Topic Learning in Text and Conversational Speech

Constantinos Boulis

A dissertation submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

University of Washington

2005

Program Authorized to Offer Degree: Electrical Engineering

University of Washington
Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Constantinos Boulis

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Chair of Supervisory Committee:

_____

Mari Ostendorf

Reading Committee:

_____

Mari Ostendorf

_____

Eve A. Riskin

_____

Joshua Goodman

Date: _____

In presenting this dissertation in partial fulfillment of the requirements for the doctoral degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to Proquest Information and Learning, 300 North Zeeb Road, Ann Arbor, MI 48106-1346, to whom the author has granted "the right to reproduce and sell (a) copies of the manuscript in microform and/or (b) printed copies of the manuscript made from microform."

Signature_____

Date_____

University of Washington

Abstract

Topic Learning in Text and Conversational Speech

Constantinos Boulis

Chair of Supervisory Committee:
Professor Mari Ostendorf
Electrical Engineering

Extracting topics from large collections of data is a crucial step to enhance information access. There has been an abundance of work on supervised topic learning methods on text, yet there are a number of directions in topic learning, that have received less attention, such as constructing feature spaces, unsupervised learning and dealing with different language genres. This dissertation addresses these issues and is concerned with topic learning in text and conversational speech.

In the first half of the dissertation, general approaches to topic learning are investigated. Algorithms to combine different partitions are suggested and evaluated on a number of text corpora, offering improvements compared to established baselines. In addition, a novel feature augmentation method is developed that adds to the bag-of-words representation, a small number of word pairs that exhibit a distinct pattern from their constituting words. The approach is evaluated on different corpora and the results show a consistent performance gain for a number of learning methods.

In the second half of the dissertation, issues that are relevant for topic learning in conversational speech are investigated. In the area of prosody, the studies involve prominence, i.e. loosely defined as phrase-level emphasis given to one or more syllables of a word. Experiments revealed that lack of prominence is an excellent indicator of low-salient words, using average word statistics from an automatic prominence detector. The role of disflu-

encies is investigated using hand-annotated self-corrections. The experiments reveal that removing disfluencies has little impact on topic classification when using the standard bag-of-words representation. Also, a quantitative analysis of lexical patterns between genders in conversations is conducted, revealing important differences, associated with the gender of the conversational partner. However, integrating gender information in a topic detection system did not improve the topic classification performance. Finally, the impact of the errors introduced by the automatic speech recognition (ASR) component is assessed. A method to cluster words according to a confusability measure derived from the ASR system is proposed and shown to offer performance gains compared to using 1-best transcripts and computational gains compared to using multiple ASR hypotheses.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

A PhD is a long and treacherous road and although it is mainly a personal feat, there are always people that critically influence its outcome. First and foremost, I would like to acknowledge my advisor, Professor Mari Ostendorf, for being supportive of my work, seeing the good sides of it when all I could see were the bad sides and pointing out the bad sides when all I could see were the good sides. Her meticulous reading of my work, attention to detail, insights that stem from decades of experience, and intelligence have definitely improved the quality of my research. Mari is a person who truly cares about her students and if I had to start a PhD all over again, there is no doubt in my mind that Mari would be the person I would choose to work with.

I also want to thank the other two members of the reading committee, Dr. Joshua Goodman and Professor Eve Riskin. Dr. Goodman's high standards of academic research have pushed me into not settling with the first positive results, but scrutinizing every experiment. Professor Riskin, provided prompt answers to my questions and an interesting perspective of this work. Special thanks go to Assistant Professor Ka Yee Yeung of University of Washington, Department of Microbiology for providing the gene expression data. I would also like to thank the National Science Foundation for funding this work through grant IIS-0121396. Any opinions or conclusions expressed in this dissertation are those of the author and do not necessarily reflect the views of this agency.

There are many people in the SSLI lab I would like to thank for a number of reasons. Scott Otterson for stimulating political or otherwise discussions, Arindam Mandal for his technical expertise, Jeremy Kahn for never being tired of talking about research, Becky Bates for improving my presentations (and pretty much everyone else's in the lab). You make SSLI a better lab.

It is unknown to me if this piece of work would have been made possible if it wasn't for

the encouragement of my wife. Zefi has seen me at my lowest, at my highest and anywhere in between. Being a PhD holder herself she knows a couple of things about the excitements and the disappointments you face along the way. Also, I cannot but acknowledge the critical support I had from my good friends Giorgos Arhonditsis and Giorgos-Yulis Papadopoulos. We all pursued PhDs at about the same time frames and I will cherish the wonderful moments we've spent together.

Last but not least, I want to thank my brother, Thanasis, for never losing his cool and providing me with a refreshing perspective. And, of course, my mother who taught me the value of education and was always pushing me to higher grounds.

Chapter 1

INTRODUCTION

Imagine the following scenario: People call a company's customer-support line to inquire about the product or service they have received from the company. The system understands the request of the customer and routes it to the appropriate department or even mines for main themes in the vast logs of customer requests. These themes are time-varying so that if a new trend comes up, e.g. a new defective product hits the market, the system is able to adaptively modify the themes. Support line operators can use these themes to ask more targeted questions to the customer and lead to an even quicker resolution of problems or improve the customer's experience. Capabilities for call-routing are already available, although such a system is still in the realm of science fiction. One of the underlying technologies that will make this a reality is topic learning, both general algorithms for topic learning and those specific to natural, spontaneous speech (with all the challenges and the opportunities that this entails).

Before delving into the details of topic learning, a definition of the term *topic* should be provided. This is not an easy task and certainly depends on the perspective of the person who is accessing the information. An article on the 2004 Iraq war can be relevant for a person who is seeking information on wars or information about the US foreign policy. There is also the issue of topic granularity. A nuanced technical discussion may appear as a single, uniform topic to one person and as many to another. In this dissertation, the topics are either prespecified, by providing a number of examples for each, or are unknown and learned automatically, in tasks where some amount of hand-labeled topics are available for performance assessment. In addition, the term *document* is interpreted to mean an atomic unit of input. This is an unfortunate choice of a name since in some cases there

is no notion of a document, but this is the term that is often used in the topic learning literature. Therefore, the word *document* can be used to refer to a news article, an email, a conversation or a segment of text.

1.1 What is topic learning?

Topic learning can span different paradigms (supervised, semi-supervised, unsupervised) and many media (text, speech, video or combination of these). Giving a definition of topic learning that encompasses all paradigms is not very informative. Instead, a short description of each one of the different paradigms is provided.

In supervised topic learning, a number of examples for each topic are available. The objective is to learn a function that maps a new document to one or more of the predefined topics. Methods that associate a document with multiple topics are more suitable for tasks such as characterizing news articles in terms of different events. For other tasks, such as call routing where the objective is to route a customer phone call to a specific destination, e.g. technical problem, bill question etc., methods that associate a single topic with a document are more suitable.

In semi-supervised topic learning, a few training examples are available as is a larger amount of documents with no topic information. The objective is to use the unannotated data in an effective way so that the mapping function of documents to topics is improved as much as possible, relative to using only the available training data. This is a paradigm that arises often in practice, where creating topically annotated data is an expensive and slow process, but large amounts of unannotated documents are available.

In unsupervised topic learning, there are no labeled training examples, only a large amount of documents. The emphasis here is different from the previous paradigms. The objective is to extract and describe the topics that are present, rather than learn a general mapping function from documents to topics. Unsupervised topic learning can be thought of as a clustering procedure so that documents on the same topic are grouped in the same cluster. The final objective is to generate a human-understandable description of the topics, such as outputting the top N features for each topic or by performing summarization.

The focus of this dissertation is on supervised and unsupervised approaches; semi-supervised approaches are mentioned here for completeness.

1.2 Why is topic learning important?

Topic learning is important because it can improve information access and extract information that was previously unattainable. Supervised topic learning can help arrange news articles or technical documents into thematic categories to facilitate retrieval and summarization, categorize emails as spam or not spam, and route phone calls to the correct destination, to name a few. Semi-supervised topic learning is applicable in almost all of the cases where supervised topic learning is. For example, we can bootstrap a system with a small number of training examples and then use the unannotated data to improve them. This is especially important when the topics are evolving over time. Then the models built for the topics should not remain static but should continue to adapt as new data become available. Unsupervised topic learning can help discover the concepts in a large set of documents and organize the results returned by a Web search query to groups (see for example http://www.vivisimo.com).

Companies such as Google and Yahoo! employ these technologies in the services they provide. For example, Google News or Yahoo! News monitor thousands of news sources, cluster articles on the same event and then classify the group to one of the predefined thematic categories. Other systems such as Columbia's Newsblaster take this a step further by performing multi-document summarization.

1.3 What are the challenges of topic learning?

The challenges in general learning problems apply in topic learning as well, for example how to find an optimum mapping function from documents to topics for supervised topic learning, or how to validate the clusters for unsupervised learning. However, the issue that sets topic learning apart from other learning problems is the huge dimensionality of the feature space. Even by using simple document representations, such as the bag-of-words where a document is represented by the count of words not the position or sequence, the

dimensionality is in the order of tens of thousands. In addition, many of these features are irrelevant for topic learning and others are redundant given a feature subset. Designing learning algorithms that are robust to the sparseness of data is crucial. For supervised learning, this would mean avoid overfitting to the training data. For unsupervised learning or clustering, sparseness of data results to partitions that can vary depending on the initial parameter values. Methods to provide stability of results are important.

1.4 What are the deficiencies of current approaches in topic learning?

A full discussion of contemporary approaches to topic learning is deferred to Chapter 2. Here, a very broad assessment of the research opportunities that exist in topic learning is provided. There has been a considerable body of work on supervised learning methods in text, e.g. decision trees, Naive Bayes, logistic regression, neural networks, maximum entropy models, support vector machines, k nearest neighbors, to name a few. In contrast, there has not been a lot of work on how to best represent a document for topic learning, e.g. feature extraction research. The prevailing assumption is that the bag-of-words representation is adequate and that other representations require even higher dimensionality that makes them impractical at best. In addition, semi-supervised and unsupervised paradigms have only recently started to attract attention. Especially in unsupervised topic learning there are challenges that are similar to some of the challenges in supervised topic learning but require quite different solutions. Examples include combination of clustering systems and feature selection. Finally, topic learning can span different media but the majority of the work has focused on text. New media bring new research opportunities. For example, spoken language is very different from written language. Features that are useful in text may not be useful in speech and vice versa. In addition, speech is a richer medium than text, containing not only *which* words have been uttered but also *how* they were uttered.

1.5 Contributions of the dissertation

There are six main contributions of this dissertation. They can be divided in two parts; the first part contributes to general methodologies for clustering, applicable on many media and

assessed here using text, speech and gene expression data. There are two main contributions in the first part:

- **Algorithms to combine different clustering outputs**. Despite the fact that there has been a lot of work, both theoretical and empirical, on combining classifiers, there has not been much work on combining clustering systems. A major complication with combining clustering systems is that there is no obvious correspondence between clusters of different systems. Cluster 1 of system 1 may correspond to cluster 7 of system 2. If the correspondence problem is tackled, then many techniques similar to classifier combination can be applied. In this dissertation, different methods to estimate the cluster correspondence are presented and evaluated. Results show that in some cases, improvement can reach 40% relative to using a single clustering system.

- **Augmenting the bag-of-words representation with selected word pairs**. The bag-of-words representation is the most popular document representation method used for text classification/clustering. Although different variations exist, such as using different weightings and normalization schemes, the basic assumption is that a phrase does not contribute more than the sum of its parts. In this dissertation, a method is presented that selects word pairs that are topically more discriminative than the individual words treated as unordered. This feature augmentation method is applicable on the supervised topic learning paradigm and was shown to consistently offer gains over a range of different classifiers and corpora.

The second part contributes algorithms that are specifically designed for topic learning in conversational speech, which has been much less studied for topic classification than written text. In conversational speech, issues such as separating content from style, the role of disfluencies, exploiting prosody and automatic speech recognition (ASR) errors come into play. Since topic learning in conversational speech has been explored less than written text, it occupies a larger portion of this dissertation than the first part. The contributions related to the second part are:

- **A quantitative analysis of gender lexical differences in telephone conversations**. Can we model style in a conversation and if so how can we exploit it to improve topic learning? The experiments reported here show that there are distinct lexical differences between genders in conversations that are not accounted for by topic or speaker and also that the gender of one speaker will influence the lexical usage pattern of the other speaker. These results show that the issue of style at the gender level is very much present in conversational speech. Simple techniques to leverage style for topic learning, such as training gender-dependent topic models, have not yet proven successful but more sophisticated modeling techniques may offer benefits.

- **Investigating the effects of disfluencies in supervised topic learning on conversational speech**. Another distinct phenomenon of conversational speech is the abundance of disfluencies. Although at the surface disfluencies appear to interrupt the flow of information, human listeners typically have little trouble understanding disfluent speech. For automatic language processing though, disfluencies falsely increment the counts of words, and since the most prevalent representation for topic classification is the bag-of-words, this raises the question of whether they can have an adverse effect on conversation classification. In this dissertation, it is shown that removing disfluencies has a small but consistent effect on supervised topic classification and that all types of disfluencies contribute. Experiments also indicate that representations other than bag-of-words, such as bag-of-word-pairs, can benefit more from removing disfluencies.

- **Using prominence to select feature subsets for supervised and unsupervised topic learning on conversational speech**. A distinct characteristic of speech is that it carries more information than the identity of the words. *How* words are pronounced can be very useful in addition to *which* words are pronounced. The degree a word is emphasized is defined as the prominence of the word. A prominence classifier with about 20% error is used to annotate each word on a large number of conversations as being prominent or not. The experiments reveal that combining measures of

prominence with lexical saliency measures can offer improved topic classification performance over using lexical saliency measures only. In addition, prominence measures can also be used in unsupervised topic learning tasks, and gains over standard feature correlation methods are reported as well.

- **A method to cluster words based on confusability derived from an automatic speech recognition system**. Confusion networks are the result of aligning multiple hypotheses of an ASR system so that at position k all the competing word alternatives appear. This framework allows the calculation of word co-occurrence statistics, so that words that are highly correlated can be collapsed to a single token. By using agglomerative clustering, the vocabulary can be reduced so that the tokens appearing are less confusable. Experiments show that this step produces word clusters that offer the same topic classification gains as word clusters constructed with stemming. An advantage of this approach is that it may provide a language-independent stemming procedure, avoiding the cost of manually building a stemmer for resource-impoverished languages.

1.6 Dissertation overview

This dissertation is structured as follows: A review of background material related to topic learning approaches in text and speech is presented in chapter 2. Chapters 3 and 4 comprise Part I of the dissertation, on general methodologies for topic learning. In chapter 3, a number of algorithms to combine multiple clustering partitions are proposed and applied to improve clustering performance and as cluster validation measures. In chapter 4, a procedure that augments the bag-of-words representation with selected word pairs is presented. Part II of the dissertation, on topic learning in conversational speech, is covered in chapters 5, 6, 7, 8. In chapter 5, gender lexical differences in conversations are analyzed. In chapter 6, the impact of disfluencies on topic classification performance of conversations is assessed. In chapter 7, the role of prominence for designing feature subsets is detailed. In chapter 8, a method to cluster words according to ASR confusability is presented. Finally, in chapter 9, a summary and future directions are presented.

Chapter 2

BACKGROUND

In this chapter, a review of the main approaches in topic learning in text and speech is presented. All the different aspects that are associated with topic learning are considered: document representations, learning methods and evaluation measures. The main emphasis is on supervised and unsupervised approaches, since the dissertation builds on these, but semi-supervised approaches are also mentioned for completeness. In addition, a review of past work relevant to topic learning in speech is also presented.

