of Prosodic and Lexical Features

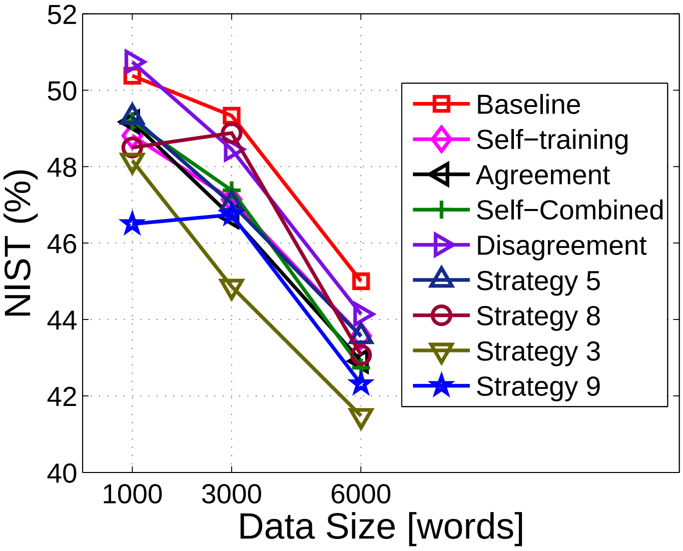


Figure 6.22: Average NIST Error Rates of Different Strategies for the Combination of Prosodic and Lexical Features

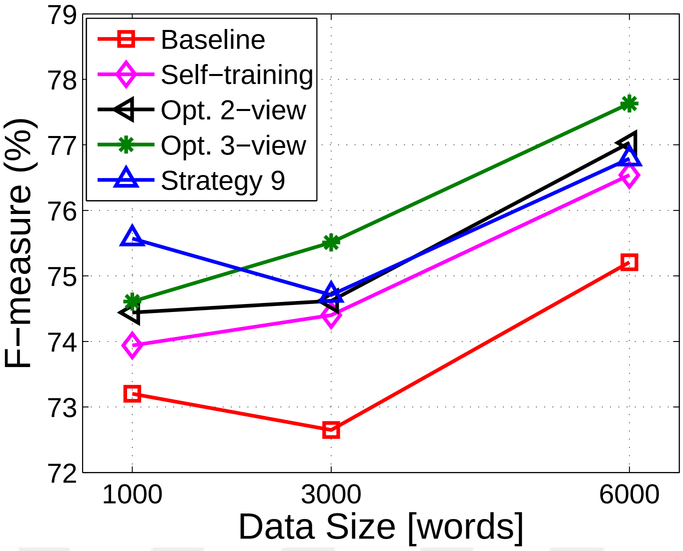


Figure 6.23: Maximum F-measure Scores of Different Strategies for the Combination of Prosodic and Lexical Features

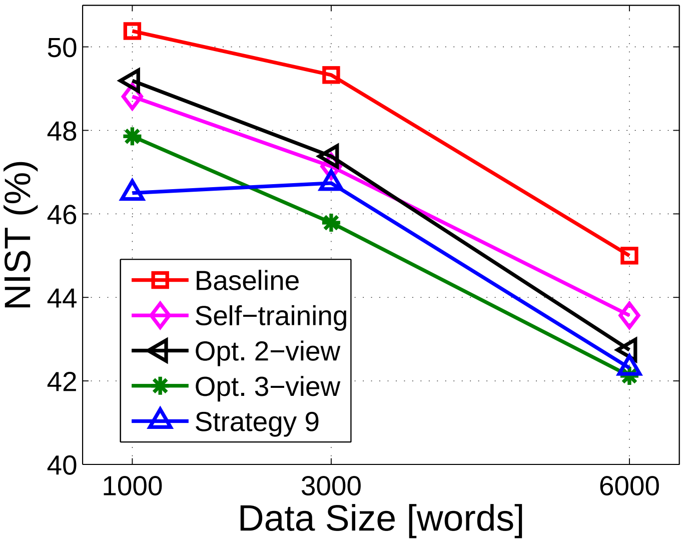


Figure 6.24: Minimum NIST Error Rates of Different Strategies for the Combination of Prosodic and Lexical Features

best 2-view and 3-view results included with Strategy 9, when different sizes of manually labeled data are available. According to these results, when only 1000 manually labeled examples are available, Strategy 9 improved F-measure scores of Baseline, Self-training, 2-view Agreement (l+p), 2-view Self-Combined (p+l) and Strategy 4 (l-m-p) with a percentage improvement of 3.2377%, 2.2045%, 2.7604%, 1.5180%, 1.2867% respectively. When only 3000 manually labeled examples are available, Strategy 4 (m+p+l) improved F-measure scores of Baseline, Self-Training, 2-view Agreement (l+p) and 2-view Self-Combined (l+p) with a percentage improvement of 3.9367%, 1.4919%, 1.9028%, 1.1927% respectively, and finally when only 6000 manually labeled examples are available, Strategy 6 (l-m-p) improved F-measure scores of Baseline, Self-Training and 2-view Self-Combined (l+p) with a percentage improvement of 3.2177%, 1.4241%, 0.7789% respectively.

6.4.6 Experimental Results Based on the Combination of Prosodic and Morphological Features

Tables 6.12, 6.13 and Figures 6.25 - 6.28 present improvement of various semi-supervised methods against the baseline. While analyzing those results, it is assumed that all features are available for the training set, but only morphological and prosodic features are available on the test set. This means, while training multi-view models, lexical view can still contribute in order to improve final binary classifier ($M_{morp+pros}$).

Strategy	**F (%)**	**NIST (%)**	**Opt. Increment**	**Opt. Iteration**	**Opt. Data Size**
Manually Labeled Data Size = 1K					
Baseline (m+p)	81.52	35.19	0	0	0
Self-Training (m+p)	82.26	34.48	450.00	9.67	4016.67
Agreement (m+p)	82.27	34.60	750.00	16.33	13583.33
Self-Combined (m+p)	82.74	33.45	833.33	10.00	8497.33
Disagreement (m+p)	82.14	34.76	283.33	8.67	3333.33

Continued on next page

Table 6.12 – *Continued from previous page*

Strategy	**F (%)**	**NIST (%)**	**Opt. Increment**	**Opt. Iteration**	**Opt. Data Size**
Strategy 1	82.03	34.26	916.67	18.67	16083.33
Strategy 2	82.11	34.50	377.78	10.89	4816.67
Strategy 3	82.42	33.76	333.33	7.33	4203.33
Strategy 4	82.61	33.58	677.78	14.45	6823.78
Strategy 5	82.13	34.80	872.22	10.67	9955.55
Strategy 6	82.37	34.40	466.67	12.89	5658.89
Strategy 7	82.55	33.67	833.33	14.78	12111.11
Strategy 8	82.69	33.45	1166.66	12.33	16000.00
Strategy 9	**83.04**	**32.47**	1166.70	18.00	22500.00
Manually Labeled Data Size = 3K					
Baseline (m+p)	82.31	33.43	0	0	0
Self-Training (m+p)	82.84	33.12	1000.00	11.00	11666.67
Agreement (m+p)	83.15	32.52	700.00	7.33	9733.33
Self-Combined (m+p)	**83.78**	**31.12**	1166.67	12.67	16342.00
Disagreement (m+p)	82.99	32.67	250.00	15.67	6916.67
Strategy 1	83.37	32.41	1000.00	15.67	18000.00
Strategy 2	83.28	32.36	383.33	12.11	7422.22
Strategy 3	82.94	32.59	366.67	20.33	12565.67
Strategy 4	83.24	32.32	761.11	15.78	14021.89
Strategy 5	83.24	32.37	494.44	10.11	6883.33
Strategy 6	83.30	32.19	661.11	13.44	10258.78
Strategy 7	83.20	32.09	722.22	13.22	11222.22
Strategy 8	82.79	32.45	750.00	16.00	14916.67
Strategy 9	83.31	32.19	1033.30	9.67	15633.00
Manually Labeled Data Size = 6K					
Baseline (m+p)	83.25	32.21	0	0	0
Self-Training (m+p)	83.53	31.74	450.00	17.67	13216.67
Agreement (m+p)	83.09	32.33	700.00	13.33	14466.67
Self-Combined (m+p)	83.44	31.41	833.33	15.33	19884.33
Disagreement (m+p)	83.22	32.60	533.33	11.00	9566.67
Strategy 1	83.35	32.26	533.33	15.33	11633.33
Strategy 2	83.61	31.44	572.22	11.89	11261.11
Strategy 3	83.73	31.38	1333.33	13.33	28506.33