2.1 Document Representation

An issue that is central in every topic learning paradigm is how to represent a document in a way that all the relevant topic information is captured. Currently, the most common representation is the bag-of-words. Under the bag-of words representation, a document is represented as a vector of dimension V, where V is the vocabulary size. Dimension i contains the number of times word i has appeared in the document. A common variation of this representation is the tf·idf family of representations. The most common member of the tf·idf family is given by equation (2.1), where dimension i is given by:

$$d_i = \frac{c_i}{\sum_{k=1}^{V} c_k} * \log(\frac{N}{f_i}) \tag{2.1}$$

where c_i is the number of times word i has appeared in the document, N is the total number of documents and f_i is the number of documents word i has appeared in. The left part of the product of equation (2.1) is the term frequency (tf) and the right part is the inverse document frequency (idf). The right part has the effect of deweighting words that occur across many documents. Therefore, words such as *and*, *it* and *the* that are very common will have a high tf but an idf that is near zero; therefore, their effect will be negligible. There are numerous other variants of the tf·idf measure using different normalizations [134].

The bag-of-words representation and its variants make the assumption that only the count of words and not the position or sequence of words are sufficient for capturing the topic information. This naive assumption treats language as lacking any syntactic, discourse and pragmatic structure. Although it is easy to construct examples where this representation fails (consider the two sentences *"This is about cars not birds"* and *"This is about birds not cars"* which have identical bag-of-words representations), researchers have not yet been successful in replacing it with a better alternative, for the purposes of topic learning. Usually, a stemming procedure [132] is also applied, collapsing all inflectional variants of a word to a single token, e.g. *run*, *running*, *runner*, *runners* are all collapsed to the token *run*. If stemming is applied, then each feature does not represent a word, rather a set of words, but since this is not a major change from the computational perspective it will still be referred to as the bag-of-words representation.

Over the past, a number of attempts have been made to augment or substitute the bag-of-words representation with richer features. In [100, 53], linguistic phrases, proper names and complex nominals are used, and in [68] noun phrase heads and proper names are used for topic clustering. In [160, 133], bigrams are added to the feature space. In [131], character n-grams are used for text classification. A recent comprehensive study [118] surveys the different approaches that have been taken thus far and evaluates them on standard text classification resources. The conclusion is that more complex features do not offer any gain when combined with state-of-the-art learning methods, such as Support Vector Machines (SVM). A fundamental challenge with more complex approaches is that they usually require a much higher dimensionality. Since V is on the order of tens of thousands, further increasing the dimensionality may aggravate the learning problem by adding redundant and/or irrelevant features, as well as relevant features that contribute new information.

Machine learning methodologies to construct feature subsets differ depending on whether we operate on a supervised or unsupervised setting. For supervised learning, there are three main approaches: the filter approach [83], the wrapper approach [89] and the embedded approach [65]. The filter approach scores features independently of the classifier and often independently of each other. Filter feature selection approaches such as [90, 98] attempt to

find a feature subset that is maximally relevant with minimal redundancies between features. These algorithms are of order $O(V^2)$, where V is the original number of features, and can therefore be infeasible to apply for large V. There are also a number of filter feature selection algorithms that address the relevance of individual features that are of $O(V)$ complexity and are the ones that are used mostly in practice. A survey of such algorithms can be found in [49]. The wrapper approach jointly computes the classifier and the subset of features and it is arguably the optimum approach but since it involves a re-training of the classifier for different feature subsets it is impractical for all but small values of V. The embedded approach, combines the filter and wrapper approaches into one by embedding a filter feature selection method into the process of classifier training, rather than treating the classifier as a black box. Designing embedded feature selection approaches may not always be feasible. For some learning methods it may not be possible to formulate an embedded approach. Since high-vocabulary topic learning task are the focus of this dissertation, only filter feature selection approaches are selected.

One of the baseline filter feature selection methods that was implemented in this thesis is the Information Gain (IG), since it has been shown before [49] that is one of the best performing methods. The IG measure is given by:

$$IG_w = H(\mathbf{C}) - p(w)H(\mathbf{C}|w) - p(\bar{w})H(\mathbf{C}|\bar{w}) \tag{2.2}$$

where $H(\mathbf{C}) = -\sum_{c=1}^{C} p(c)\log p(c)$ denotes the entropy of the discrete topic category random variable $\mathbf{C}$ and $\bar{w}$ denotes the event "*not* w". Each document is represented with the Bernoulli model, i.e. a vector of 1 or 0 depending if the word appears or not in the document. Note that the IG measure takes into account the absence as well as the presence of a feature.

For unsupervised learning, the notion of relevance is not applicable since there is no topic information. Relevance is substituted with other criteria, such as separation of clusters [44] that require a clustering method. Therefore, for the unsupervised setting there are no filter feature selection methods. Common methods for unsupervised feature selection are to combine highly dependent features ([116, 150]).

2.2 Supervised Topic Learning

The goal of supervised topic learning is to learn a function that maps a document to one or more topics, given a number of examples for each topic. More formally, given n tuples $(\vec{d_n}, \vec{t_n})$, where $\vec{d_n}$ is the vector representation of document d_n and $\vec{t_n}$ is a vector with the topics that are associated with d_n, the goal of supervised topic learning is to learn a function $\Phi : \Re^V \rightarrow \{0,1\}^T$, where V is the dimensionality of the input space and T is the number of topics. If more than one topic is output per document then we have a multi-label paradigm, else if exactly one topic per document is output then we have a single-label paradigm.

Estimating the function Φ in its most general form with large values of T can be a challenge, since there are 2^T distinct outputs. Therefore, a number of assumptions are usually taken, the most common of which is to learn T binary functions $\Psi_t : \Re^V \rightarrow \{0,1\}$, $t = 1 \ldots T$. A new document will be evaluated for all T functions and a separate decision will be made for each. Therefore, the number of distinct outputs that need to be learned have been reduced from 2^T to $2T$. In order for this classification strategy to be optimal, the topic binary random variables must be independent of each other. For single-label classification, a function Φ is estimated such that $\Phi : \Re^V \rightarrow \{1, \ldots, T\}$. It should be noted that a single-label classifier can be constructed with binary classifiers and a voting mechanism. Also, a multi-label classifier can be constructed using a single-label classifier with soft outputs.

2.2.1 Methods

Estimating the functions that map documents to topics, either multi-label or single-label, is carried out using a number of classification methods, such as Naive Bayes [109], Support Vector Machines (SVMs) [79], Maximum Entropy [122], Rocchio [78], probabilistic indexing [54], decision trees [106] and k-nearest-neighbors (kNN) [106] to name a few. These classifiers are applicable to general situations but they have been found to be successful in the topic classification context as well. The tradeoffs that are associated with these classifiers in general settings apply in the topic classification setting as well. For example, SVMs, Naive Bayes and Maximum Entropy are all linear classifiers (although SVMs can learn a non-linear boundary with the appropriate choice of kernel), while decision trees and kNN are

not. Decision trees are not appropriate when a large number of features are needed and each can make a useful contribution to classification, but they can learn concepts that linear classifiers can not (such as the XOR concept). A caveat that applies in topic classification more than other classification problems is that, due to the high dimensionality of the input space, it is easy for a classifier to overfit. This is one reason why low-variance, high-bias classifiers such as Naive Bayes are often shown to be particularly effective. See [140] for a more detailed survey in supervised topic learning methods.

The common strategy behind all these models is to use multiple single-label classifiers to achieve multi-label classification. In [138, 108, 165] a more direct approach is taken where a document is associated with a subset of topics and each word is assumed to be generated from a different topic selected from the subset.

2.2.2 Evaluation

Since a multi-label topic classifier is a system that returns a set of topics given a document, information retrieval evaluation measures are applicable. In a multi-label setting, T 2×2 contingency tables are constructed, each for a topic. If H_k is the set of topics that document d_k, is hypothesized to map to, and C_k is the set of topics that are known to be associated with document d_k, then the four entries of each contingency table are given by:

$$TP_t = \sum_k I(t \in H_k \; and \; t \in C_k) \tag{2.3}$$

$$FP_t = \sum_k I(t \in H_k \; and \; t \notin C_k) \tag{2.4}$$

$$FN_t = \sum_k I(t \notin H_k \; and \; t \in C_k) \tag{2.5}$$

$$TN_t = \sum_k I(t \notin H_k \; and \; t \notin C_k) \tag{2.6}$$

where $I(\cdot)$ is the indicator function, i.e. it is zero when its argument is false and 1 otherwise. Given a contingency table we calculate two quantities, precision and recall. Precision is the fraction of hypothesized topics that are correct, and recall is the fraction of correct topics that are hypothesized. Therefore, $P_t = \frac{TP_t}{TP_t+FN_t}$ and $R_t = \frac{TP_t}{TP_t+FP_t}$. A perfect classifier will have $P_t = R_t = 1$, but a trivial classifier that hypothesizes every single topic for all

documents will have $P_t \approx 1/T$ and $R_t = 1$. The F-measure is used to combine precision and recall into one measure, and is given by $F = \frac{2PR}{P+R}$.

There are two main measures that are used for evaluating multi-label supervised topic learning systems; the microF and the macroF. To calculate the microF, the contingency tables for all topics are added together and precision and recall are computed on the joint table. To calculate the macroF, precision and recall are computed for each topic and then averaged. Therefore for microF we use:

$$microP = \frac{\sum_{t=1}^{T} TP_t}{\sum_t (TP_t + FP_t)} \tag{2.7}$$

$$microR = \frac{\sum_{t=1}^{T} TP_t}{\sum_t (TP_t + FN_t)} \tag{2.8}$$

While for macroF we use:

$$macroP = \sum_{t=1}^{T} (\frac{TP_t}{TP_t + FP_t}) \tag{2.9}$$

$$macroR = \sum_{t=1}^{T} (\frac{TP_t}{TP_t + FN_t}) \tag{2.10}$$

MacroF gives equal weight to all topics, independent of the frequency of their occurrence. Therefore, if a classifier tends to operate poorly for a less common topic, macroF will be lower than microF. MicroF, on the other hand, will be influenced more by commonly occurring topics.

Evaluating single-label systems is done using classification accuracy:

$$Acc = \frac{\sum_k I(\Phi(\vec{d_k}) = t_k)}{K} \tag{2.11}$$

where k is the index through documents, K is the number of documents in the test set, and t_k is the correct topic label for document k.

2.3 Semi-Supervised Topic Learning Methods

This dissertation does not address the problem of semi-supervised topic learning, but for completeness of the literature review, a brief survey of work in the area is provided. A general problem with supervised learning is that producing large amounts of labeled data

is an expensive and slow process. In many cases, the supervision provided will be in forms other than the labels of a large number of documents. In [124, 7], a small number of labeled documents are available to bootstrap the system, and then large amounts of unlabeled data are leveraged to improve the effectiveness of the system using the EM algorithm [36]. In [81], an extension to the SVM algorithm to handle unlabeled data is presented and applied for text classification. In [25], topic language models are built using initial labeled data and then queries to the Web are issued to retrieve documents that are similar to the current models. The retrieved documents are then treated as unlabeled data and used to re-train the models. In [139], Wordnet is used to extract hypernyms and provide a more robust term-level comparison.

In [168], no labeled data exists, but supervision is provided in terms of *must-link* and *cannot-link* constraints. The *must-link* constraints are a set of pairs of documents that must be in the same class, while the *cannot-link* set contains pairs of documents that cannot be in the same class. A formulation with probabilistic constraints is given in [97]. In [87, 8, 16, 173] a metric is trained that satisfies the constraints. In [60], clustering is pursued under the constraint that the discovered partition must be as different as possible than a given partition. In [110], the topics are bootstrapped with keywords, while in [58] a combination of keyword-bootstrapping and active learning for selecting a few documents to be annotated by the user, is pursued.

2.4 Unsupervised Topic Learning

In unsupervised topic learning, there are no examples for a topic and often even the number of topics is unknown. The objective of such methods is to group together topically similar documents. There are two ways that unsupervised topic learning methods can be useful. They can be used to improve supervised learning models or as an exploratory data analysis tool to help uncover systematic trends previously unknown.

In statistical modeling clustering can be used in three ways: to improve supervised learning models for determining parameter sharing, for learning hidden structure and for computational gains. The first way is to reduce the variance and increase the bias of the

model by clustering together similar documents or words and training shared parameters on the clusters. This is suitable only when the variance of the model is significantly larger than the bias, therefore such approaches are more likely to be successful under sparse data conditions. Examples of this approach are cluster-based text classification methods [151, 38], class-based language models [24] or combinations of $n-$gram and class-based language models [61] that can control the tradeoff between variance and bias in a more detailed way. A second way is to make the model more complex by uncovering hidden modes in the data. This is suitable when there are a lot of data and robust decisions can be made. An example of such approaches is the topic-dependent language model [75, 174, 141, 57, 13, 25]. A third way is to use clustering for reducing the execution time of existing supervised methods. For example, in [63] and [62] word clusters are shown to significantly speed-up the training of language models.

As an exploratory data analysis tool, unsupervised topic learning can help uncover the underlying topics in large collections of documents [39, 19, 149, 15]. The emphasis here is to extract meaningful topics and present them to the user, usually by displaying a set of words for each topic. An important research question that has not yet been adequately addressed is how to present the description of the clusters found to the user. Words, phrases, documents or paraphrasing text are some options.

2.4.1 Methods

Unsupervised topic learning methods, just as supervised, can be single-label or multi-label. Single-label methods are referred to as clustering methods and they usually produce a hard decision, i.e. partition a set of documents to non-overlapping subsets. There are three main families of methods that are used to cluster documents into groups.

The first family includes hierarchical clustering algorithms [77], agglomerative or divisive. Agglomerative clustering starts from placing each document in a separate cluster. At each step the pair of clusters with the highest similarity is merged; the similarities between the new cluster and every other cluster are updated; and the number of remaining clusters is decremented by one. To compute the similarity between two clusters three alternatives

are provided: choose the minimum, maximum or average similarity between documents in different clusters. Conversely, divisive clustering starts from placing every document in a single cluster. At each step, two decisions are made: select which cluster to split and how. For both agglomerative and divisive hierarchical clustering, the process repeats until a specified number of clusters is reached or an automatic measure is met, e.g. the maximum similarity is above/below a threshold.

The second family includes partitional clustering algorithms [77, 182]. Unlike hierarchical clustering, partitional algorithms produce a partition of K clusters without first generating a partition to K-1 or K+1 clusters. Probably the most popular partitional based algorithm is the k-means algorithm [104], which attempts to minimize the objective function:

$$\hat{g} = \arg\min_{g(\cdot)} \sum_{j=1}^{N} ||\vec{d_j} - \vec{c}_{g(\vec{d_j})}||^2 \tag{2.12}$$

where $||\cdot||^2$ is the Euclidean distance and $\vec{c_i}$, $i = 1 \dots K$ is the centroid of cluster i. Global minimization of this function is known to be NP-hard but efficient techniques that find local optima are known. The technique that k-means uses is iterative. Starting from an initial assignment of documents to clusters (which can be random), the algorithm iterates between computing the centroids of the clusters and assigning documents to clusters. This scheme is known to converge to a local minimum of equation (2.12). Recently, there have been a number of extensions to the k-means algorithm. One extension is to use kernels instead of Euclidean distances [136], thus being able to find clusters that are non-linearly separable. Another extension is to assume a generative probabilistic model with multiple modes and attempt to fit the model to the unlabeled documents [50, 114]. The modes of the distribution correspond to different clusters in the data. These methods are known as model-based methods and training is carried out using the EM algorithm [36]. For example, the k-means algorithm can be shown to be identical to the Viterbi analog of the EM training of generative probabilistic model with the following assumptions: a) the probability distribution that is fitted on the data is a mixture of Gaussians, with b) diagonal unity covariance matrix multiplied by a scaling factor σ^2 and shared by all components of the mixture. An advantage of model-based methods is that model selection techniques can

be applied to find the number of clusters in the data [28] or to jointly cluster the documents and find the most important features [96].