Continued on next page

Table 6.12 – *Continued from previous page*

Strategy	**F (%)**	**NIST (%)**	**Opt. Increment**	**Opt. Iteration**	**Opt. Data Size**
Strategy 4	**83.84**	31.22	1000.00	16.33	19134.22
Strategy 5	83.57	31.45	505.56	13.22	13216.67
Strategy 6	83.59	31.82	611.11	8.44	11361.44
Strategy 7	83.09	32.57	594.44	12.00	13627.78
Strategy 8	83.25	32.59	1000.00	8.66	14666.67
Strategy 9	83.85	**31.09**	750.00	14.66	19000.00

Table 6.12: Average Results of the Different Strategies for the Combination of Prosodic and Morphological Features

Figures 6.25 - 6.26 and Table 6.12 presents experimental results of Baseline, Self-training, different strategies of two-view and three-view co-training strategies. Figures 6.27 - 6.28 and Table 6.13 compares Baseline and Self-training results to best 2-view and 3-view results included with Strategy 9, when different sizes of manually labeled data are available. According to these results, when only 1000

Man. Labeled Data = 1K	**Strategy**	**F (%)**	**NIST (%)**
Baseline	-	81.52	35.19
Self-Training	-	82.26	34.48
Co-Training (2-View)	Self-Combined (m+p)	82.74	33.45
Co-Training (3-View)	Strategy 4 (l-m-p)	82.76	33.19
Co-Training (3-View)	Strategy 6 (l-p-m)	82.79	33.69
Committee-Based (3-View)	Strategy 9 (l+m+p)	**83.04**	**32.47**
Man. Labeled Data = 3K	**Strategy**	**F (%)**	**NIST (%)**
Baseline	-	82.31	33.43
Self-Training	-	82.84	33.12
Co-Training (2-View)	Self-Combined (m+p)	**83.78**	**31.12**
Co-Training (3-View)	Strategy 4 (l-m-p)	83.40	32.21
Committee-Based (3-View)	Strategy 9 (l+m+p)	83.31	32.19
Man. Labeled Data = 6K	**Strategy**	**F (%)**	**NIST (%)**
Baseline	-	83.25	32.21
Self-Training	-	83.53	31.74
Co-Training (2-View)	Self-Combined (m+p)	83.44	31.41
Co-Training (3-View)	Strategy 4 (l-p-m)	**84.02**	**30.81**
Committee-Based (3-View)	Strategy 9 (l+m+p)	83.85	31.09

Table 6.13: Maximum F-measure Scores and Minimum NIST Error Rates of the Different Strategies for the Combination of Prosodic and Morphological Features

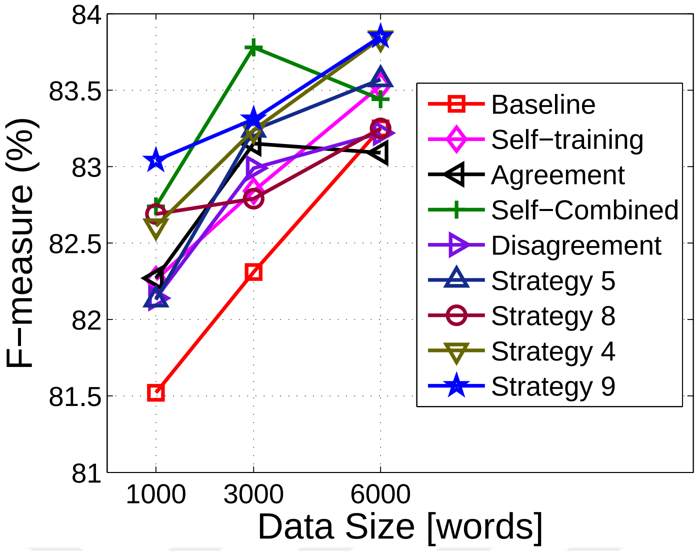


Figure 6.25: Average F-measure Scores of Different Strategies for the Combination of Prosodic and Morphological Features

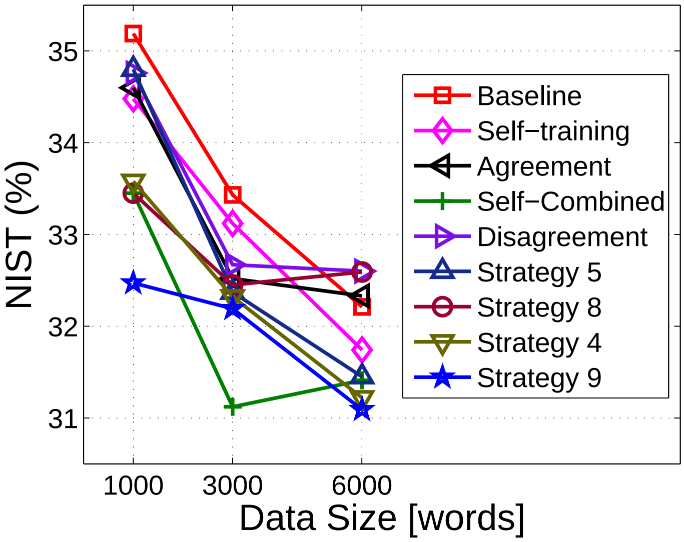


Figure 6.26: Average NIST Error Rates of Different Strategies for the Combination of Prosodic and Morphological Features

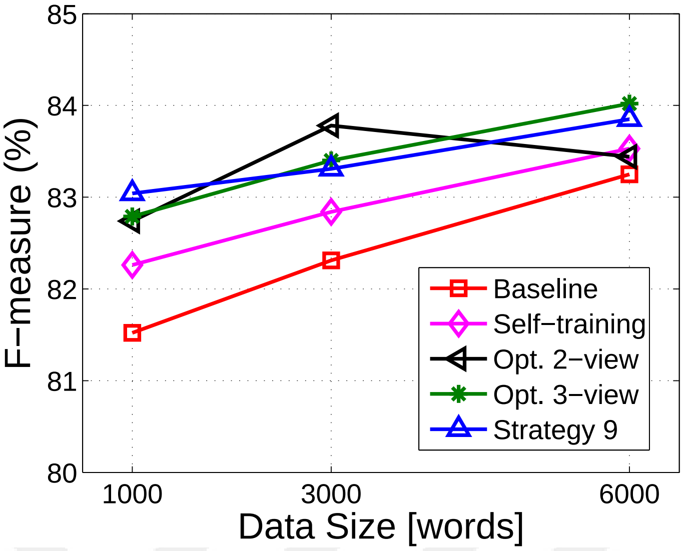


Figure 6.27: Maximum F-measure Scores of Different Strategies for the Combination of Prosodic and Morphological Features