The third family is graph-based algorithms. A graph $G = (V, E, A)$ is used to represent the documents and their interactions. V is a set of vertices, each vertex represents a document, E is the set of edges and A is a $|V| \times |V|$ matrix with the weight of each edge. If an edge does not appear then the corresponding entry in A will be zero. An edge between two documents represents their similarity. A number of different objective functions [37] are used to partition the graph into subgraphs. Spectral methods [119] have been used to optimize some of the objective functions. The minimal spanning tree algorithm [55] has also been used as a way to find tightly connected subgraphs.

Multi-label unsupervised topic learning is a rather recent research topic [130, 71, 149, 19]. In [130, 71] a probabilistic extension of the Latent Semantic Analysis is given where a document can simultaneously belong to more than one latent factors. A similar model appears in [149] where a generative probabilistic model is assumed, that first samples a subset of topics for a document, then repeats the following two steps K times (K is the number of words in the document): i) samples a topic from the subset, then ii) sample a word from the topic. The work in [19] further extends [71] by using a Dirichlet prior for the topic distribution, reducing the number of free parameters.

2.4.2 Evaluation

There are two main categories of methods to evaluate unsupervised topic learning systems. The first relies on the availability of topic(s) that are associated with each document and evaluates how well the clusters map to the topics. Techniques in this category include accuracy based on a mapping of clusters to topics using maximum bipartite matching [93], normalized mutual information between clusters and topics [158], and the adjusted rand index [73].

Maximum bipartite matching can be used to find the one-to-one mapping of clusters to classes. The problem of finding the optimum assignment of M clusters to M classes can be formulated and solved with linear programming. If $n_{i,j}$ is the number of documents that are

assigned on cluster i and class j, $\lambda_{i,j}$=1 if cluster i is assigned to class j and 0 otherwise are the parameters to estimate, then we seek to find: $\max_{\lambda_{i,j}} \sum_{i,j} n_{i,j}\lambda_{i,j}$ under the constraints $\sum_i \lambda_{i,j} = 1$ and $\sum_j \lambda_{i,j} = 1$. The constraints will ensure a one-to-one mapping. Accuracy is then defined as:

$$Acc = \frac{\sum_{i=1}^{M} n_{i,\hat{j}}}{D} \tag{2.13}$$

where $\hat{j} : \lambda_{i,\hat{j}} = 1$ is the class that cluster i maps to, and D is the total number of documents. This evaluation measure is most meaningful when the number of clusters is equal to the number of classes.

Another evaluation measure that is used in this dissertation is the normalized mutual information (NMI) between clusters and classes. The measure does not assume a fixed cluster-to-class mapping but rather takes the average mutual information between every cluster-class pair. It is given by:

$$NMI = \frac{\sum_{i=1}^{M}\sum_{j=1}^{M'} n_{i,j} \log\left(\frac{n_{i,j}D}{n_i m_j}\right)}{\sqrt{\sum_{i=1}^{M} n_i \log\frac{n_i}{D} \sum_{j=1}^{M'} m_j \log\frac{m_j}{D}}} \tag{2.14}$$

where $n_{i,j}$ is the number of documents cluster i and class j agree, n_i is the number of documents assigned to cluster i, m_j the number of documents of class j, and D is the total number of documents. It can be shown that $0 < NMI \leq 1$ with NMI=1 corresponding to perfect classification accuracy.

A third evaluation measure used in this dissertation is the adjusted Rand index [73], a measure similar to NMI that is broadly used in evaluating partitions in gene expression data. The adjusted Rand index is a modification of the Rand index which is the ratio of pairs of points that have been correctly clustered together and correctly not clustered together over all pairs of points. The problem with the original Rand index is that its expected value is not zero when the partitions are chosen at random. For more information on how to calculate the adjusted Rand index the reader is referred to [178]. Both NMI and the adjusted Rand index can be used even when the number of clusters is not equal to the number of classes. For all 3 measures, a higher score is better.

The second category of evaluation methods, is to present a description of each cluster to the user and let him/her evaluate the quality of the cluster. For example, in [149] a set of

words for each topic are presented. The user can evaluate recall in these lists but evaluating precision would mean that all the relevant words for a topic must be available, a task that is not straightforward [149]. This method has the advantage of not requiring annotated data, but the disadvantage that the use of human judgments is costly and not practical in system development.

2.5 Topic Learning in Speech

The methods that were covered in the previous sections can be applied to a broad range of inputs, since the proposed methods do not exploit the specific opportunities that new genres provide. In this section, a survey of applications and methods pertaining to topic learning in speech is presented. Some of the methods that handle informal style and presence of errors may also be extended to other sources, such as transcripts of Internet chats.

2.5.1 Application areas and general approaches

There are a number of tasks that directly or indirectly rely on topic learning in speech, such as call-routing, topic categorization of conversations, and spoken document retrieval. In call-routing, a person places a phone call, describes the intent of the call in a natural, unconstrained way and is then routed to the appropriate department. For example AT&T's How May I Help You^{TM} system [64] routes phone calls to one of 15 categories, such as *credit* and *area code*. Call-routing is therefore a single-label classification task. The vector-space representation is adopted for call-routing in [31], using stemmed words, and in some occasions hand-selected bigrams and trigrams. Standard feature selection/combination techniques are applied [101], and the final feature representation is refined with human expertise. A number of learning models and training methods (generative or discriminative [85]) have been used for call-routing, such as Naive Bayes, Maximum Entropy [29] and boosting-based algorithms [183, 135]. Lately, the emphasis has been placed on semi-supervised approaches using ASR transcripts, since obtaining human-generated transcripts is the most expensive and time-consuming step. Approaches that continue to update the classifier using unannotated and untranscribed data [74] and/or select data [66] have been popular.

Topic learning in speech has a prominent role in a number of other projects as well. Most recently in the MALACH project [26], a project with the goal of searching through multilingual oral history archives. The corpus consists of 120,000 hours of oral interviews from survivors of the Holocaust. One of the goals was to categorize segments of interviews to M-of-N prespecified topics (multi-label classification, N=323, M variable). ASR transcripts are used as the input and two classifiers kNN and Maximum Entropy are evaluated with kNN achieving superior results. Overall, the topic categorization performance is low (microF is about 0.20-0.25), but the ASR module is not the main factor behind the low performance. The relative degradation of text categorization performance between the human and ASR transcripts with 42% WER was only 15%. But when ASR transcripts with 52% WER were used the relative degradation was 28% which shows that after a certain threshold topic categorization performance drops sharply.

Another project that shares a number of issues with topic learning in speech is the Topic Detection and Tracking project (TDT) (http://www.nist.gov/speech/tests/tdt/). Although most of the work focuses on text data coming from online sources, a sizable portion is directed towards broadcast news from television or radio. An important problem there is topic segmentation, i.e. detecting boundaries between topically distinct segments. For example, in a 45-minute political roundtable discussion, people may discuss a number of different topics but not explicitly indicate the boundaries between them. The most popular approach towards this problem is to use a combination of discourse and topical cohesion features [10]. An interesting perspective on this problem is to augment the model with prosodic cues to segmentation [163, 145, 99], which were shown to be very helpful. Another direction is to investigate the effects ASR errors have on topic segmentation and detection [112].

Topic learning in speech is also relevant for the Spoken Document Retrieval project [56], which is concerned with the retrieval of spoken documents that are related to a textual query. The spoken documents may include broadcast news shows and debates from television and radio. There are a number of issues that are shared between spoken document retrieval and topic learning in speech, such as the bag-of-words representation and approaches to handle ASR errors. The majority of the system submissions for past evaluations [56] followed the

modular approach, where the output of a standard ASR module is the input to standard information retrieval modules. An example of a commercial application of spoken document retrieval technology is SpeechBot[1] [161], indexing tens of thousands of hours of speech from various talk shows on television and radio.

2.5.2 Handling ASR errors

An important issue in topic learning in speech is that the ASR system that is employed to convert speech to text introduces a number of errors. As of 2005, typical word error rates for conversational speech are around 15% [127], although they can be higher if the acoustic conditions degrade, e.g. multi-speaker overlap or highly disfluent speech [155]. As validated from broadcast news to conversational speech, the relationship between ASR word error rate and topic classification/clustering performance is not linear. For example, retrieval performance between 0% (true transcripts) and 30% word error rate was not statistically different [56]. When the word error rate rose to 50% then retrieval performance seriously degraded [56]. In addition, not all ASR errors have the same impact on topic learning. For example, proper nouns tend to be out-of-vocabulary (OOV) words for an ASR system, but also tend to be important for capturing an emerging news events. Correctly recognizing proper nouns is more important than other parts of speech, such as function words. A number of studies are available that attempt to better integrate the ASR module to high-level spoken language processing tasks, such as call-routing or spoken document retrieval. These approaches can be divided into two broad categories. The first category is the set of techniques that modify the ASR system to output subword sequences or individual keywords. The second category is the set of techniques that use word confidences and/or alternative hypotheses of an ASR system that outputs word sequences. A discussion on the different issues associated with a spoken retrieval system can be found in [103].

Examples of the first category are [121, 2, 12, 126, 113, 27]. In [121], n-grams of various subword units (phones, syllables, broad phonetic classes) are investigated for use in a spoken document retrieval system, as a possible way of representing out-of-vocabulary (OOV) words

[1] http://speechbot.research.compaq.com/

and automatically removing the common morphological and inflectional parts from words. In [2], phone n-grams are used for call-routing classification combined with unsupervised learning, while in [12], the goal is to learn the subword units as well. In [126], phone n-grams are used as features for representing spoken documents. A dynamic programming procedure is applied that allows for approximate matches to account for pronunciation variability. In [113, 171, 48], a word spotter is used. In a keyword spotter system, speech is searched for a number of specific words. Using a word spotter is an intermediate approach between word sequence ASR and subword n-grams. The rationale behind using a word spotter is that for information retrieval/extraction common words are excluded from document representations and therefore do not need to be correctly recognized. All these methods, although shown to be moderately successful some years ago, are less popular by now due to significant advances in large vocabulary ASR (e.g. conversational speech or broadcast news). Subword-based approaches rely on subword recognizers (e.g. phone) that are known to have higher error rates than modern word recognizers. Furthermore, if the classification error of a subword recognizer is ϵ, then the classification error of n-grams of subwords will be $1-(1-\epsilon)^n$ (assuming independence of errors between consecutive subwords). For a subword recognizer with 30% error this would result in a 75% classification error of 4-grams of subwords. Even with human transcriptions available, subword n-grams did not produce better results than word unigrams, as suggested by the results in [27]. However, the approaches in the first category (subword-based or keyword-based) can be more robust to unknown conditions. A heavily mismatched language model can introduce the wrong type of constraints and a phone-based approach can be applied to other languages without requiring a language-specific ASR module.

Examples of the second category are [146, 120, 128, 30, 164]. In [146], a scheme is devised to moderate the effect of insertions, where each word of the 1-best hypothesis is weighted by an estimate of the confidence that the hypothesized word is correct. Confidence is estimated using the lattice density for the current word, i.e. how many competing hypotheses exist for a given word. It is shown that this heuristic weighting scheme can recover about 20% of the retrieval performance lost when moving from error-free text to text with 36% WER. In [128], word confidences for the 1-best output are used to better detect named entities by

allowing the algorithm to choose whether or not to ignore a word based on ASR confidence and word context. It is shown that the approach offers significant improvements over the baseline systems of no confidence scoring. These two schemes, although shown effective, can only moderate the effect of insertions and substitutions but cannot compensate for deletions. In [120], a scheme that assigns confidences according to the frequency of each word in a N-best list is proposed. This scheme can compensate for both insertions and deletions, but the confidence estimation technique was not shown to be robust. The gains in retrieval performance are moderate, and the system deteriorates fast if more than the top 5 hypotheses are used. In [30], the word confusion matrix is constructed by aligning the hypothesized word transcriptions with the true transcripts. The word confusion matrix is then used for query expansion and improved results are observed over a baseline of using the 1-best hypothesis. The word confusion matrix can model the type of errors of the ASR module, but the true transcripts are needed, making this method impractical in many scenarios. Finally, in [164], confusion networks are used to obtain confidences for alternative ASR hypotheses and can be easily integrated on classifiers based on the bag-of-words representation.

The contributions of this dissertation are mainly on issues other than handling ASR errors that have not received a lot of attention in the past. There is a considerable body of linguistics literature that links prosody with various levels of information structure, but a limited number of them have been explored from an engineering perspective (an exception is topic segmentation in broadcast news). In addition, linguistic phenomena abundant in conversational speech, such as disfluencies, have not been investigated for the purposes of improving access to spoken information.

PART I
GENERAL METHODOLOGIES FOR TOPIC LEARNING

The first part of the dissertation is concerned with topic learning methodologies that are designed to accommodate a range of possible types of input. The suggested methodologies are applied in different genres of text - news articles, newsgroup postings, web pages and transcripts of conversations - and in non-text input such as gene expression profiles.

In Chapter 3, unsupervised topic learning is pursued. A number of methods are presented that combine multiple clustering partitions for improved performance and robustness. In addition, the proposed methodologies can be used for cluster validation, i.e. assessing the correctness of discovered clusters. In Chapter 4, a feature augmentation method for supervised topic learning is presented. The bag-of-words representation is augmented with selected word pairs that contribute much more than their constituting words. The emphasis here is to find features that contribute to better performance across multiple classifiers and genres of text.

Chapter 3

COMBINING MULTIPLE CLUSTERING PARTITIONS

Over the past years, different clustering algorithms have been proposed that yield different partitions of the same dataset. Moreover, some algorithms, like k-means, produce different partitions if initialized differently. The large number of alternatives raises the possibility of leveraging results by consolidating the different partitions into one. In this chapter, we present different methods to combine hard or soft partitions using either linear programming techniques or singular value decomposition, leading to both increased performance and robustness for a variety of underlying clustering algorithms. In addition, we show that the degree of similarity of corresponding clusters is a measure of cluster validity. The cluster-specific measure can be used to output a few good clusters rather than making a decision for every object. Experiments are reported on three different tasks, electronic newsgroup postings, conversational speech and gene expression data.

3.1 Introduction and Problem Definition

Clustering, i.e. the task of grouping similar objects together, is an important tool for organizing or mining vast amounts of data for useful information. Most of the clustering approaches that have been proposed so far [77] have focused on a number of important issues such as the criteria to optimize (e.g. the intra-cluster distance for k-means or the normalized cut for graph-partitioning algorithms) and the methods to optimize or approximately optimize these criteria (e.g. the EM method [36] for k-means and the generalized eigenvalue method for the normalized cut [142]). Other important issues include the choice of distance measures [5], feature weighting and selection [96]. The wealth of different approaches will result in a wealth of different clustering partitions on the same dataset. Moreover, in some cases, seeding an algorithm with different initial conditions will produce different partitions, such as model-based clustering algorithms trained with EM. A natural question that arises

is how to combine all the different partitions into a single one.