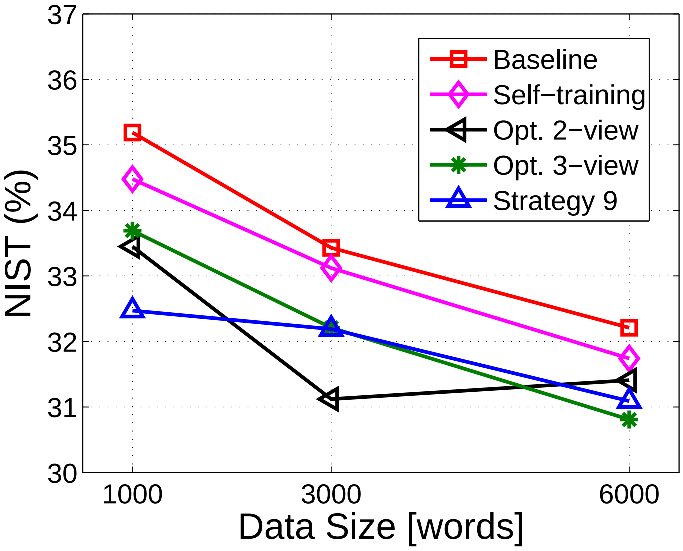


Figure 6.28: Minimum NIST Error Rates of Different Strategies for the Combination of Prosodic and Morphological Features

manually labeled examples are available, Strategy 9 improved F-measure scores of Baseline, Self-training, 2-view Self-Combined (m+p) and 3-view Strategy 6 (l-p-m) with a percentage improvement of 1.8646%, 0.9482%, 0.3626%, 0.3020% respectively. When only 3000 manually labeled examples are available, 2-view Self-Combined strategy (m+p) improved F-measure scores of Baseline and Self-Training with a percentage improvement of 1.7859%, 1.1347% respectively, and finally when only 6000 manually labeled examples are available, Strategy 4 (l-p-m) improved F-measure scores of Baseline, Self-Training and 2-view Self-Combined (l+p) with a percentage improvement of 3.2177%, 1.4241%, 0.7789% respectively.

6.5 Average Results Based on Different Strategies

Table 6.14 presents additional information and contributions of different views for different strategy groups. First column presents initial-view (single-view) cases which are baseline measurements and self-training. The additional information to initial view and contributed views of different 2-view combinations are shown in second and third columns. Note that the strategies do not add any additional features to initial-views feature set They only provide better example selections. When first three column of Table 6.14 is compared, nine different initial view, additional information and contribution combinations are observed. On the other hand, last two columns of Table 6.14 presents additional information provided 3-view methods to initial view. When this columns are compared to initial-view column, six different initial-view, additional information and contribution combinations are observed.

Initial-View	2-view methods		3-view methods	
	Additional Information	Contribution	Additional Information	Contribution
l	m	l + m	m + p	l + m + p
l	p	l + p	m + p	l + m + p
m	l	l + m	p + l	l + m + p
m	p	p + m	p + l	l + m + p
p	l	l + p	l + m	l + m + p
p	m	p + m	l + m	l + m + p
l+m	-	l + m	p	l + m + p
l+p	-	l + p	m	l + m + p
p+m	-	p + m	l	l + m + p

Table 6.14: Baseline and Additional Information Based on Different Experimental Sets

The curves in Figures 6.29 - 6.30 and Table 6.15 presents average results of different strategies. These results were also verified using t-test. According to these results when only 1000 manually labeled examples are available, average results of 2-view strategies, especially the disagreement strategy, outperform baseline (8.0879% relative F-measure improvement) and self-training (4.1911% relative F-measure improvement) results. Moreover 3-view strategies, especially strategy 9 outperform not only baseline (15.3031% relative F-measure improvement)

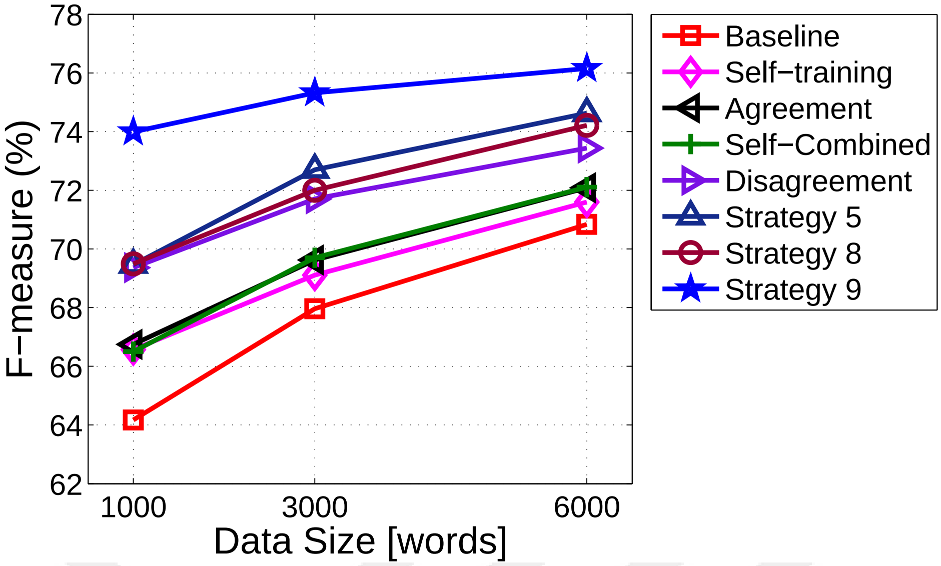


Figure 6.29: Average F-measure Scores of Different Strategies

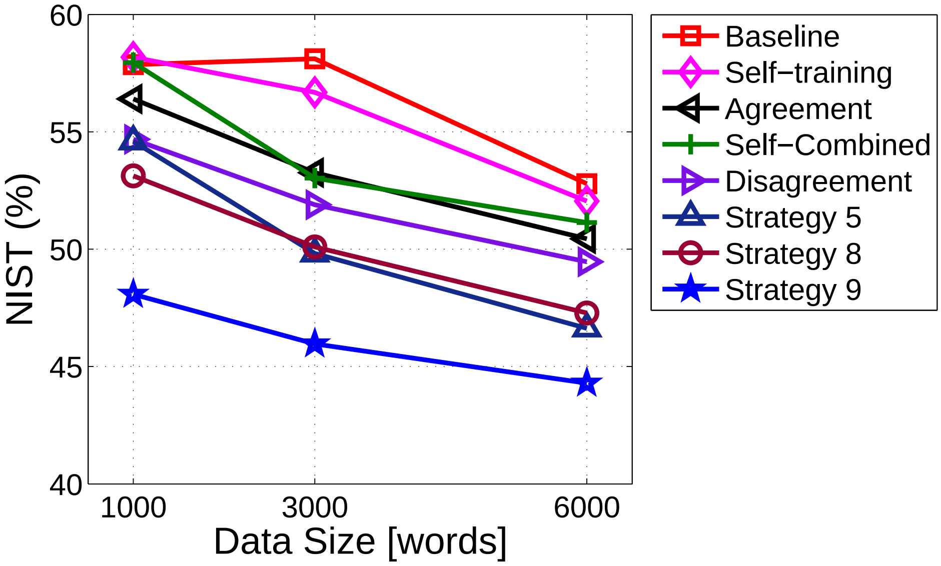


Figure 6.30: Average NIST Error Rates of Different Strategies

and self-training strategies (11.1462% relative F-measure improvements respectively), but also outperforms the 2-view disagreement strategy (6.6753% relative F-measure improvement) and Strategy 9 also outperforms 3-view commitee-based strategy (Strategy 8) (6.4758% relative F-measure improvement). In addition when only 3000 manually labeled examples are available, average results of 2-view strategies, especially the disagreement strategy, outperform baseline

(5.5180% relative F-measure improvement) and self-training (3.7621% relative F-measure improvement) results. Moreover 3-view strategies, especially strategy 9 outperforms not only baseline (10.8299% relative F-measure improvement) and self-training strategies (8.9857% relative F-measure improvement), but also outperforms the 2-view disagreement strategy (5.0342% relative F-measure improvement) and Strategy 9 also outperforms 3-view Strategy 5 (3.6039% relative F-measure improvement). Finally when only 6000 manually labeled examples are available, average results of 2-view strategies, especially the disagreement strategy, outperform baseline (3.6702% relative F-measure improvement) and self-training (2.5698% relative F-measure improvement) results. Moreover 3-view strategies, especially Strategy 9 outperforms not only baseline (7.4958% relative F-measure improvement) and self-training strategies (6.3547% relative F-measure improvement), but also outperforms the 2-view disagreement strategy (3.6901% relative F-measure improvement respectively) and Strategy 9 also outperforms 3-view Strategy 5 (2.0367% relative F-measure improvement).

Manually Labeled Data = 1000 Words					
Algorithm	F-measure (%)	NIST (%)	Avg. # Increment	Avg. # Iters	Avg. # Added
Baseline	64.17	57.86	0	0	0
Self-Training	66.57	58.18	776	15	14709
Agreement (2-View)	66.75	56.40	959	15	15848
Self-Combined (2-View)	66.50	57.95	676	15	9356
Disagreement (2-View)	69.36	54.68	830	12	11970
Strategy 1 (3-View)	69.13	54.53	1086.11	14.00	17186.11
Strategy 2 (3-View)	69.25	54.18	763.89	11.59	11039.82
Strategy 3 (3-View)	69.52	54.53	647.22	11.44	10038.11
Strategy 4 (3-View)	69.29	54.11	820.37	15.85	12639.74
Strategy 5 (3-View)	69.46	54.59	787.04	12.07	11775.93
Strategy 6 (3-View)	69.58	53.78	588.89	12.67	7825.70
Strategy 7 (3-View)	69.05	53.93	925.93	15.43	15546.30
Strategy 8 (3-View)	69.49	53.12	877.77	13.39	13577.78
Strategy 9 (3-View)	**73.99**	**48.08**	894	19	17850
Manually Labeled Data = 3000 Words					
Algorithm	F-measure (%)	NIST (%)	Avg. # Increment	Avg. # Iters	Avg. # Added
Baseline	67.96	58.11	0	0	0
Self-Training	69.11	56.68	781	12	10359
Agreement (2-View)	69.62	53.26	1020	11	12766
Self-Combined (2-View)	69.70	53.03	888	15	13464
Disagreement (2-View)	71.71	51.89	770	13	12411
Strategy 1 (3-View)	71.60	50.87	913.89	15.56	20188.89
Strategy 2 (3-View)	72.08	50.73	760.19	13.32	14405.56
Strategy 3 (3-View)	71.77	50.54	1025.00	15.83	22621.00
Strategy 4 (3-View)	71.91	50.62	1053.70	17.59	20910.04
Strategy 5 (3-View)	72.70	49.82	823.15	13.82	15957.41
Strategy 6 (3-View)	72.23	50.08	968.52	15.22	17855.65
Strategy 7 (3-View)	71.54	51.02	887.96	15.33	17484.26
Strategy 8 (3-View)	72.00	50.10	866.67	15.89	18297.22
Strategy 9 (3-View)	**75.32**	**45.96**	1242	19	29217
Manually Labeled Data = 6000 Words					
Algorithm	F-measure (%)	NIST (%)	Avg. # Increment	Avg. # Iters	Avg. # Added
Baseline	70.84	52.79	0	0	0
Self-Training	71.60	52.05	818	14	18228
Agreement (2-View)	72.08	50.44	1089	13	20978
Self-Combined (2-View)	72.11	51.13	998	15	20971
Disagreement (2-View)	73.44	49.46	946	17	23241
Strategy 1 (3-View)	73.76	47.92	1044.44	17.83	24650.00
Strategy 2 (3-View)	74.48	47.02	964.82	16.48	23338.89
Strategy 3 (3-View)	74.26	47.23	1222.22	17.06	33187.72
Strategy 4 (3-View)	74.10	47.83	1089.81	16.80	23128.02
Strategy 5 (3-View)	74.63	46.63	1018.52	17.48	24874.08
Strategy 6 (3-View)	74.59	46.69	983.33	16.98	23107.39
Strategy 7 (3-View)	73.96	47.49	1037.96	17.37	26273.15
Strategy 8 (3-View)	74.21	47.28	1152.78	14.72	24250.00
Strategy 9 (3-View)	**76.15**	**44.29**	1153	20	30333

Table 6.15: Average Results of Different Strategies When Only 1000, 3000 and 6000 Manually Labeled Examples are Available

6.6 Statistical Analysis of the Experimental Results

According to the experimental results that presented in Figures 6.29 - 6.30 and Table 6.15, Strategy 9 is the most effective strategy. We also apply two tail t-test over average results of different strategies for different feature sets to show that Strategy 9 provides significant improvement, statistically. Table 6.16 present average F-measure scores and NIST error rates of different strategies when different sizes of manually examples are available for different feature sets. For each feature set, average results of different strategies in Table 6.16, and average results of Strategy 9 were considered as sample mean (μ) and the null-hypothesis of the following two tail t-test, respectively.

- $h_0 : \mu =$ null hypothesis (Average results of Strategy 9 exists in confidence interval i.e., Strategy 9 does not provide significant improvement.)
- $h_1 : \mu \neq$ null hypothesis (Average results of Strategy 9 does not exist in confidence interval i.e., Strategy 9 provides significant improvement.)

	l			m			p		
Algorithm	Contribution	F (%)	NIST (%)	Contribution	F (%)	NIST (%)	Contribution	F (%)	NIST (%)
Baseline	l	41.21	85.51	m	75.68	48.66	p	71.42	54.26
Self-Training	l	45.43	86.90	m	76.01	48.26	p	72.39	51.47
Agreement	l+m	47.65	75.27	m+l	76.08	48.36	p+l	72.38	51.44
Agreement	l+p	46.07	80.68	m+p	76.39	47.00	p+m	72.69	50.80
Self-Combined	l+m	46.32	79.65	m+l	75.81	48.72	p+l	72.80	50.91
Self-Combined	l+p	45.72	82.32	m+p	76.45	47.18	p+m	72.81	50.66
Disagreement	l+m	54.13	70.33	m+l	75.93	48.80	p+l	72.23	51.65
Disagreement	l+p	58.87	68.79	m+p	76.53	46.97	p+m	72.14	52.10
Strategy 5	l+m+p	50.32	77.09	l+m+p	76.47	47.27	l+m+p	72.11	52.28
Strategy 8	l+m+p	47.91	76.77	l+m+p	76.42	47.27	l+m+p	72.40	51.33
Strategy 9	l+m+p	63.34	60.33	l+m+p	77.45	45.09	l+m+p	73.07	49.80
	l and m			l and p			p and m		
Algorithm	Contribution	F (%)	NIST (%)	Contribution	F (%)	NIST (%)	Contribution	F (%)	NIST (%)
Baseline	l,m	76.31	47.63	l,p	73.69	48.24	p,m	82.36	33.61
Self-Training	l,m	76.35	47.98	l,p	74.96	46.51	p,m	82.88	33.11
Agreement	l+m	76.52	47.37	l+p	74.74	46.26	p+m	82.84	33.15
Self-Combined	l+m	76.27	48.50	l+p	75.36	46.44	p+m	83.32	31.99
Disagreement	l+m	76.46	47.40	l+p	74.51	47.78	p+m	82.78	33.34
Strategy 5	l+m+p	76.94	46.34	l+m+p	74.75	46.63	l+m+p	82.98	32.87
Strategy 8	l+m+p	76.93	45.96	l+m+p	74.83	46.82	l+m+p	82.91	32.83
Strategy 9	l+m+p	77.95	44.28	l+m+p	75.69	45.18	l+m+p	83.40	31.92