More formally, we define as $\mathcal{X} = \{\vec{x}_1 \dots \vec{x}_D\}$ a set of D vectors to be partitioned in a number of C clusters. Each clustering system $s \in \{1, \dots, S\}$ estimates a function $g_s(\vec{x}_d) \in \{1 : C_s\}$ that takes as input an object $\vec{x_d}$ and outputs the partition label it is mapped to. Notice that the definition does not require the same number of clusters for each system and can also be extended to soft clustering systems by having the function $g_s(\cdot)$ output a vector of cluster membership probabilities (or weights) instead of a scalar. The objective of the clustering combination method is to estimate a function $\Phi(g_1(\vec{x_d}), \dots, g_S(\vec{x_d})) \in 1 : C$ that takes as input the output of each one of the clustering systems and outputs a final choice for object $\vec{x_d}$. The function $\Phi(\cdot)$ uses the output of the clustering systems only and not the object representations $\vec{x}_d$, allowing the clustering combination process to be detached from the object space. This offers three main advantages to the combination process. First, it can be applied in the same way for discrete or continuous attribute problems. Second, it allows the individual clustering algorithms to be run in parallel. Lastly, the combination process respects the privacy of the objects since it is totally unaware of their attributes. Security and privacy issues are important aspects of any learning algorithm, and it is believed that they hinder wider applicability of machine learning and data mining solutions [1].

In classification, the problem of combining classifiers has received a lot of attention over the past years [9]. Theoretical and empirical advances have been recorded and many variants are still being developed. Although clustering is in some respects a problem similar to classification, there has been very little work on issues of combining clustering systems. To see the connection between classification and clustering, consider the goal of classification which is to estimate a function $\phi(\vec{x})$ where $\phi : \vec{x} \rightarrow y \; \forall \vec{x}$ with $\vec{x}$ being the input and y being the class label it maps to. A set of D $(\vec{x}_d, y_d)$ tuples are presented for training. Recall, we defined a clustering algorithm as a function that partitions a set of objects $\mathcal{X}$ to C clusters. We can take this a step further and given the partition estimate a function $\psi(\vec{x})$ where $\psi : \vec{x} \rightarrow y \; \forall \vec{x}$, instead for only $\vec{x_d}$. With this perspective, the goals of classification and clustering become identical, i.e. estimate a function that maps objects to integer numbers (or to a vector of weights if soft decisions are desired). The difference is that in classification we are given D training tuples $(\vec{x}_d, y_d)$ while in clustering we are only given $\vec{x}_d$.

Despite the similarities between clustering and classification, the main problem with combining clustering systems is that the correspondence between cluster labels of different systems is unknown. For example, consider two clustering systems applied to nine data points and clustered in three groups. System A's output is $\vec{o}_A = [1, 1, 2, 3, 2, 2, 1, 3, 3]$ and system B's output is $\vec{o}_B = [2, 2, 3, 1, 1, 3, 2, 1, 1]$, where the i-th element of $\vec{o}$ is the group to which data point i is assigned. Although the two systems appear to be making different decisions, they are in fact very similar. Cluster 1 of system A and cluster 2 of system B are identical, and cluster 2 of system A and cluster 3 of system B agree 2 out of 3 times, as cluster 3 of system A and cluster 1 of system B. If the correspondence problem is solved, then a number of combination schemes can be applied.

Finding the optimum correspondence of clusters requires a criterion and a method for optimization. The criterion used here is maximum agreement, i.e. find the correspondence where clusters of different systems make the maximum number of the same decisions. Second, we must optimize the selected criterion. Even if we assume a 0 or 1 correspondence between clusters with only two systems of C clusters each, a brute-force approach would require the evaluation of $C!$ possible solutions. Although, techniques with $O(C^3)$ complexity exist for finding the correspondence of clusters of two systems, such as the maximum weight bipartite graph matching [93], they do not apply in a straightforward way for $S > 2$ systems. In this chapter, three novel methods are presented for determining the correspondence of clusters of an arbitrary number of systems and combining them. Two of the three methods are formulated and solved with integer linear optimization, and the third uses singular value decomposition. After determining the correspondence between clusters of different systems, the different partitions can be consolidated into one by a number of methods, such as voting or weighted average. Experiments on text and speech document clustering tasks and two gene expression datasets show that combination schemes can achieve improved performance and robustness compared to individual algorithms or common alternatives to combination such as selecting the system with the highest objective function.

The second part of this chapter deals with the issue of cluster validation. A problem that permeates all clustering algorithms is that, because of the lack of labeled data, there is a disconnect between the function that is optimized and the function that is desired.

For example, in document clustering, a common function to minimize is the intra-cluster tf-idf distance between documents, while what is desired is a function that maps documents to (latent) topics. This disconnect makes it possible for clustering algorithms to produce poor or irrelevant clusters. Establishing ways to reject some clusters or assign confidences to clusters (not objects) is an important step that has not received a lot of attention. In this chapter, we show that the degree of overlap between corresponding clusters can be used as a cluster validation measure. Using the new measure, a few, good clusters can be selected rather than attempting to cluster every point in the dataset. Despite the fact that cluster validation has been recognized as an important part of the clustering process, past work assessing the validity of specific clusters dates back to the 70s and early 80s. Most of the recent previous work under the name cluster validation has used different measures to estimate the number of clusters present.

This chapter is structured as follows. In section 3.2, we review related work; in section 3.3, we present our three methods for finding the correspondence of clusters of different systems. In section 3.4, we introduce the similarity of corresponding clusters as a cluster-specific validation measure, allowing us to assign confidences to entire clusters. In section 3.5, we present our experiments on different clustering tasks, i.e. document clustering and gene expression data. Finally, in section 3.6, we discuss our conclusions.

3.2 Related Work on Combining Partitions and Cluster Validation

Combining multiple clustering systems has recently attracted the interest of several researchers in the machine learning community. In [158], three different approaches for combining clusters based on graph-partitioning are proposed and evaluated. The first approach avoids the correspondence problem by defining a pairwise similarity matrix between data points. Each system is represented by a $D \times D$ matrix (D is the total number of observations) where the (i, j) position is either 1 if observations i and j belong to the same cluster and 0 otherwise. The average of all matrices is used as the input to a final similarity-based clustering algorithm. The core of this idea also appears in [117, 51, 43, 181, 46]. A disadvantage of this approach is that it has quadratic memory and computational requirements

in D. Even by exploiting the fact that each of the $D \times D$ matrices is symmetric and sparse, this approach is impractical for high D.

The second approach taken in [158], is that of a hypergraph cutting problem. Each one of the clusters of each system is assumed to be a hyperedge in a hypergraph. The problem of finding consensus among systems is formulated as partitioning a hypergraph by cutting a minimum number of hyperedges. This approach is linear with the number of data points, but requires fairly balanced data sets and all hyperedges having the same weight. A somewhat similar approach is presented in [162], where each data point is represented with a set of meta-features. Each meta-feature is the cluster membership for each system, and the data points are clustered using a mixture model. An advantage of [162] is that it can handle missing meta-features, i.e. a system failing to cluster some data points. Algorithms of this type avoid the cluster correspondence problem by clustering directly the data points using the meta-features.

The third approach presented in [158], is to deal with the cluster correspondence problem directly. As stated in [158], the objective is to *"cluster clusters"*, where each cluster of a system is a vertex in a graph and the weight of the edge between two vertices is the binary Jaccard distance between the two clusters. Each cluster is represented with a D-dimensional binary vector where the i-th position is equal to one if the i-th data point is hypothesized to belong to the cluster. The objective is to partition the graph into approximately balanced, minimally connected sub-graphs. All the clusters (vertices) that belong to the same metacluster (sub-graph) are then combined by taking the average of their representations. Objects are assigned to the metacluster they most strongly belong to. The same core idea of *clustering clusters* can also be found in [43, 40, 52, 23]. In [43], different clustering solutions are obtained by resampling and are aligned with the clusters estimated on all the data. In both [40, 52], the different clustering solutions are obtained by multiple runs of the k-means algorithm with different initial conditions. An agglomerative pairwise cluster merging scheme is used, with a greedy search method to determine the corresponding clusters. In [23], a two-stage clustering procedure is proposed. Resampling is used to obtain multiple solutions of k-means. The output centroids from multiple runs are clustered with a new k-means run. A disadvantage of [23] is that it requires access to the original features

of the data points, while all other schemes of this category do not.

The existing approaches in this category view the cluster correspondence problem as another clustering problem in a proper feature space. For example, Strehl's Meta-Clustering Algorithm (MCLA) [158] uses a graph-partitioning method to identify cluster correspondence, producing approximately balanced metaclusters. However, the assumption about approximately balanced metaclusters is not always justified and in this chapter we offer a motivation on why this is the case, present two alternatives to approximate balancing and in section 3.5 we experimentally demonstrate where these modified schemes can offer improvements over baseline methods.

To see why approximately balanced metaclusters may not be the optimum approach, consider the case of S partitions with C clusters each. Further assume that each one of the S partitions is not very different from each other, e.g. no cluster of a partition is split into two clusters of another partition. Therefore, when estimating correspondence we would like to enforce the following two constraints: a) a single cluster maps to a single meta-cluster and b) no two clusters of the same system are mapped to the same metacluster. This situation is depicted graphically in Figure 3.1.

The standard view of the clustering combination methods is to cluster clusters, that is find a mapping from clusters to meta-clusters just as one would find a mapping from objects to clusters. Integrating the constraints of b) is not a straightforward extension for most clustering methods. For example, the probabilistic framework for clustering assumes that objects are independent and identically distributed (i.i.d). On the other hand, in the case we have described the objects are not i.i.d, in fact due to the constraints of b) clusters of the same system are very much dependent of each other. A suitable framework for integrating constraints is integer linear programming (ILP). A detailed presentation of a method integrating these constraints is given in the next subsection. These constraints produce exactly balanced meta-clusters and also make sure that each one of the clusters in a meta-cluster is from a different system, two requirements that previous methods could not enforce. Strehl's MCLA method produces only approximately balanced meta-clusters with no regard to where the clusters of a metacluster come from.

Now consider a very different situation from above. Suppose we have S partitions but

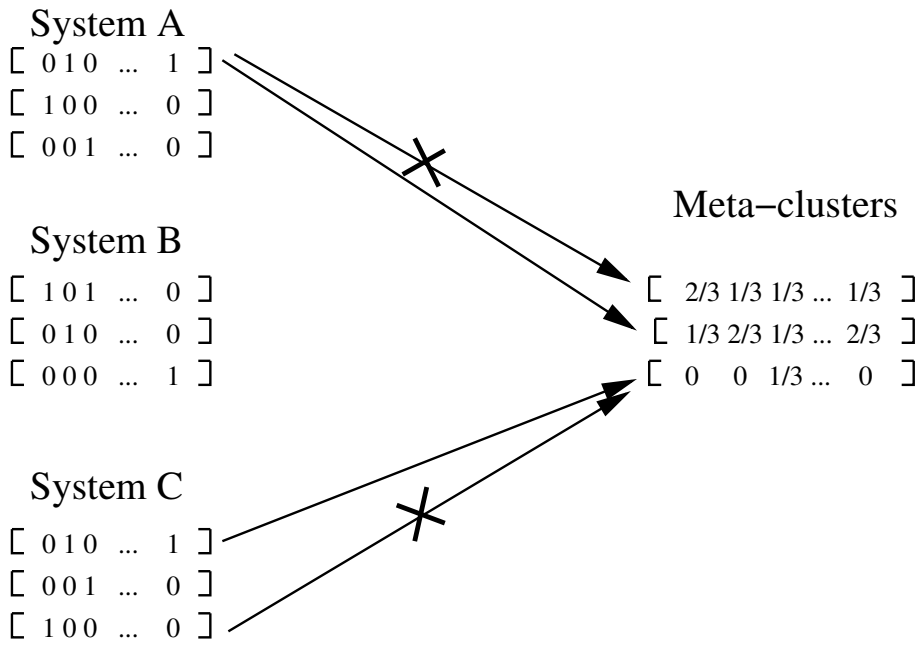

Figure 3.1: Graphic representation of constrained cluster correspondence. Crosses represent not allowable mappings. Two clusters of the same system are not allowed to map to the same metacluster and a cluster is allowed to map to only one metacluster.

with very different number of clusters for each one of them, this is a common case in practice since the correct number of clusters is not known a priori. Different partitions do not map nicely to each other, e.g. a cluster of one partition may map to many clusters of another partition. Therefore, the constraints we have enforced before, i.e. no two clusters of the same system being mapped to the same metacluster, are no longer meaningful. In fact, the metaclusters could be very skewed, and one metacluster may have many more clusters than another. Again, Strehl's MCLA method that forces approximate balancing should not be the optimum approach. In such a case we should not impose any constraints on the size of the metacluster or to the membership of a cluster. The ILP framework is also suitable in such a situation, although in such a case other frameworks can also be used. We also propose a clustering combination method that is based on the Singular Value Decomposition of the $\mathbf{R}$=$D \times CS$ matrix, where each row of $\mathbf{R}$ is an object represented with the cluster memberships of all systems. Using the top C singular vectors to approximate $\mathbf{R}$ can be

viewed as finding corresponding clusters and linearly combining them.

There has also been considerable work on cluster validation, see [67] for a survey, but assigning confidences to clusters has not received a lot of attention. The issue of assessing the validity of individual clusters has received attention during the 70s and 80s (see [76] for a review), but during the 90s the term *cluster validation* refers to finding the number of clusters in a large set of data points. In [47], a measure is used to assign object confidences. In [167], a measure based on the separation of clusters in the attribute space is introduced to characterize cluster novelty. Cluster validation criteria based on the stability of partitions have been suggested before [95, 42, 117, 14], but they have been used to estimate the number of clusters rather than extracting a few, good clusters. A partition may be unstable as a whole, i.e. successive runs of a clustering algorithm produce significantly different partitions, but a subset of the partition may exhibit high stability. i.e. there may be a few clusters that are well detected. In this chapter we propose a method that can be used as a cluster-wise confidence measure, by showing that it correlates with the degree a cluster maps to a real entity.

3.3 Finding and Combining Corresponding Clusters

In the first half of this chapter, we attempt to first find a correspondence between clusters and then combine clusters without requiring the original observation attributes. Therefore, our approaches fall in the third category of cluster combination methods, as detailed in section 3.2.

3.3.1 Constrained and unconstrained combination

Suppose we represent cluster c of system s with the $D \times 1$ vector $\vec{R}_{\{c,s\}}$ (D is the total number of objects) with the d-th element $r^d_{\{c,s\}}$ being:

$$r^d_{\{c,s\}} = \begin{cases} 1 & \text{if } g_s(\vec{x}_d) = c \\ 0 & \text{otherwise} \end{cases}$$

where $\vec{x}_d$ is the vector representation of object d and $g_s(\cdot)$ is the clustering function of system s. The procedure for finding the correspondence between clusters of different systems and

then combining them is shown in Algorithm 1.

The iterative scheme of Algorithm 1 groups similar clusters together so that all clusters assigned to the same meta-cluster are assumed to be in correspondence with each other. The scheme of Algorithm 1 is iterative, starting from an initial condition and then alternating the reward-calculation and maximization steps for a fixed number of iterations or until convergence. In addition, the similarity of a cluster to a metacluster is taken to be the average of the similarities between the cluster and every other cluster in that metacluster. It should be noted that the cluster-to-cluster similarity calculation scales linearly with the number of objects to be clustered. Additional computational gains can be made if we take advantage of the sparsity of the vectors.

The constraints of equation (3.4) ensure that a cluster will be assigned to only one meta-cluster. The optional constraints of equation (3.5) ensure that no two clusters of the same system will be mapped to the same metacluster. The constraints of equation (3.5) are more meaningful when the different clustering systems have the same number of clusters and partitions are not very different from each other. In such a case, two clusters of the same system will represent different entities and should never be mapped to the same meta-cluster. On the other hand, if a class is split into two clusters then constraints (3.5) are no longer true.

The GNU Linear Programming library[1] was used to implement Algorithm 1. Using only the constraints of equation (3.4) will be referred to as the unconstrained combination scheme, while using both constraints of equations (3.4) and (3.5) will be referred to as the constrained combination scheme. It should be noted that using an ILP package for the unconstrained combination scheme is not necessary - any clustering algorithm can be used, such as k-means. In fact, using ILP for (3.4) requires more computations than k-means. The reason ILP was used for (3.4) is for purposes of consistency for experimental comparisons.

After finding the correspondence of clusters of different partitions the next step is to combine the clusters. Here, the vector representations of all corresponding clusters are averaged to obtain matrix $\mathbf{F}$. The final partition is achieved by using the rows of matrix

[1] http://www.gnu.org/software/glpk/glpk.html

Algorithm 1 Constrained/Unconstrained combination of multiple partitions

Define the agreement between clusters $\{c, s\}$ and $\{c', s'\}$ as:

$$g_{\{c,s\},\{c',s'\}} = \frac{\vec{R}^T_{\{c,s\}} \cdot \vec{R}_{\{c',s'\}}}{||\vec{R}_{\{c,s\}}||^2 ||\vec{R}_{\{c',s'\}}||^2} \tag{3.1}$$

The parameters to be estimated are:

$$\lambda^m_{\{c,s\}} = \begin{cases} 1 & \text{if } \{c, s\} \in I_m \\ 0 & \text{otherwise} \end{cases}$$

where $I(m)$ is the set of clusters $\{c, s\}$ that belong to the same metacluster m.

Pick initial values for $\vec{\lambda}$

while number of iterations $\leq$ K **do**

Step 1. Calculate reward

Compute similarity of cluster $\{c, s\}$ to metacluster m $\forall c, s, m$:

$$h^{\{m\}}_{\{c,s\}} = \frac{1}{|I(m)|} \sum_{\{c',s'\} \in I(m)} g_{\{c,s\},\{c',s'\}} \tag{3.2}$$

Step 2. Maximize

$$\vec{\lambda}^* = \underset{\vec{\lambda}}{\text{argmax}} \sum_{m=1}^{M} \sum_{s=1}^{S} \sum_{c=1}^{C_s} \lambda^{\{m\}}_{\{c,s\}} h^{\{m\}}_{\{c,s\}} \tag{3.3}$$

subject to the constraints:

$$\sum_{m=1}^{M} \lambda^{\{m\}}_{\{c,s\}} = 1, \quad \forall c, s \tag{3.4}$$

Optionally we may want to add the following constraints:

$$\sum_{c=1}^{C_s} \lambda^{\{m\}}_{\{c,s\}} = 1, \quad \forall s, m \tag{3.5}$$

end while

The output is a $D \times M$ matrix with the m-th column $\vec{F}_m$ equal to:

$$\vec{F}_m = \frac{1}{|I(m)|} \sum_{\{c,s\} \in I(m)} \vec{R}_{\{c,s\}} \tag{3.6}$$

Matrix $\mathbf{F}$ can be the final output or the rows of $\mathbf{F}$ can be used as new object representations for a final clustering stage.

$\mathbf{F}$ as new object representations and clustering them with k-means. Instead of taking the average and then clustering, we could have used the median - which is equivalent to voting - but there is no guarantee that an object will be assigned to at least one class. Other operators are possible, for example taking the average of the T most similar clusters, i.e. trimmed average.

3.3.2 Singular value decomposition combination

The last combination approach we introduce is based on Singular Value Decomposition (SVD). We construct matrix $\mathbf{R}$ of size $D \times \mathcal{C}$ (D is the number of objects, $\mathcal{C}= \sum_{s=1}^{S} C_s$, C_s is the number of clusters of system s). Each row represents an object and is the concatenation of outputs of different systems. For example, the first C_1 dimensions of the d-th row will be the output of the first system for the d-th object, e.g. a vector with a one associated with the assigned cluster and zeros in all other elements or a vector of cluster posteriors for a soft decision. The next C_2 dimensions will be the output of the second system and so on. $\mathbf{R}$ can be approximated as $\mathbf{R} \approx \mathbf{U} * \mathbf{S} * \mathbf{\Lambda^t}$ where $\mathbf{U}$ is orthogonal and of size $D \times M$, $(M < \mathcal{C})$, $\mathbf{S}$ is diagonal and of size $M \times M$ and $\mathbf{\Lambda}$ is orthogonal and of size $\mathcal{C} \times M$. The final metaspace is $\mathbf{R} * \mathbf{\Lambda}$ of size $D \times M$. If we define $p^{(d)}_{\{s,c\}} = p(cluster = c | object = d, system = s)$, then the $\phi_{d,m}$ element of $\mathbf{R} * \mathbf{\Lambda}$ is given by:

$$\phi_{d,m} = \sum_{s=1}^{S} \sum_{c=1}^{C_s} \lambda_{\{s,c\},m} \; p^{(d)}_{\{s,c\}} \tag{3.7}$$

The λ_* parameters are estimated using SVD and can be interpreted as a similarity between cluster $\{c, s\}$ and metacluster m. Since λ_* are not constrained to be between zero and one, a final clustering is performed using the $\phi_{d,m}$ representation and k-means. In our experiments, all systems had the same number of clusters ($C_s = C, \forall s$) and the number of metaclusters was equal to the true number of classes ($M = C$). But there is nothing in principle that prohibits SVD to be used in the more general case where $C_i \neq C_j$.

Under the assumption that all systems have the same number of clusters equal to the true number of classes, SVD is of $O(S^3C^3)$ complexity, but computations scale down with the number of zero elements of $\mathbf{R}$ and with the number of singular vectors that need to be computed (here C out of SC). For example, in a 2.6GHz Pentium-4 processor with 2GB

of RAM running Matlab, computing the first 40 singular vectors of a 20K×4K matrix was completed in a few minutes.

3.4 Similarity of Corresponding Clusters as a Cluster Validity Measure

An important issue of the clustering process is validation of the results. After the clustering algorithm has generated a partition how confident are we that each one of the clusters represents a real entity? We are not interested in object-level confidences, as in classification, but rather in cluster-level confidences. If we were able to find a cluster-wise measure that correlates with cluster performance then we could reduce the risk of choosing false clusters.

Up to this point we have presented various ways for estimating the correspondence between clusters of different systems. In this section, we introduce the average similarity of corresponding clusters as a possible cluster validity measure.

For each metacluster $m = 1 : M$ calculate:

$$Confidence_m = \frac{1}{|I_m|^2} \sum_{\{c,s\} \in I_m} \sum_{\{c',s'\} \in I_m} \cos(\vec{R}_{\{c,s\}}, \vec{R}_{\{c',s'\}}) \tag{3.8}$$

where I_m is the set of corresponding clusters for metacluster m. The measure of equation 3.8 is always between 0 and 1. The hypothesis we test is that the bigger the similarity the more likely it is that the meta-cluster maps to a real entity. Stated otherwise, the stronger the consensus among systems for a specific cluster is, the more likely the meta-cluster is real.

There are two main measures of cluster validity in the literature [67]: *compactness* and *separation*. Compactness measures how close the members of each cluster are and separation measures how distinct clusters are. There are a number of cluster validity criteria that combine these two measures. Also, there are a number of different distance measures that can be used in each criterion. For example, in [4], 18 different validity indices are compared. The fact that there is no obvious way to combine compactness and separation to a single measure is a shortcoming of current validation techniques.

The cluster validity measure of equation (3.8), although only calculating the intra-metacluster similarity, is able to measure the separation of clusters as well. If two classes

are close to each other (low separation) then different clustering algorithms will produce different results and the corresponding clusters will be dissimilar to each other. Therefore, similarity in the metaspace captures both compactness and separation in the attribute space.

Recently, methods to estimate the number of clusters in a dataset were suggested that are based on the stability of partitions [95, 42, 117, 14]. A clustering algorithm runs multiple times and the similarity of all possible pairs of partitions is calculated using the Hungarian method [93]. The degree of similarity of partitions is shown to be an indicator of the number of clusters in a dataset. Equation (3.8) is similar with [95, 42, 117, 14] in the respect that they both use multiple partitions but the goal here is to assess the validity of individual clusters, not estimate the number of clusters. Even if we cannot correctly estimate the number of clusters, we may still be able to find a few clusters that map well to real entities.

3.5 Experiments

3.5.1 Description of the data sets

Our experiments are conducted in three main tasks, clustering electronic postings (newsgroups), conversation clustering by topic, and clustering of gene expression data.

For clustering electronic postings, we used the 20Newsgroups corpus [94], a collection of 18827 postings to electronic discussion forums or newsgroups. There are 20 different classes in 20Newsgroups and the corpus is almost perfectly balanced, i.e. an equal number of postings per newsgroup. Preprocessing consisted of converting all numbers to a single token and removing the *From:* field. Words with 5 or more occurrences were kept, resulting in a vocabulary of 34658 words. 20Newsgroups has a mix of formally and informally written articles, with postings ranging from carefully prepared essays to less-structured emails.

For the conversation clustering task, we used the Fisher corpus[2] [32] a collection of 5-minute telephone conversations on a predetermined topic. The topic was selected from a list of 40 before the start of the conversation. Clustering conversations by topic can be important in a number of scenarios, such as summarizing business meetings or mining customer service

[2]http://www.ldc.upenn.edu/Fisher/

call-centers. After eliminating conversations where at least one of the speakers was non-native and conversations with topicality 0 or 1 we were left with 10127 conversations or 20254 conversation sides. The topicality label gives the degree to which the suggested topic was followed and is an integer number from 0 to 4, 0 being the worse. There were about 15M words in the collection, and conversation sides were unequally divided among the 40 topics. The median number of sides per topic was 478 with a standard deviation of 202 (max 1018, min 198). The original transcripts were minimally processed; acronyms were normalized to a sequence of characters with no intervening spaces, e.g. *t. v.* to *tv*; word fragments were converted to the same token *wordfragment*; all words were lowercased; and punctuation marks and special characters were removed. Some non-lexical tokens are maintained such as *laughter* and filled pauses such as *uh*, *um*. Backchannels and acknowledgments such as *uh-huh*, *mm-hmm* are also kept. Words in the default stoplist of CLUTO (total 427 words) were removed and only words with 5 or more occurrences were kept, leading to a vocabulary of 22859 words. The Fisher corpus was created to facilitate speech recognition research and, to the best of our knowledge, it has not been used before for conversation clustering. The Fisher corpus brings interesting new challenges to the problem of document clustering. It bears the same core characteristics of document clustering, such as a very high dimensional space, but unlike other corpora such as Reuters-21578 or 20Newsgroups it consists of transcripts of spoken language. The language is less structured and more spontaneous than written text, including disfluencies such as repetitions, restarts and deletions both at the word and above-word level. An additional difficulty stems from the fact that 14% of words in spoken language text are pronouns vs. 2% in written text [137]. Since pronouns substitute for nouns or noun phrases that are generally considered to convey semantic information, they may have a negative impact on clustering or classification performance. On the other hand, the vocabulary is about half the size of a comparable corpus of written text. Also, conversation clustering involves first converting speech into text which is a procedure that generates errors (state-of-the-art systems achieve a word error rate of about 15%-20% [127]). In this chapter we have not dealt with the issue of ASR errors, i.e. the input to the clustering algorithms is the human-transcribed conversations; it is addressed in a later chapter in the thesis.

The third task we have applied our methods is clustering of gene expression data. There were two subtasks. The first was the yeast cell cycle data set as used in [177]. Gene expression levels from 384 genes were monitored in 5 phases of the yeast cell cycle, in 17 time points. The goal is to cluster the 384 genes, each represented as a 17-dimensional vector, to 5 clusters. The second subtask is the multiple tumor data set as used in [176]. Gene expression levels from 7128 genes were recorded for 123 cancer patients with one of 11 possible cancer types. The final experiments were conducted using 680 genes since not all were considered relevant for cancer classification. The objective is to cluster the 123 cases, each represented as a 680-dimensional vector to 11 groups. Analyzing gene expression data can reveal which groups of genes are co-expressed as a result of being co-regulated and can offer clues to the design of novel drugs.

The tasks we have chosen represent real, challenging clustering problems yet they are different in many respects. The conversation and newsgroup clustering tasks are discrete attribute clustering problems, whereas the gene expression data sets use continuous attributes. In conversation and newsgroup clustering there are many hundreds of points per group, whereas at most a dozen for the multiple tumor set. Finally, the distribution of number of points per group is much more imbalanced in Fisher than in gene expression data sets or 20Newsgroups. The choice of different tasks was made to evaluate the degree to which our methodologies are task independent.

3.5.2 First stage clustering algorithms

In this chapter we used a number of different clustering algorithms. Most of them were implemented in CLUTO[3], a software toolkit designed for clustering in high and low dimensional spaces. A number of different optimization criteria and distance metrics are available through CLUTO. Some of criteria, such as I_1 and I_2 attempt to minimize different variants of intra-cluster similarity. For example, using the I_2 criterion with the cosine distance maximizes the function $\sum_{k=1}^{M} \sqrt{\sum_{\vec{u},\vec{v} \in c_k} \cos(\vec{u}, \vec{v})}$, where c_k is the set of documents in cluster k and $\vec{u}, \vec{v}$ are the tf-idf vector representations of documents u, v respectively. Other criteria

[3]http://www-users.cs.umn.edu/~karypis/cluto/

attempt to minimize the inter-cluster similarity and yet other criteria, such as H_1 and H_2, attempt to optimize a combination of both. The G_1 and G_1' criteria are based on graph partitioning. For more information on the optimization criteria and methods, see [182].

For the document clustering experiments, we have also implemented a model-based clustering algorithm, the mixture of multinomials (MixMulti). The main assumption is that each topic can be represented as a distinct multinomial, i.e. different counts of each word given a topic. The learning process consists of fitting a mixture of multinomials in the entire dataset. The probability of a document d is given by: $p(d) \propto \sum_{c=1}^{M} p(c) \prod_{w \in W_d} p(w|c)^{n(w,d)}$ where M is the number of topics, W_d is the set of unique words that appear in document d, $p(w|c)$ is the probability of word w given cluster c, and $n(w,d)$ is the count of word w in document d. The cluster c that each document is generated from is assumed to be hidden. Training such a model is carried out using the Expectation-Maximization algorithm [36]. In practice, smoothing the multinomial distributions is necessary. In this chapter the add-one smoothing technique was used, i.e. a special case of the Dirichlet prior where all words are assumed to be seen at least once per cluster. The mixture of multinomials algorithm is the unsupervised analogue of the Naive Bayes algorithm and has been successfully used in the past for document clustering [123]. Mixture models, in general, have been extensively used for data mining and pattern discovery [28].

For the gene expression data, the k-means algorithm is used since it was shown to achieve the state-of-the-art performance compared to other clustering algorithms [177].

3.5.3 Cluster correspondence results

Various criteria for evaluating clustering performance have been suggested in the past based on comparing the partitions with ground truth. In this chapter, we use the three evaluation criteria that were covered in chapter 2; classification accuracy after a one-to-one mapping of clusters to classes, normalized mutual information (NMI) between clusters and topics, and (for later experiments) the adjusted Rand index. Results are reported in all three corpora, but the most extensive experiments are performed on the Fisher corpus, since conversational speech is the main target of our work.

In Table 3.1, the clustering combination results for the 20Newsgroups corpus are shown. 100 systems were generated for each clustering method and then combined with the three different combination algorithms or the system with the highest value of the objective function is selected (Best of 100). From Table 3.1 we can observe that the combination schemes offer consistent improvement over all clustering methods except for I_1, with the MixMulti method being benefited most. The finding that I_1 did not give improvement may imply that good first stage clustering is needed to get further improvement from metaclustering.