Table 6.16: Average Results of Different Strategies When Different Sizes of Manually Examples are Available for Different Feature Sets

Feature Set	Null-Hypothesis	N	μ	σ	SE Mean	95% CI	T	P
l	63.34% F-measure	11	49.72	6.54	1.97	(45.33 - 54.12)	-6.90	0.000
l	60.33% NIST	11	76.70	7.82	2.36	(71.44 - 81.95)	6.94	0.000
m	77.45% F-measure	11	76.292	0.484	0.146	(75.967 - 76.617)	-7.93	0.000
m	45.09% NIST	11	47.599	1.108	0.334	(46.855 - 48.343)	7.51	0.000
p	73.07% F-measure	11	72.404	0.448	0.135	(72.103 - 72.704)	-4.94	0.001
p	49.80% NIST	11	51.518	1.145	0.345	(50.749 - 52.288)	4.98	0.001
l and m	77.95% F-measure	8	76.716	0.563	0.199	(76.246 - 77.186)	-6.21	0.000
l and m	44.28% NIST	8	46.932	1.349	0.477	(45.804 - 48.060)	5.56	0.001
l and p	75.69% F-measure	8	74.816	0.593	0.210	(74.320 - 75.312)	-4.17	0.004
l and p	45.18% NIST	8	46.730	0.936	0.331	(45.948 - 47.513)	4.68	0.002
p and m	83.40% F-measure	8	82.933	0.324	0.114	(82.663 - 83.204)	-4.08	0.005
p and m	31.92% NIST	8	32.854	0.608	0.215	(32.346 - 33.362)	4.35	0.003

Table 6.17: Results of the Two Tail t-test

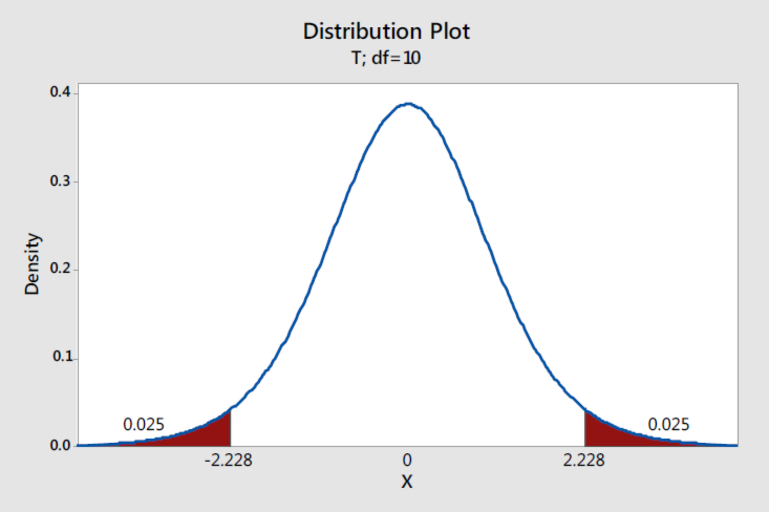


Figure 6.31: t-distribution With df=10

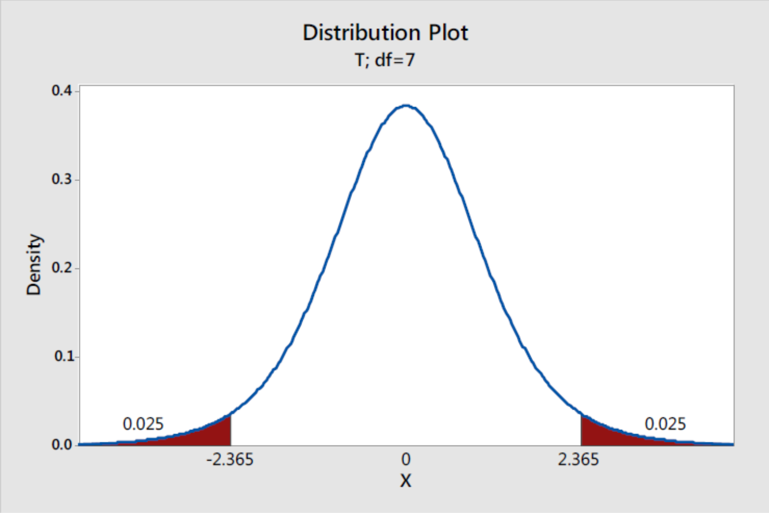


Figure 6.32: t-distribution With df=7

Table 6.17 presents the two tail t-test results. The definitions of the variables in Table 6.17 are as follows: "N" presents the sample size, "μ" presents the sample mean, "σ" presents the variance, "SE Mean" presents the standard error mean, "95% CI" presents the 95% confidence interval, "T" presents the t-value, and finally "P" presents the p-value. When only lexical, prosodic or morphological features are available, 11 different experimental results are presented in Table 6.16, which corresponds to degree of freedom (df) equals to 10 and $t_{\alpha=0.025} = \pm 2.228$, as shown in the Figure 6.31. On the other hand, when binary combinations of prosodic, lexical and morphological features are available, 8 different experimental results are presented in Table 6.16, which corresponds to degree of freedom (df) equals to 7 and $t_{\alpha=0.025} = \pm 2.365$, as shown in the Figure 6.32.

Two tail t-test rejects the null hypothesis (case h_1) when $P \leq \alpha/2 = 0.025$, $T < t_{\alpha=0.025}$ for the negative tail, and the value of the null-hypothesis exceeds 95% CI under $\alpha = 0.05$ or 95% CI. Otherwise, fails to reject the null hypothesis (case h_0). Minitab has been used as a statistical computation software.

The two tail t-test results that presented in Table 6.17 shows that,

- When only lexical features are available, average F-measure score of Strategy 9 (63.34% F-measure) exceeds 95% CI, and this result provides $T = -6.90 < t_{\alpha=0.025} = -2.228$ and $P = 0.000 < 0.025$. On the other hand, average NIST error rate of Strategy 9 (60.33% NIST) is under 95% CI, and this result provides $T = 6.94 > t_{\alpha=0.025} = 2.228$ and $P = 0.000 < 0.025$.

- When only morphological features are available, average F-measure score of Strategy 9 (76.292% F-measure) exceeds 95% CI, and this result provides $T = -7.93 < t_{\alpha=0.025} = -2.228$ and $P = 0.000 < 0.025$. On the other hand, average NIST error rate of Strategy 9 (47.599% NIST) is under 95% CI, and this result provides $T = 7.51 > t_{\alpha=0.025} = 2.228$ and $P = 0.000 < 0.025$.

- When only prosodic features are available, average F-measure score of Strategy 9 (72.404% F-measure) exceeds 95% CI, and this result provides $T = -4.94 < t_{\alpha=0.025} = -2.228$ and $P = 0.001 < 0.025$. On the other hand, average NIST error rate of Strategy 9 (51.518% NIST) is under 95% CI, and this result provides $T = 4.98 > t_{\alpha=0.025} = 2.228$ and $P = 0.001 < 0.025$.

- When the combination of lexical and morphological features are available, average F-measure score of Strategy 9 (76.716% F-measure) exceeds 95% CI, and this result provides $T = -6.21 < t_{\alpha=0.025} = -2.365$ and $P = 0.000 < 0.025$. On the other hand, average NIST error rate of Strategy 9 (49.932% NIST) is under 95% CI, and this result provides $T = 5.56 > t_{\alpha=0.025} = 2.365$ and $P = 0.001 < 0.025$.