In Table 3.2, the results of applying the three partition combination methods on a number of different clustering algorithms is shown for the Fisher corpus. For each one of the clustering algorithms we generated 1000 partitions by seeding the algorithms with a different initial condition and running until convergence. The 1000 runs were divided in 10 sets of 100. For each set, we applied our combination schemes and then calculated the average and standard deviation. This way we can also assess the robustness of each approach to the underlying systems. In the Fisher experiments all clustering methods benefit, with the biggest gain again for MixMulti.

For both data sets, we compare our approaches with the common alternative of selecting the system with the highest objective function, sometimes referred to as multiple restarts. Note that this is possible only if the partitions are generated with the same clustering algorithm. On the other hand, our combination methods do not have this requirement, making it possible to combine partitions generated by heterogeneous systems. Overall, we can see that having a single run is clearly inferior to applying a combination algorithm or selecting the max. In one case, the mixture of multinomials, the decrease of classification error is more than 30% relative. Also, the variability of partitions of a single run is much higher than any combination scheme or selecting the max. Comparing the combination methods with the multiple restarts method, we see a small but consistent improvement in performance for all clustering algorithms, although not necessarily with the same combination method. The best numbers for each clustering algorithm, along with cases not statistically different from the best, are shown in bold. For the Fisher clustering task of 20254 documents, an

Table 3.1: Average performance of different combination schemes on various clustering algorithms for the 20Newsgroups corpus. The same 100 systems are used for each combination method.

	Single Run	**Best of 100 runs**	**SVD Combin.**	**Constr. Combin.**	**Unconstr. Combin.**
$\boldsymbol{I_1}$					
Accuracy	**.422**	.412	**.418**	.417	.408
NMI	.486	.485	.481	.480	.463
$\boldsymbol{I_2}$					
Accuracy	.575	.603	**.634**	.615	**.639**
NMI	.601	.621	.637	.628	.640
$\boldsymbol{G_1}$					
Accuracy	.535	.561	**.581**	.562	**.578**
NMI	.561	.585	.593	.581	.582
$\boldsymbol{G'_1}$					
Accuracy	.576	.608	**.642**	.630	.563
NMI	.584	.603	.631	.622	.620
$\boldsymbol{H_1}$					
Accuracy	.570	.584	**.636**	**.641**	.549
NMI	.593	.610	.629	.627	.592
$\boldsymbol{H_2}$					
Accuracy	.586	.611	**.656**	.639	.602
NMI	.598	.616	.646	.634	.628
MixMulti					
Accuracy	.534	.620	**.679**	**.677**	.621
NMI	.587	.625	.662	.656	.651

Table 3.2: Average performance of different combination schemes (and standard deviation in parentheses) on various clustering algorithms for the Fisher corpus. 10 trials are performed where each trial combines 100 systems. The top scores for classification accuracy, along with the scores that are not statistically different, are highlighted in bold.

	Single Run	**Best of 100 runs**	**SVD Combin.**	**Constr. Combin.**	**Unconstr. Combin.**
$\boldsymbol{I_1}$					
Accuracy	.691(.023)	.719(.014)	.724(.012)	**.732**(.009)	**.737**(.003)
Adj. Rand	.470(.025)	.469(.018)	.490(.010)	.489(.010)	.491(.005)
$\boldsymbol{I_2}$					
Accuracy	.740(.025)	.780(.012)	.782(.012)	**.793**(.002)	**.794**(.004)
Adj. Rand	.643(.021)	.672(.013)	.672(.011)	.684(.002)	.686(.002)
$\boldsymbol{H_1}$					
Accuracy	.753(.027)	.791(.014)	.796(.014)	**.806**(.002)	**.802**(.007)
Adj. Rand	.668(.022)	.695(.014)	.697(.010)	.710(.002)	.708(.004)
$\boldsymbol{H_2}$					
Accuracy	.742(.026)	**.780**(.023)	**.780**(.000)	**.784**(.006)	**.784**(.007)
Adj. Rand	.655(.020)	.683(.018)	.681(.001)	.693(.003)	.692(.004)
$\boldsymbol{G_1}$					
Accuracy	.642(.019)	.619(.014)	**.670**(.001)	.659(.010)	**.669**(.008)
Adj. Rand	.472(.016)	.458(.009)	.520(.003)	.490(.016)	.483(.001)
$\boldsymbol{G'_1}$					
Accuracy	.716(.024)	.755(.023)	.755(.001)	**.765**(.008)	**.759**(.005)
Adj. Rand	.609(.018)	.638(.018)	.635(.001)	.653(.007)	.649(.002)
MixMulti					
Accuracy	.657(.031)	.714(.022)	**.767**(.001)	.740(.010)	**.761**(.004)
Adj. Rand	.582(.025)	.624(.023)	.671(.001)	.664(.006)	.659(.006)

absolute difference in classification accuracy of 0.007 is significant at the 0.05 level[4]. The highlighted results indicate that all combination algorithms offer gains, but unconstrained combination most consistently gives high results for Fisher. Constrained combination was not better than unconstrained combination or SVD, probably because the clusters in the combined partitions were not in a one-to-one correspondence with each other. This was confirmed by evaluating single partitions where a few topics would sometimes be almost entirely missed while some other times they would be detected with good performance.

An important but infrequently addressed problem in iterative-based partitioning algorithms, is the robustness of partitions. The same algorithm, on the same dataset will produce different partitions if initialized differently. For example, 100 systems generated with the H_1 criterion had classification accuracies ranging from 0.802 to 0.690. Therefore even if H_1 is on average the best-performing criterion on the Fisher corpus, individual runs can have worse performance than below-average criteria. Using the common approach of selecting the system with the highest objective function reduces the variability of partitions but does not eliminate the problem. The combination methods proposed here show a clear improvement in this issue. For all cases, we see a significant decline in variability relative to multiple restarts and even more to single runs.

In Table 3.3, the performance as a parameter of the number of combined partitions is shown. The mixture of multinomials method is used to generate the partitions. We can observe that performance converges quickly as the number of partitions is increased. Using more than 100 partitions does not offer added gains.

In Table 3.4, the combination of heterogeneous systems is shown. Unlike Table 3.2 where the different partitions where generated with different initial conditions of the same criterion, here we use partitions generated from different criteria. Note that we cannot select the "best of 100" system as in Table 3.2 because partitions are generated with different objective functions. From Table 3.4 we can see that the combination methods offer increased performance and robustness compared to single systems, and perform close to the best systems for all cases. Combining the best and the worse clustering algorithms, H_1 and

[4]We do not have significance tests for scores other than classification accuracy.

Table 3.3: Average performance of SVD combination (and standard deviation in parentheses) for the Fisher corpus and for different number of partitions combined, generated with MixMulti. 10 trials are performed for each reported number.

	1	**3**	**10**	**50**	**100**	**300**
Acc.	.657(.031)	.715(.026)	.747(.001)	.763(.004)	.767(.001)	.768(.001)
NMI	.727(.013)	.750(.008)	.769(.003)	.778(.002)	.779(.001)	.779(.001)
A.R.	.582(.025)	.631(.017)	.663(.006)	.670(.003)	.671(.001)	.671(.001)

MixMulti, produced the best results that we could obtain on the Fisher corpus.

The next set of experiments is on gene expression data. Table 3.5 shows the effect of the various combination methods using k-means as the underlying system on the yeast dataset. We did not experiment with other methods in the gene expression data, since k-means was compared with other clustering algorithms and was shown to be one of the top performers in this task in [177]. It is interesting to see that selecting the system with the optimum objective function, i.e. minimum total intra-cluster distance for k-means, leads to worse results both in terms of performance and robustness compared to the single run case. Analyzing the results of the multiple restarts method we observed two modes. One with very high performance and one with very low. Multiple restarts was not very reliable in filtering out the least-performing systems. On the other hand, the other combination approaches offer gains both in terms of performance and robustness.

In Table 3.6, the results are shown for the multiple tumor dataset. The same trends as with the yeast dataset can be observed. Selecting the system with the minimum total intra-cluster distance degrades the results than using a single system, and the three combination methods offer improvement in both performance and robustness over the single run case. The difference between classification accuracy of SVD combination and the best of 100 runs was significant at the 0.08 level for the yeast set and at the 0.2 level for the multiple tumor set. These differences are less significant than in the Fisher corpus, primarily because of small size of the datasets.

Table 3.4: Average performance of SVD combination (and standard deviation in parentheses) on the Fisher corpus using partitions generated by heterogeneous criteria. 10 trials are performed where each trial combines 100 systems.

	SVD combined
All criteria	
Accuracy	.762(.018)
NMI	.790(.005)
Adj. Rand	.649(.017)
$\boldsymbol{H_2}$**,**$\boldsymbol{H_1}$**,**$\boldsymbol{I_2}$	
Accuracy	.787(.013)
NMI	.799(.004)
Adj. Rand	.686(.012)
$\boldsymbol{H_1}$**,MixMulti**	
Accuracy	.815(.007)
NMI	.812(.004)
Adj. Rand	.713(.002)

3.5.4 Comparison with other cluster combination methods

In Table 3.7 we compare the performance of the unconstrained combination scheme and two baseline cluster combination algorithms, MCLA [158] and Topchy's mixture model [162]. In [162] each data point is represented as a $C * S$ binary vector and a mixture model is estimated on the new representation. Topchy's method avoids the cluster correspondence problem by clustering data points directly using a meta-representation. The mixture model was trained using EM with smoothed counts, 10 runs are performed and the one with the highest likelihood retained. We can observe from Table 3.7 that it is not the case that a combination method is always better than another. In four out of seven cases the unconstrained combination attained the highest adjusted rand index. MCLA had the highest performance in two cases and the mixture model of [162] in one. It is interesting that in the cases where MCLA achieves the best performance it does so by a wide margin.

Table 3.5: Average performance of different combination schemes (and standard deviation in parentheses) on the yeast dataset with 10 trials where each trial combines 100 systems. k-means is applied to generate the clustering systems.

	Single Run	**Best of 100 runs**	**SVD Combin.**	**Constr. Combin.**	**Unconstr. Combin.**
Accuracy	.699(.041)	.689(.048)	.721(.025)	.707(.033)	.708(.047)
NMI	.525(.013)	.521(.018)	.531(.010)	.529(.011)	.533(.019)
Adj. Rand	.485(.019)	.478(.021)	.498(.010)	.497(.012)	.496(.019)

Table 3.6: Average performance of different combination schemes (and standard deviation in parentheses) on the multiple-tumor dataset with 10 trials where each trial combines 100 systems. k-means is applied to generate the clustering systems.

	Single Run	**Best of 100 runs**	**SVD Combin.**	**Constr. Combin.**	**Unconstr. Combin.**
Accuracy	.850(.045)	.801(.061)	.870(.027)	.844(.021)	.844(.022)
NMI	.885(.021)	.869(.033)	.892(.012)	.882(.013)	.882(.013)
Adj. Rand	.864(.060)	.784(.091)	.894(.016)	.876(.015)	.870(.015)

Table 3.7: Adjusted Rand Index of the unconstrained combination scheme with two baseline cluster combination methods. The same 100 partitions are combined with 40 clusters each.

	Topchy's [162]	**MCLA [158]**	**Unconstr.**
$\boldsymbol{I_1}$	.532	.628	.491
$\boldsymbol{I_2}$	.660	.664	.686
$\boldsymbol{H_1}$	.688	.687	.708
$\boldsymbol{H_2}$	.678	.666	.692
$\boldsymbol{G_1}$	.522	.628	.483
$\boldsymbol{G_1'}$	.611	.628	.649
MixMulti	.682	.669	.659

In Table 3.8 we compare the performance of the unconstrained combination scheme with MCLA, when combining partitions with very different numbers of clusters. There are 46 partitions that are combined with number of clusters ranging from 15 to 60. Under such conditions, the MCLA assumption that meta-clusters should be approximately balanced is likely to be a poor one. The results confirm this; MCLA is worse than the unconstrained combination of partitions. Comparing with Table 3.7, where the combined partitions had the same number of clusters (equal to the true number of topics), we observe that in the two cases where MCLA was better than the unconstrained combination, the differences are either reduced or reversed. For the rest of the cases unconstrained combination holds its advantage.

3.5.5 Cluster validity results

In section 3.4 we introduced a cluster validity measure that was based on the stability of different solutions. The hypothesis is that less distinct clusters will have less stable membership vectors than distinct clusters.

To evaluate this hypothesis, we generate a number of systems, estimate the correspondence between clusters of different systems and for each metacluster computed the average similarity according to equation (3.8), a measure of agreement between corresponding clus-

Table 3.8: Adjusted Rand Index of combining partitions with different number of clusters on the Fisher corpus. Generated partitions have between 15-60 clusters; the number of clusters of the final partition is 40.

	Unconstr.	**MCLA**
$\boldsymbol{I_1}$	.409	.491
$\boldsymbol{I_2}$	.622	.584
$\boldsymbol{H_1}$	.589	.579
$\boldsymbol{H_2}$	.602	.588
$\boldsymbol{G_1}$	.473	.461
$\boldsymbol{G_1'}$	.601	.578
MixMulti	.590	.589

ters. For each meta-cluster, we calculate the median of all its clusters and then computed the F-measure between the meta-cluster and all classes, and picked the best. The F-measure is commonly used in information retrieval and is suitable when comparing sets of different sizes. If we define $a_{1,1}$ as the count of objects that are in common between the cluster and the class, $a_{1,2}$ the count of objects that are in the cluster but not in the class and $a_{2,1}$ the count of objects that are not in the cluster but are in the class, then we have that $R = \frac{a_{1,1}}{a_{1,1}+a_{2,1}}$, $P = \frac{a_{1,1}}{a_{1,1}+a_{1,2}}$ and $F = \frac{2PR}{P+R}$. The process is repeated using different number of clusters.

We have also compared our method with a measure that takes into account both the compactness and separation of a cluster in the original attribute space. For cluster c, the measure is defined as [76]:

$$S_c = \frac{\min_{c'} ||\mu_{c'} - \mu_c||^2}{\frac{1}{|I_c|}\sum_{x \in I_c} ||x - \mu_c||^2} \tag{3.9}$$

where I_c is the set of objects that belong to cluster c and μ_c is the centroid of c. Since we have evaluated the method on the Fisher corpus, the tf·idf representation was used for each document x. In Figure 3.2, we observe that for small values of the number of clusters the correlation coefficient between meta-cluster confidence and F-measure is between 0.65-

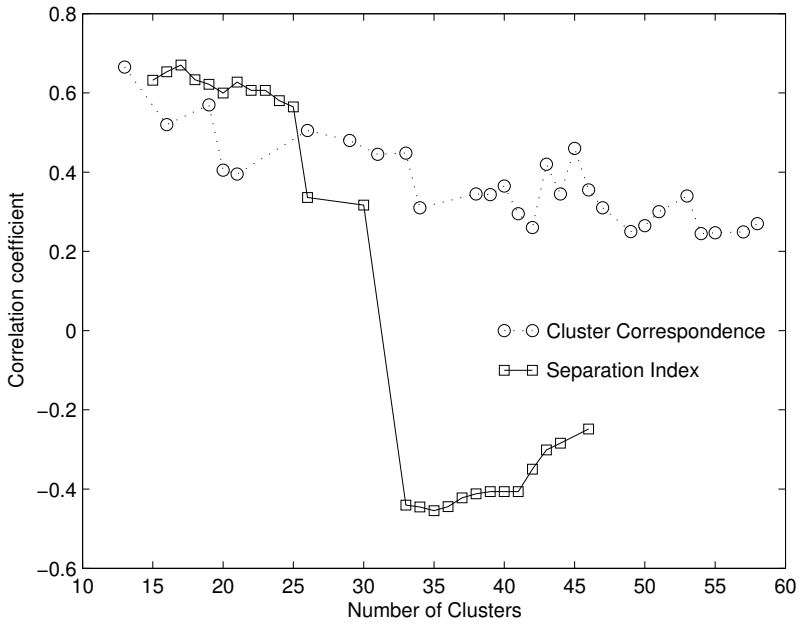

Figure 3.2: Correlation coefficients between F-measure and cluster validation measures for the Fisher corpus and for various number of clusters. Only values with p value ≤ 0.1 are shown.

0.40 with a p value less than 0.1, a clear indication of relation[5]. The same is true for the separation index as well, but as the number of clusters increases above 30 there is a sudden flip in the correlation sign. Since the point of sign flip is not readily predictable, this results in the separation index being less useful than the cluster confidence measure of equation (3.8). As the number of clusters grows larger, the correlation coefficient grows smaller. Although not shown in Figure 3.2 because the p-value was higher than 0.1, we ran experiments with up to 80 clusters.

[5]the p-value is the probability of getting a correlation as large as the observed value by random chance, when the true correlation is zero

3.6 Discussion

In this chapter we have presented three new methods for the problem of combining multiple clustering partitions. Two of the methods are based on integer linear programming and the third is based on singular value decomposition. The combination methods have been tested on two different and challenging tasks, i.e. document clustering and clustering gene expression data, using partitions generated from a variety of clustering algorithms. Results show that combination methods can offer improvement in two important issues: performance and robustness. The combined partition was shown to achieve higher performance than the common alternative of selecting the partition with the highest objective function (multiple restarts) and also shown to offer increased robustness. Robustness of partitions, i.e. the degree to which different runs of the same algorithm produce similar partitions, is an important but infrequently addressed problem in clustering. Comparing with previous cluster combination methods, we found that it is not the case that a combination method always outperforms the other. However, the unconstrained combination scheme was shown to achieve the highest performance for the majority of the cases.

We have also proposed a cluster validity measure based on the degree of similarities between corresponding clusters. Unlike past work that has used cluster validity measures as a measure for estimating the number of clusters, we use the new cluster validity measure to assign cluster-specific confidences. The new measure was shown to correlate with how accurately the cluster maps to a real entity better than the separation index. By having available a cluster-specific validity measure, we can output a few clusters that we are confident they are correct and reduce the risk of reporting false clusters. Thus, effectively we are clustering a subset of the available objects. In contrast with the separation index as a cluster validation baseline, we found that the correlation of the cluster correspondence measure stayed positive.

One of the most intriguing findings of our experiments, which was confirmed in 20Newsgroups and Switchboard-I [21], is that combining partitions generated with mixture of multinomials recorded a decrease of classification error of 30% relative to a single run. These results suggests that when evaluating a clustering algorithm it is also important to bench-

mark its performance when multiple partitions are combined and not only when taking a single run.

In future work, it will be interesting to explore the reasons that allow the combination of some partitions to be more successful than others. We have observed that poor first stage performance hinders performance of metaclustering, but that using the best first stage system does not always lead to the best metaclustering. It appears that average single run performance is not a very good predictor of multiple run metaclustering, and other factors (such as complementary cluster confidences) may be useful for identifying good candidates for metaclustering. In addition, the cluster validity measure that was introduced can be used for feature selection for clustering. The hypothesis, which remains to be confirmed, is that removing a relevant feature decreases the similarity of the corresponding clusters and removing an irrelevant feature increases the similarity of the corresponding clusters.

Chapter 4

TEXT CLASSIFICATION BY AUGMENTING THE BAG-OF-WORDS REPRESENTATION WITH REDUNDANCY-COMPENSATED BIGRAMS

In this chapter, we revisit a filter feature selection method previously introduced in the literature and dismissed as non-effective. We show that the reason for the poor performance attained by this measure is not due to the measure itself but due to the way it was estimated. Properly modifying the estimation method by smoothing the probability estimates resulted in a performance that was similar to a top performing method. The old measure is interpreted under new light and it is hypothesized that it can give superior results than other methods when selecting word pairs, instead of individual words. We propose a modification to the filter feature selection method to add word pairs to the bag-of-words representation, elaborate its relation with another feature selection method, and provide experimental results that show consistent gains over using individual words.

4.1 Background

A major challenge of the text classification problem is the representation of a document. The simplest and almost universally used approach is the bag-of-words representation, where the document is represented with a vector of the word counts that appear in it. Depending on the classification method, the bag-of-words vector can be normalized to unity and scaled so that common words are less important than rare words, such as in the tf·idf representation.

Despite the simplicity of such a representation, classification methods that use the bag-of-words feature space often achieve high performance. In the past, a number of attempts have been made to augment or substitute the bag-of-words representation with richer features. In [100, 53] linguistic phrases, proper names and complex nominals are used, and in [160, 133]

bigrams are added to the feature space. In [131], character n-grams are used for text classification and sense disambiguated words are used in [118]. A recent comprehensive study [118] surveys the different approaches that have been taken thus far and evaluates them on standard text classification resources. The conclusion is that more complex features do not offer any gain when combined with state-of-the-art learning methods, such as Support Vector Machines (SVMs).

We argue that a reason past approaches have failed to show improvements is that they have looked only at the *relevance* of the new features and not *redundancy*. The issues of relevance and redundancy are both central to the choice of optimum feature subset selection [90, 179]. Relevance is the degree to which a feature is useful for classification by itself, and redundancy is the degree to which a feature is correlated with other features. If a feature has high relevance but is also strongly correlated with other equally or more relevant features, adding it to the feature subset can actually hurt classification performance in the typical situation when training is limited. When constructing more complex representations, the number of potential features can increase exponentially. For example, using bigrams increases the vector dimension from V to V^2, where V is the vocabulary size. With so many features, care must be taken to include not simply those that are relevant by themselves but only those that are jointly relevant with the rest of the features.

Past approaches in feature subset selection, include [179, 90, 98, 17]. In [179], all features with relevancy above a threshold are selected and then an elimination procedure is employed: if the pairwise correlation between features f_i and f_j is higher than the correlation of f_j with class c then f_j is eliminated as a potential feature. In [90], the feature selection problem is framed as finding the Markov blanket of feature f_i in a Markov Random Field. A greedy, backward elimination procedure is employed where each feature is eliminated one at a time. In [98] features are ranked according to the difference of relevance (information gain of f_i) and redundancy (maximum information gain of f_i and f_j, for all f_j). In [17] a measure called Explaining Away Residual (EAR) is proposed. The EAR measure adds variables $Z \subseteq W$ to the variable X to maximize $E_{p(Q,X,W)}[\log \hat{p}(Q|X,W))]$, where Q is the classification variable. The quantity $p(Q,X,W)$ is the true joint probability and $\hat{p}(Q|X,W)$ is the fitted distribution. Maximizing the above quantity can be shown to be equivalent to

maximizing $I(X;Z|Q) - I(X;Z)$ where $I(X;Y)$ is the mutual information between X and Y given by:

$$I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{4.1}$$

A major problem with determining redundancy is the amount of computation needed. Algorithms such as [179, 90, 98] are of order $O(T^2)$ where T is the original number of features. Adding bigrams as potential features makes such an approach impractical, since $T = V + V^2$ and V is usually on the order of tens of thousands. Even approaches such as [179] with less than quadratic requirements can pose overwhelming computational burdens. The unmanageable computational requirements prohibit the use of these techniques for features of higher-order than unigrams.

In this chapter, we augment the bag-of-words representation with selected bigrams. We propose a filter approach to feature selection with a criterion for determining the redundancy of a bigram based on its unigrams. Although this approach is not optimum, meaning that only a portion of possible feature combinations are examined for redundancy, it is shown that it can offer gains in challenging text classification tasks and that it scales efficiently with vocabulary size. Performance is not the only reason bigrams are a suitable target for augmenting the feature space. Another important reason is that a common way to interpret and describe the topics present is to output the top-N discriminative features. Adding bigrams to the list can offer a more natural interpretation, although we have no formal way of measuring this.

4.2 Revisiting the KL-divergence filter feature selection

We propose to use the measure given in equation (4.2) to rank words according to their individual relevance.

$$KL_w = D[p(c|w)||p(c)] = \sum_{c=1}^{C} p(c|w) \log \frac{p(c|w)}{p(c)} \tag{4.2}$$

A measure very similar to (4.2) has been suggested before in the literature although not in the exact same formulation. In [175], an experimental comparison of many different filter feature selection methods is presented and one of the methods, termed mutual information

(I_w) is given by:

$$I_w = \sum_{c=1}^{C} p(c) \log \frac{p(w,c)}{p(w)p(c)} = \sum_{c=1}^{C} p(c) \log \frac{p(c|w)}{p(c)} = -D[p(c)||p(c|w)] \tag{4.3}$$

The term mutual information is a poor choice, since equation (4.3) does not express mutual information. On the other hand, the Information Gain (IG) measure is mutual information if we assume a Bernoulli random variable for word w. It can be observed that equations (4.2) and (4.3) are very similar. In [175], an experimental comparison of (4.3) with other filter feature selection methods showed that I_w had an abysmal performance, clearly worse than most of the other methods. The conclusion drawn by [175] was that I_w should be dismissed as a filter feature selection method for text classification. Because [175] was an influential paper in the text classification and feature selection literature, receiving more than 500 citations up to date,[1] subsequent studies have adopted the claim made in [175]. For example, a more recent comprehensive experimental comparison of filter feature selection methods [49] does not even evaluate I_w, citing [175] as proof of its inefficiency.

We claim that the reason that KL_w and I_w have exhibited poor performance is not due to the measure itself but due to the way it was estimated. Equation 4.2 will give a higher rank to words with posterior topic distributions that are very much different from the prior topic distribution. However, if a word has occurred only a few times, unreliable estimates for $p(c|w)$ will cause the KL measure to have unreliable values. In [175] the issue of smoothing is not raised, hence probably not implemented. In addition, an implicit assumption that is made in [175] is the choice of word representation. Here, we experiment with two choices: the binary representation, where each word is represented with 1 or 0 depending on whether it appeared or not in the document, and with the counts representation where the actual counts of each word in the document are used. Although not explicitly stated in [175], we assume that the binary representation was used in their experiments with I_w.

We have assessed the effect of these two factors (smoothing and representation) in the Fisher corpus [32], a collection of 5-minute telephone conversations on a predetermined topic, that was described in more detail in chapter 3. No default stopword list was used and

[1] According to Google Scholar

only words with 5 or more occurrences were kept. The vocabulary size was 23286. There are 20254 conversation sides in 40 topics and a 10-fold random 80/20 train/test split was used. Naive Bayes with Laplacian prior is used as the classification method. When smoothing was applied in feature selection, the number of times each word has been observed for each topic was incremented by 10. For the binary representation model the smoothed topic posterior distribution for word w is given by:

$$\hat{p}_b(c|w) = \frac{10 + n(c,w)}{10 * C + n(w)} \tag{4.4}$$

where $n(c,w)$ is the number of documents of topic c that word w is present and $n(w)$ is the number of documents word w is present. C is the number of topics. For the count representation, we calculate $p(c|w)$ through Bayes rule and estimate $p(w|c)$ using:

$$\hat{p}_m(w|c) = \frac{10 + \sum_{d \in I_c} c(w,d)}{10 * V + \sum_{w'} \sum_{d \in I_c} c(w',d)} \tag{4.5}$$

where $c(w,d)$ is the number of times word w has appeared in document d, I_c is the set of documents for topic c, and V is the vocabulary size.

The results are shown in Figure 4.1. From Figure 4.1 it is obvious that smoothing is a crucial factor in making KL a viable feature selection method. Comparing with Information Gain (IG), which is considered one of the top performing filter feature selection methods, KL (multinomial or binomial) achieves statistically the same accuracy (89.45% for KL compared to 89.29% for IG) but tends to hold the highest value for a wider range of features. On the other hand, when a very small number of features is desired (500 or less) IG is clearly better. Notice the crucial role of smoothing, despite the fact that we have limited our vocabulary to words with 5 or more occurrences and that the 80/20 train/test split results on average in 400 conversation sides per topic. Therefore, smoothing is expected to be even more important when there are fewer training data and/or we lower the number of occurrences threshold for the words in the vocabulary. The choice of representation (binary or counts) also impacts the results depending on the desired range but it is not as crucial as smoothing. Using I_w without smoothing as a filter feature selection method, resulted in performance worse than KL_w without smoothing. This is because for most words there will be at least a topic c for which $p(c|w) = 0$ therefore $p(c) \log p(c|w) = -\infty$. Therefore for

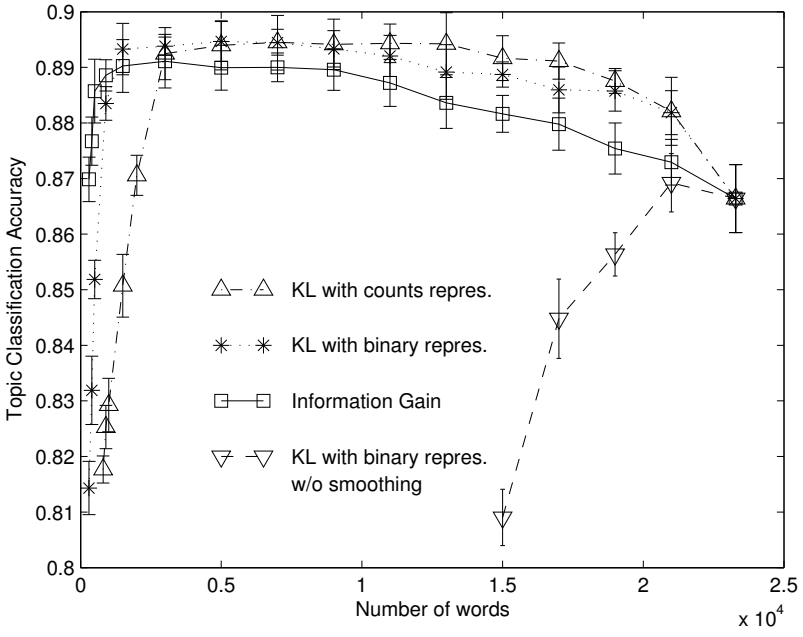

Figure 4.1: Comparative performance of various filter feature selection methods on the Fisher corpus. Naive Bayes with Laplace prior is used as the classification method.

most words $I_w = -\infty$. On the other hand KL_w has $p(c|w)\log p(c|w)$ terms so if $p(c|w) = 0$ then $p(c|w)\log p(c|w) = 0$.

Although not explicitly mentioned in [175], top performing methods, such as Information Gain have a built-in smoothing mechanism. For example, IG can be rewritten as:

$$IG_w = H(C) - p(w)H(C|w) - p(\bar{w})H(C|\bar{w}) \tag{4.6}$$

$$= -\sum_{c=1}^{C} p(c)\log p(c) - p(w)H(C|w) - p(\bar{w})H(C|\bar{w}) \tag{4.7}$$

$$= -\sum_{c=1}^{C} (p(w)p(c|w) + p(\bar{w})p(c|\bar{w}))\log p(c) - p(w)H(C|w) - p(\bar{w})H(C|\bar{w}) \tag{4.8}$$

$$= -p(w)\left[\sum_{c=1}^{C} p(c|w)\log p(c) + H(C|w)\right] - p(\bar{w})\left[\sum_{c=1}^{C} p(c|\bar{w})\log p(c) + H(C|\bar{w})\right] \tag{4.9}$$

$$= \quad p(w)KL_w + p(\bar{w})KL_{\bar{w}} \tag{4.10}$$

If a word w has appeared very few times then IG_w will be comprised mostly of $KL_{\bar{w}}$, i.e.

the KL-divergence of the distribution of the absence of w. Since w will be absent for almost all topics, $KL_{\bar{w}}$ will tend to be low. On the other hand, the KL measure puts all its weight on the observed instances of w. With proper smoothing, even words with a small number of occurrences can have high (but sensible) scores. Since this problem is present when scoring words, it will be exacerbated when scoring word pairs. This observation is the basis for the next contribution of this chapter, a modification to KL to augment the bag-of-words representation with selected word pairs.