- When the combination of lexical and prosodic features are available, average F-measure score of Strategy 9 (74.816% F-measure) exceeds 95% CI, and this result provides $T = -4.17 < t_{\alpha=0.025} = -2.365$ and $P = 0.004 < 0.025$. On the other hand, average NIST error rate of Strategy 9 (46.730% NIST) is under 95% CU, and this result provides $T = 4.68 > t_{\alpha=0.025} = 2.365$ and $P = 0.002 < 0.025$.

- When the combination of prosodic and morphological features are available, average F-measure score of Strategy 9 (82.933% F-measure) exceeds 95% CI, and this result provides $T = -4.08 < t_{\alpha=0.025} = -2.365$ and $P = 0.005 < 0.025$. On the other hand, average NIST error rate of Strategy 9 (32.854% NIST) is under 95% CI, and this result provides $T = 4.35 > t_{\alpha=0.025} = 2.365$ and $P = 0.003 < 0.025$.

These results show that the improvement of Strategy 9 is statistically significant.

6.7 Analysis and Discussion

Table 6.1 and Figures 6.1 - 6.2 present comparative results of different baseline models. In those results, it has been shown that strongest models are trained by morphological features and they are followed by prosodic and lexical features, when the baseline results of lexical-only, prosodic-only and morphological-only models are compared. The reason is that broadcast news are presented with a regular grammar and emphasis. Since, the part of speech (POS) structure of Turkish is Subject + Object + Verb, the Verb POS tag provide important cues on sentence boundaries. However, some hard examples for the morphological models, which are wrongly labeled can be corrected by either prosodic or lexical model. For instance in the first and second cases, the morphological model estimated a sentence boundary (SB) because the words “toplandi” (assembled) and “kanitlamiyor” (is not proving) are verbs, but for the lexical view because of they are followed by conjunctions “ve” (and), “tersine” (in contrast) the lexical model corrected the decision by hypothesizing as non-sentence boundary (WB). In addition to the prosodic view, the interested words are preceded by a short pause without any hidden-speech event related with a sentence boundary.

Similarly, morphological or lexical models may also correct an indecisive decision of the prosodic model. For instance, in the third case, the prosodic model is failed because of a self-correction is preceded by a long pause after the word “onumuzdeki” (forthcoming) but corrected by the lexical and morphological models. In the fourth case the sentence boundary is followed by a short pause, in addition, the acoustical condition of this example is telephone conversation hence prosodic model failed, but morphological and lexical models hypothesized as sentence boundary with a higher confidence score since the word “yasandi” (happened) is a verb for the morphological model and the word “yasandi” has a higher probability to precede a sentence boundary for the lexical model.

Case 1: *...hastane cevresinde* ***toplandi*** *{WB} ve kendisini oven...* (...they assembled near the hospital and commended himself...)

Case 2: *...resmi veriler* ***kanitlamiyor*** *{WB} tersine Alman Istatistik Enstitusu'nun...* (...official data do not offer a proof. In contrast Federal Statistical Office of Germany...)

Case 3: *...aciklama yapti ve tahmin ediyorum* ***onumuzdeki*** *{WB} [pause] [repetition: onumuzdeki] sure icinde...* (...made an explanation and I guess in the forthcoming...)

Case 4: *... oldukca ilginc sahneler* ***yasandi*** *{SB} ornegin ...* (...very interesting events have occured. For instance..)

Another interesting question is whether there is a correlation between initial manually labeled data size with either amount of iteration, the amount of automatically labeled data or amount of automatically added data in one iteration. When we examine average results of the different strategies for different feature sets in section 6.4 and Table 6.15 we can not observe such a correlation. By performing various experiments, it has been observed that amount of either increment, iterations or added examples are correlated with the distribution of the initial manually labeled data rather than its size.

6.8 Scenario of Concatenating Trained Models to Online ASR Systems

This section presents improvement of different semi-supervised learning strategies in table 6.18 under the following scenario. In this scenario, different lexical models were trained using different learning strategies such as baseline, self-training, co-training with disagreement strategy (p+l) and Committee-Based Learning Strategy 9, using 1000 initially manually labeled data. Then it has assumed that those models were integrated with an ASR system, which also provides lexical features and marks sentence boundaries using the models. Since in experiments, test set used only for performance evaluation, this set also could be considered as output of this concatenated ASR system. Table 6.18 presents different cases under this scenario.

In the first case, baseline model marks the word “ismi” as sentence boundary, however different models trained by using Self-Training, Co-Training with Disagreement Strategy and Committee-Based Learning Strategy 9 corrects the decision. This case can be considered as a simple example since lexical model corrects itself by using self-training method. In the second case, the sentence boundary after the word “aliniyor” could be detected by the models which are trained by using Co-Training with Disagreement Strategy and Committee-Based Learning Strategy 9. In this example we observe the contribution of prosodic view on lexical model training process. Finally in the last example the sentence boundary after the word “dinliyorsunuz” could be detected by using only the model trained by using Committee-Based Learning Strategy 9. Therefore in this example we observe contribution of prosodic and morphological view on lexical model training process.

This scenario shows that proposed learning methods are helpful to train statistical models to be concatenated with online ASR systems.

Case 1					
WORD	Baseline	Self-Training	Disagreement (L+P)	Strategy 9	Original
bu	n	n	n	n	n
degisimin	n	n	n	n	n
ilk	n	n	n	n	n
onemli	n	n	n	n	n
ismi	**s**	**n**	**n**	**n**	**n**
savunma	n	n	n	n	n
bakani	n	n	n	n	n
donald	n	n	n	n	n
Case 2					
WORD	Baseline	Self-Training	Disagreement (L+P)	Strategy 9	Original
bolton	n	n	n	n	n
istifa	n	n	n	n	n
etti	s	s	s	s	s
irana	n	n	n	n	n
yaptirimlar	n	n	n	n	n
bugn	n	s	n	n	n
pariste	n	n	n	n	n
ele	n	n	n	n	n
aliniyor	**n**	**n**	**s**	**s**	**s**
avrupa	n	n	n	n	n
birliginin	n	n	n	n	n
Case 3					
WORD	Baseline	Self-Training	Disagreement (L+P)	Strategy 9	Original
sesi	n	n	n	n	n
yayinlarini	n	n	n	n	n
kisa	n	n	n	n	n
dalgadan	n	n	n	n	n
ve	n	n	n	n	n
ntv	n	n	n	n	n
radyodan	n	n	n	n	n
dinliyorsunuz	**n**	**n**	**n**	**s**	**s**
yayinlarimizla	n	n	n	n	n
ilgili	n	n	n	n	n
bilgi	n	n	n	n	n

Table 6.18: Improvements of Different Semi-Supervised Learning Strategies on Lexical Model

Chapter 7

Conclusion

In this work, new effective semi-supervised machine learning strategies were proposed for sentence segmentation task when only small sets of sentence boundary-labeled data are available. Three-view co-training and committee-based strategies on the sentence boundary classification problem using lexical, prosodic and morphological information, were proposed.

The experimental results on the Voice of America (VOA) Turkish BN data show the effectiveness of these algorithms for the sentence segmentation task. The experimental results show that baseline models of lexical, prosodic, morphological and their binary combinations (lexical+morphological, lexical+prosodic, prosodic+morphological) are highly improved by using newly proposed three-view co-training and committee-based learning approaches especially when only a small set of manually labeled examples are available. For instance Committee-Based Learning Strategy 9 improved the average baseline F-measure of 64.17% to 73.99%, 67.96% to 75.32% and 70.84% to 76.15% when only 1000, 3000 and 6000 initial manually labeled examples are available, respectively. This strategy improved the baseline F-measure of 67.66% to 75.15% on the average when only small different sizes of initial manually labeled examples are available.

In addition the experimental results show that the newly proposed strategies called Three-View Co-Training Strategy 5 and Committee-Based Learning Strategies (Strategy 8 and Strategy 9) outperform not only all the other strategies including agreement and self-combined with two-view co-training but also the best strategy based on disagreement with two-view co-training described in [7]. On the other hand, in section 6.8 it has been shown that those strategies could be used to train lexical features based effective models, which can easily integrated with online ASR systems.

References

[1] D. Jones, F. Wolf, E. Gibson, E. Williams, E. Fedorenko, D. Reynolds, and M. Zissman, "Measuring the readability of automatic speech-to-text transcripts," in *Proc. Eurospeech*, Geneva, Switzerland, 2003.

[2] J. Kolar, "Automatic segmentation of speech into sentence-like units," Ph.D. dissertation, University of West Bohemia, Pilsen, Czech Republic, 2008.

[3] U. Guz, B. Favre, D. Hakkani-Tür, and G. Tur, "Generative and discriminative methods using morphological information for sentence segmentation of Turkish," *Special Issue on Processing Morphologically Rich Languages, IEEE Transactions on Audio, Speech and Language Processing*, vol. 17, pp. 895–903, 2009.

[4] E. Shriberg, A. Stolcke, D. Hakkani-Tür, and G. Tur, "Prosody-based automatic segmentation of speech into sentences and topics," *Speech Communication*, vol. 32, pp. 127–154, 2000.

[5] S. Cuendet, D. Hakkani-Tür, and G. Tur, "Model adaptation for sentence segmentation from speech," in *Proc. IEEE/ACL SLT Workshop*, Aruba, 2006.

[6] U. Guz, S. Cuendet, D. Hakkani-Tür, and G. Tur, "Co-training using prosodic and lexical information for sentence segmentation," in *Proc. Interspeech, ISCA*, Antwerp, Belgium, 2007, pp. 2597–2600.

[7] U. Guz and S. Cuendet and G. Tur and D. Hakkani-Tür, “Multi-view semi-supervised learning for dialog-act segmentation of speech,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 2, pp. 320–329, 2010.

[8] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters, “The ICSI MRDA meeting corpus,” in *Proc. IEEE ICASSP*, vol. 1, 2003, pp. I–364–I–367.

[9] E. Shriberg, R. Dhillon, S. Bhagat, J. Ang, and H. Carvey, “The ICSI meeting recorder dialog act (MRDA) corpus,” in *Proc. SigDial*, Boston, MA, 2004.

[10] D. Dalva, U. Guz, and H. Gurkan, “Effective semi-supervised learning strategies for automatic sentence segmentation,” *Pattern Recognition Letters*, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865517303720

[11] G. Tur and D. Hakkani-Tür, *Human / Human Conversation Understanding.* John Wiley and Sons, Ltd, 2011.

[12] M. Shugrina, “Formatting time aligned asr transcripts for readability,” in *Proc. Annual Conference on the American Chapter of the ACL*, LA, USA, 2010, pp. 128–206.

[13] A. Stolcke and E. Shriberg, “Statistical language modeling for speech disfluencies,” in *Proc.* IEEE ICASSP, Atlanta, GA, 1996, pp. 405–408.

[14] J. Kolar, J. Svec, and J. Psutka, “Automatic punctuation annotation in Czech broadcast news speech,” in *Proc. Conf. Speech Comput.*, vol. 9, 2004.

[15] D. Hakkani-Tür, G. Tur, A. Stolcke, and E. Shriberg, “Combining words and prosody for information extraction from speech,” in *Proceedings of European Conference of Speech Communication and Technology (EUROSPEECH)*, 1999.

[16] Y. Liu, A. Stolcke, E. Shriberg, and M. Harper, “Using conditional random fields for sentence boundary detection in speech,” in *Proc. Annu. Meeting Assoc. Comp. Linguist. (ACL)*, Ann Arbor, MI, 2005.

[17] A. L. Berger, S. A. D. Pietra, and V. J. D. Pietra, “A maximum entropy approach to natural language processing,” *Comp. Linguist.*, vol. 22, pp. 39–71, 1996.

[18] M. Zimmermann, D. Hakkani-Tür, J. Fung, N. Mirghafori, L. Gottlieb, Y. Liu, and E. Shriberg, “The ICSI+ multi-lingual sentence segmentation system,” in *Proc. Interspeech*, Pittsburgh, PA, 2006.

[19] F. Batista, I. Trancoso, and N. J. Mamede, “Comparing automatic rich transcription for portuguese, spanish and english broadcast news,” in *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding*, 2009.

[20] F. Batista, H. Moniz, I. Trancoso, H. Meinedo, A. I. Mata, and N. J. Mamede, “Extending the punctuation module for european portuguese,” in *Proceedings of Conference of International Speech Communication Association*, 2010.

[21] S. Cuendet, D. Hakkani-Tür, and G. Tur, “Model adaptation for sentence segmentation from speech,” in *Proceedings of Speech and Language Technology*, 2010.

[22] G. Tur, U. Guz, and D. Hakkani-Tür, “Model adaptation for dialog act tagging,” in *IEEE/ACL Workshop on Spoken Language Technology*, ARUBA, 2006, pp. 94–97.

[23] Y. Liu, E. Shriberg, A. Stolcke, B. Peskin, J. Ang, D. Hillard, M. Ostendorf, M. Tomalin, P. Woodland, and M. Harper, “Structural metadata research in the EARS program,” in *Proc. IEEE ICASSP*, 2005.

[24] Y. Liu, E. Shriberg, A. Stolcke, and M. Harper, "Comparing HMM, maximum entropy, and conditional random fields for disfluency detection," in *Proc. Interspeech*, 2005, pp. 3313–3316.

[25] C. Zong and F. Ren, "Chinese utterance segmentation in spoken language translation," in *Proc. CICLing*, vol. 4, 2003, pp. 516–525.

[26] J. Fung, D. Hakkani-Tür, M. M. Doss, E. Shriberg, S. Cuendet, and N. Mirghafori, "Prosodic features and feature selection for multilingual sentence segmentation," in *Proc. Interspeech*, Antwerp, Belgium, 2007.

[27] K. Oflazer, "Two-level description of Turkish morphology," *Literary Linguist. Comput.*, vol. 9, 1994.

[28] G. Tur, "A statistical information extraction system for Turkish," Ph.D. dissertation, Dept. of Comput. Sci., Bilkent Univ., Ankara, Turkey, 2000.

[29] Z. Huang, L. Chen, and M. Harper, "Purdue prosodic feature extraction tool based on praat," *School of Electrical and Computer Engineering, Purdue University, USA*, 2006.

[30] D. Dalva, I. Revidi, U. Guz, and H. Gurkan, "Extraction and comparison of various prosodic feature sets on sentence segmentation task for Turkish broadcast news data," in *IEEE International Joint Conference on Computer Sciences and Software Engineering (JCSSE)*, vol. 11, Pattaya, Thailand, 2014, pp. 70–73.

[31] D. Dalva and I.D. Revidi and U. Guz and H. Gurkan, "Extracting the prosodic information for Turkish broadcast news data and using on the sentence segmentation task," in *IEEE Signal Processing and Communications Applications Conference (SIU)*, vol. 22, Trabzon, Turkey, 2014, pp. 1810–1813.

[32] D. Dalva, "Automatic speech recognition system for Turkish spoken language," Master's thesis, Graduate School of Science and Engineering, FMV

Isik University, (Supervisor: Assoc. Prof. Umit Guz, Co-supervisor: Assoc. Prof. Hakan Gurkan), Istanbul, Turkey, 2012.

[33] I. D. Revidi, “Prosodic, morphological and lexical feature extraction of Turkish broadcast news data,” Master’s thesis, Graduate School of Science and Engineering, FMV Isik University, (Supervisor: Assoc. Prof. Umit Guz, Co-supervisor: Assoc. Prof. Hakan Gurkan), Istanbul, Turkey, 2014.

[34] X. Zhu, “Semi-supervised learning with graphs,” Ph.D. dissertation, Language Technologies Institute, School of Computer Science, Cornegie Mellon University, USA, 2005.

[35] K. Nigam and R. Ghani, “Understanding the behaviour of co-training,” in *Proc. Workshop Text Mining 6th ACM SIGKDD*, 2000.

[36] J. L. Gauvain and C.-H. Lee, “Maximum a posteriori estimation for multivariate gaussian mixture observation of Markov chains,” *IEEE Trans. Speech Audio Process.*, vol. 2, pp. 291–298, 1994.

[37] M. Bacchiani and B. Roark, “Unsupervised language model adaptation,” in *Proc. IEEE ICASSP*, Hong Kong, 2003, pp. 173–176.

[38] D. Hakkani-Tür, G. Tur, M. Rahim, and G. Riccardi, “Unsupervised and active learning in automatic speech recognition for call classification,” in *Proc. IEEE ICASSP*, Montreal, QC, Canada, 2004, pp. 429–432.

[39] W. Wang, Z. Huang, and M. Harper, “Semi-supervised learning for partofspeech tagging of Mandarin transcribed speech,” in *Proc. IEEE ICASSP*, Honolulu, HI, 2007, pp. 137–140.

[40] R. Mihalcea, “Co-training and self-training for word sense disambiguation,” in *Proc. Conf. Comput. Natural Lang. Learn. (CoNLL)*, Boston, MA, 2004.

[41] D. McClosky, E. Charniak, and M. Johnson, “Effective self-training for parsing,” in *Proc. NAACL*, New York, 2006.

[42] A. Blum and T. Mitchell, “Combining labeled and unlabeled data with co-training,” *Proc. COLT, Madison, WI*, 1998.

[43] T. M. Mitchell, “The role of unlabeled data in supervised learning,” in *Proc. Int. Colloquium Cognitive Sci.*, vol. 6, San Sebastian, Spain, 1999.

[44] S. Abney, “Bootstrapping,” in *Proc. Annu. Meeting ACL*, 2002.

[45] S. Kiritchenko and S. Matwin, “Email classification with co-training,” *Centre for Advanced Studies on Collaborative Research (CASCON)*, 2001.

[46] G. Tur, “Co-adaptation: Adaptive co-training for semi-supervised learning,” in *IEEE ICASSP*, 2009, pp. 3721–3724.

[47] J. H. Jeon and Y. Liu, “Automatic prosodic event detection using a novel labeling and selection method in co-training,” *Speech Communication*, vol. 54, no. 3, pp. 445 – 458, 2012.

[48] X. Cui, J. Huang, and J. Chien, “Multi-view and multi-objective semi-supervised learning for HMM based automatic speech recognition,” *IEEE Trans. on Audio, Speech, and Lang. Process.*, vol. 20, no. 7, pp. 1923 – 1935, 2012.

[49] E. Arisoy, D. Can, S. Parlak, H. Sak, and M. Saraclar, “Turkish broadcast news transcriptions and retrieval,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 5, pp. 874 – 883, 2009.

[50] M. Saraclar, “Turkish broadcast news speech and transcripts LDC2012S06.” in *Linguistic Data Consortium, Philadelphia*, 2012.

[51] E. O. Selkirk, “The relation between sound and structure,” *Phonology and Syntax*, 1986.

[52] J. Ang, Y. Liu, , and E. Shriberg, “Automatic dialog act segmentation and classification in multiparty meetings,” in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, 2005.

[53] R. Fernandez and R. W. Picard, “Dialog act classification from prosodic features using support vector machines,” in *Proceedings of Speech Prosody*, 2002.

[54] A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. EEs-Dykema, and M. Meeter, “Dialog act modeling for automatic tagging and recognition of conversational speech,” in *Computational Linguistics*, 2000.

[55] H. Wright, M. Poesio, and S. Isard, “Using high level dialogue information for dialogue act recognition using prosodic features,” in *Proceedings of ESCA Tutorial and Research Workshop on Dialogue and Prosody*, 1999.

[56] J. Ang, R. Dhillon, A. Krupski, E. Shriberg, and A. Stolcke, “Prosody-based automatic detection of annoyance and frustation in human-computer dialog,” in *Proceedings of International Conference on Speech and Language Processing*, 2002.

[57] R. Tato, R. Santos, R. Kompe, and J. Pardo, “Emotional space improves emotion recognition,” in *Proceedings of International Conference on Spoken Language Processing*, 2002.

[58] S. Yacoub, S. Simske, X. Lin, and J. Burns, “Recognition of emotions in interactive voice response systems,” in *Proceedings of European Conference on Speech Communication and Technology*, 2003.

[59] E. Shriberg, L. Ferrer, S. Kajarekar, A. Ventakaraman, and A. Stolcke, “Modeling prosodic feature sequences for speaker recognition,” *Speech Comm.*, vol. 46, pp. 455–472, 2005.

[60] D. A. R. Andre G. Adami, Radu Mihaescu and J. J. Godfrey, “Modeling prosodic dynamics for speaker recognition,” in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, 2003.

[61] B. Peskin, J. Navratil, J. Abramson, D. Jones, D. Klusacek, D. A. Reynolds, and B. Xiang, “Using prosodic and conversational features for

high-performance speaker recognition," in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, 2003.

[62] D. A. Reynolds, W. Andrews, J. Campbell, J. Navratil, B. Peskin, A. Adami, Q. Jin, D. Klusacek, J. Abramson, R. Mihaescu, J. Godfrey, D. Jones, and B. Xiang, "Exploiting high-level information for high-accuracy speaker recognition," in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, 2003.

[63] J. G. Fung, "Automatic design of prosodic features for sentence segmentation," Ph.D. dissertation, Electrical Engineering and Computer Sciences University of California at Berkeley, USA, 2011.

[64] M. Swerts, "Prosodic features at discourse boundaries of different strength," *Journal of the Acoustic Society of America*, 1997.

[65] J. Snedecker and J. Trueswell, "Using prosody to avoid ambiguity," *Journal of Memory and Language*, 2003.

[66] R. W. Frick, "Communication emotion: The role of prosodic features," in *Psychological Bulletin*, 1985.

[67] Y. Gotoh and S. Renals, "Sentence boundary detection in broadcast speech transcripts," in *Proceedings of the International Speech Communication Assocciation (ISCA) Workshop: Automatic Speech Recognition: Challenges for the New Millenium*, 2000.

[68] L. Ferrer, "Prosodic features for the switchboard database," *Speech Technology and Research Lab., SRI International, Merlo Park, CA 94025*, 2002.

[69] H. Bratt and L. Ferrer, "Algemy [computer software] sri international," 2011.

[70] U. Guz, G. Tur, D. Hakkani-Tür, and S. Cuendet, "Cascaded model adaptation for dialog act segmentation and tagging," *Computer, Speech and Language (CSL), Elsevier*, vol. 24, no. 2, pp. 289 – 306, 2010.

[71] B. Favre, D. Hakkani-Tür, and S. Cuendet, "Icsiboost." *[Online]. Available: http://code.google.come/p/icsiboost*, vol. 9, 2007.

[72] R. E. Schapire, "The boosting approach to machine learning an overview," *ATT Labs Research, Shannon Laboratory*, 2001.

[73] R. E. Schapire, "The strength of weak learnability," *Machine Learning, Kluwer Academic Publishers, Boston*, vol. 5, pp. 197–227, 1990.

[74] A. Niculescu-Mizil and R. A. Caruana, "Obtaining calibrated probabilities from boosting," in *Proc. of the Twenty-First Conference on Uncertainty in Artificial Intelligence (UAI)*, 2005.