4.3 Adding relevant and non-redundant bigrams

A problem with measures such as IG and KL is that they do not consider the interactions of features, rather they evaluate each feature independently. Therefore, they have no way of dealing with redundancy. To compensate for the fact that the KL-divergence cannot account for feature redundancy we define the new measure Redundancy-Compensated KL (RCKL) as:

$$RCKL_{w_i w_{i+1}} = KL_{w_i w_{i+1}} - KL_{w_i} - KL_{w_{i+1}} \tag{4.11}$$

In the KL-divergence the count representation is used, i.e. each document is represented as a vector of word counts. The word-topic distributions are smoothed by assuming that every word in the vocabulary is observed at least 10 times for each topic. With the RCKL measure, if a bigram is highly relevant, i.e. $KL_{w_i w_{i+1}}$ is high, but its unigrams are also highly relevant it will be less likely to get added. In words, equation (4.11) can be described as *How much more topic information can $w_i w_{i+1}$ give us compared to its unigrams?* To illustrate the basic idea, consider some examples from one of our data sets. For the topic *trials*, the words *commit* and *perjury* are deemed to be important for classification. The bigram *commit perjury*, although being by itself very much relevant, does not add further information than the words *commit* and *perjury*. As another example, the bigram *a holiday* is redundant given that the word *holiday* is already included in the feature subset. Examples of relevant and non-redundant bigrams would be *big brother* for the topic *reality shows*, or *second hand* for the topic *smoking*.

In [17], a measure is presented to augment the feature space representation with features

that are directly related to improved classification performance. The measure is termed Explaining Away Residual (EAR) and the presentation is for general classification problems. The EAR measure for feature pair XY is given by:

$$EAR_{XY} = I(X;Y|C) - I(X;Y) \tag{4.12}$$

where $I(X;Y)$ is the mutual information between random variables X and Y. The EAR and RCKL measures are closely related. To see this, using the KL definition of equation (4.2), the RCKL measure can be expressed as:

$$RCKL_{xy} = KL_{xy} - KL_x - KL_y \tag{4.13}$$

$$= \sum_c p(c|x,y) \log \frac{p(c|x,y)}{p(c)} - \sum_c p(c|x) \log \frac{p(c|x)}{p(c)} - \sum_c p(c|y) \log \frac{p(c|y)}{p(c)} \tag{4.14}$$

$$= \sum_c p(c|x,y) \log \frac{p(x,y|c)}{p(x,y)} - \sum_c p(c|x) \log \frac{p(x|c)}{p(x)} - \sum_c p(c|y) \log \frac{p(y|c)}{p(y)} \tag{4.15}$$

The EAR measure can be written as:

$$EAR_{xy} = \quad I(X;Y|C) - I(X;Y) = \sum_c I(X;Y|C=c) - I(X;Y) \tag{4.16}$$

$$= \sum_{x,y} \sum_c p(x,y,c) \log \frac{p(x,y|c)}{p(x|c)p(y|c)} - \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{4.17}$$

$$= \sum_{x,y} \sum_c p(x,y,c) \log \frac{p(x,y|c)p(x)p(y)}{p(x|c)p(y|c)p(x,y)} \tag{4.18}$$

$$= \sum_{x,y} \sum_c p(x,y,c) \left(\log \frac{p(x,y|c)}{p(x,y)} - \log \frac{p(x|c)}{p(x)} - \log \frac{p(y|c)}{p(y)} \right) \tag{4.19}$$

$$= \sum_{x,y} p(x,y) KL_{x,y} - \sum_x p(x) KL_x - \sum_y p(y) KL_y \tag{4.20}$$

Therefore the relation between EAR and RCKL is the same as the one between IG and KL. The reasons for preferring KL over IG apply for preferring RCKL over EAR. In section 4.4, we experimentally compare the two methods.

4.4 Experiments

4.4.1 Description of corpora used

We conducted experiments on three large corpora. The first is the Fisher corpus [32] that was described in more detail in chapter 3.

The second corpus is 20Newsgroups [94], a collection of 18827 postings to electronic discussion forums or newsgroups. There are 20 different classes in 20Newsgroups and the corpus is almost perfectly balanced, i.e. equal number of postings per newsgroup. Preprocessing consisted of converting all numbers to a single token and removing the *From:* field. Words with 5 or more occurrences were kept, resulting in a vocabulary of 34658 words.

The third corpus is a common subset of WebKB [34]. WebKB is a collection of html pages from different categories. For the feature selection experiments we selected 4 classes (faculty, student, project, course) of 4199 pages in total. This is a subset that has been used before [98]. Standard preprocessing was followed, such as keeping only the text of each web page and ignoring hyperlinks and headers and converting numbers to special tokens. The vocabulary of words with 2 or more occurrences consisted of 26087 words.

For 20Newsgroups and WebKB the best published topic classification results appear in [11]. The results in this chapter are not directly comparable to the results in [11] due to small variations of train/test splits (an 80/20 split is used in this chapter, while a 75/25 split in [11]) and different SVM training methodologies (one-vs-all is used in [11] vs. one-vs-one in this chapter, and the training error cost is optimized on a held out set in [11], while the default value is used here). However, it is acknowledged that SVM offer the best results in 20Newsgroups and WebKB. All three of the corpora are examples of single-label collections, i.e. each document is associated with a single class. A more general setting is a multi-label corpus where a document is associated with a set of classes, not necessarily of fixed length. Examples of multi-label corpora are Reuters-21758 and OHSUMED. Training multi-label classifiers was not investigated in this thesis.

4.4.2 Learning methods and evaluation measures

Two learning methods were used throughout our experiments: Naive Bayes [109] and Support Vector Machines (SVMs) [82]. The two methods are the most common used for text classification, with Naive Bayes representing a standard baseline and SVMs being the state-of-the-art method in text classification. Since our feature augmentation method is a filter approach, we would like to investigate how it performs for more than one classifier. For Naive

Bayes we used the *Rainbow* toolkit (http://www-2.cs.cmu.edu/mccallum/bow/rainbow/). For SVMs we used the *SVMLight* toolkit (http://svmlight.joachims.org/) with default values, i.e. the weight assigned to the training error is set to the variance of the training data and the radial basis function is used as the kernel. Since SVMs are inherently binary classifiers and *SVMLight* does not have implemented multi-class approaches to classification, we used the one-vs-one approach. In the one-vs-one approach, given a C-category classification problem, $C * (C-1)/2$ binary classifiers are constructed for every pair of classes. For each pair $\{i, j\}$ a function $H_{ij}(\vec{d})$ is estimated, where $\vec{d}$ is the vector representation of document d. During testing, if $H_{ij}(\vec{d}) > 0$ then $votes(i) = votes(i) + 1$ else $votes(j) = votes(j) + 1$. Document d is assigned to the class with the maximum number of votes $\hat{i} = \text{argmax}_i \, votes(i)$. SVMs require much larger computational resources than Naive Bayes, although both can be run in parallel on multiple machines. For Naive Bayes, the feature counts were used as input, while for the SVMs the tf·idf measure was used. During feature selection the counts representation is used.

Since we operate in a single-label setting, the class with the highest likelihood (for Naive Bayes) or number of votes (for SVMs) was selected as output. Classification accuracy was used as the evaluation measure. Micro-F, which is a common evaluation measure in text classification, is identical to classification accuracy for the single-label case.

4.4.3 Results

In all our experiments we used 10 random 80/20 train/test splits and averaged the classification accuracies over all splits. In Table 4.1 we see the performance of both learning methods, Naive Bayes and SVMs, for a varying number of unigrams selected according to KL (4.2) and bigrams selected according to RCKL (4.11). It is not always clear what criterion we should use to select the optimum number of features. One choice could be the highest classification accuracy on a held-out set. Another choice could be the ratio of classification accuracy and number of features, so that we prefer classifiers with low numbers of features. From Table 4.1 we see a clear gain from adding bigrams for both Naive Bayes and SVMs. Table 4.1 also reveals a smooth accuracy variation for different number of bigrams, therefore

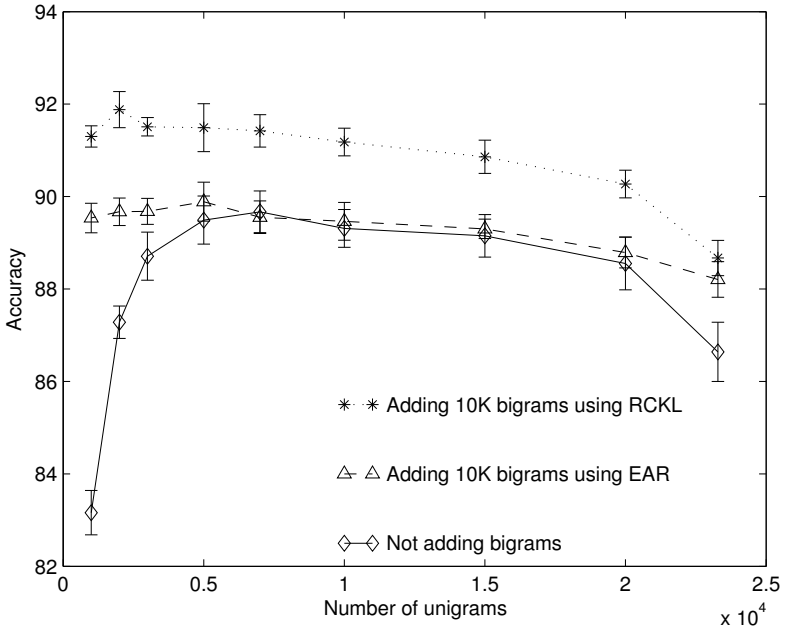

Figure 4.2: Naive Bayes performance with and without adding bigrams on the Fisher corpus.

having an automatic method for determining the number of bigrams should not be radically different from the optimum case. In Figures 4.2 and 4.3, we plot four columns of Table 4.1 with the associated standard deviations to show the difference between unigrams-only and the mix of unigrams and bigrams. In Figure 4.2 we also plot the accuracy of adding 10K bigrams according to the EAR criterion. It is clear that adding bigrams according to EAR is not as successful as using the RCKL measure. In Table 4.2, we see the performance of using bigrams-only. We observe that it is the combination of unigrams and bigrams that achieves the highest accuracy rather than unigrams-only or bigrams-only representations. In addition, from Table 4.1, we can see that by using 1K unigrams and 1K bigrams gives the same performance as 7K unigrams or 5K bigrams with Naive Bayes. This can be important when we want the most compact model for the fastest calculation and the smallest memory or disk footprint.

In Table 4.3 we see the performance of the feature augmentation method on the 20Newsgroups corpus. This corpus is qualitatively different from Fisher. Some of the documents

Table 4.1: 10-fold cross validation mean accuracies using a mix of unigrams and bigrams on the Fisher corpus. Bigrams are selected according to (4.11). Standard deviations are in 0.2-0.4 range. Horizontal axis is bigrams, vertical unigrams.

		0	**0.5K**	**1K**	**3K**	**5K**	**10K**	**20K**	**90K**
23286	**NB**	86.64	87.91	87.97	88.21	88.41	88.67	88.61	84.02
	SVM	90.84	91.33	91.28	91.87	91.38	91.22	91.53	90.61
20K	**NB**	88.55	89.25	89.31	89.95	90.25	90.27	90.12	84.62
	SVM	91.01	91.54	91.25	91.53	92.11	91.86	91.85	90.83
15K	**NB**	89.15	90.00	90.11	90.52	90.70	90.86	90.75	85.07
	SVM	91.07	91.19	91.76	91.83	92.18	91.76	91.48	90.39
10K	**NB**	89.31	90.09	90.46	90.53	91.07	91.18	91.38	85.08
	SVM	90.87	91.52	91.40	91.72	92.02	91.61	91.48	90.81
7K	**NB**	89.67	90.38	90.67	90.91	91.14	91.42	91.30	85.07
	SVM	90.61	91.33	91.35	91.43	91.94	91.76	91.73	90.73
5K	**NB**	89.49	90.57	90.70	91.10	91.34	91.49	91.46	85.15
	SVM	90.26	90.86	91.24	91.39	91.67	91.72	91.60	90.30
3K	**NB**	88.71	90.34	90.75	90.97	91.26	91.51	91.45	84.55
	SVM	89.32	90.50	91.11	91.49	91.44	91.65	91.52	90.21
2K	**NB**	87.28	90.16	90.46	90.97	91.38	91.88	91.64	84.29
	SVM	87.63	90.17	90.23	90.93	91.40	91.58	91.48	90.00
1K	**NB**	83.16	88.94	89.87	90.62	91.02	91.30	91.47	83.58
	SVM	80.96	88.90	89.44	90.57	90.95	90.78	90.11	89.88

Table 4.2: 10-fold cross validation mean accuracies using only bigrams on the Fisher corpus. Bigrams are ranked according to $KL_{w_i w_{i+1}}$. Standard deviations are in the range 0.2-0.4

	1K	**5K**	**10K**	**20K**	**50K**	**100K**	**150K**	**230K**
NB	85.69	89.00	89.91	90.63	90.71	89.61	87.35	73.60
SVM	80.01	88.25	89.75	90.42	91.02	90.19	90.11	90.23

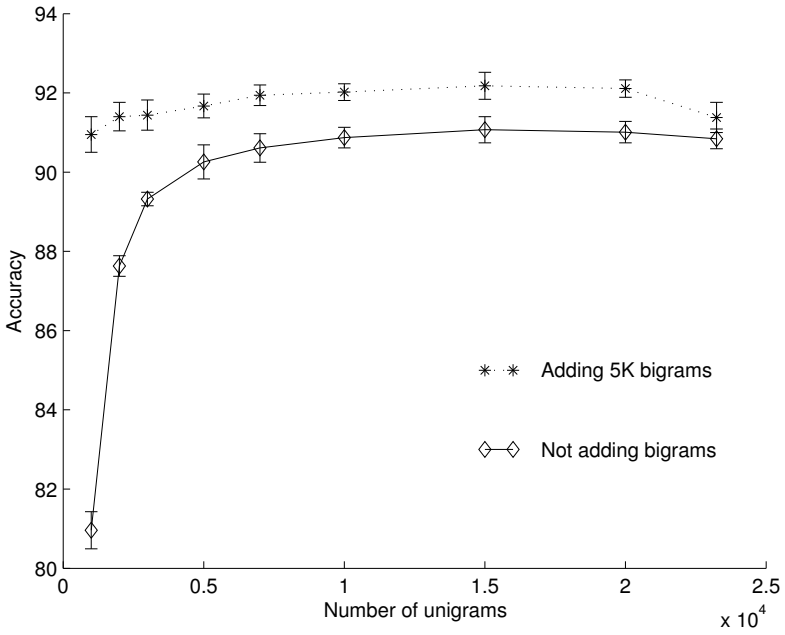

Figure 4.3: SVM performance with and without adding bigrams on the Fisher corpus.

are very small (42 with 5 or fewer words, and 93 with 10 or fewer words) and the vocabulary is much bigger than Fisher's (34658 vs. 23286). Applying feature selection on unigrams resulted in a slight increase of classification accuracy for up to 30K features and then a constant degradation of performance. The degradation was even worse if IG was used as the feature selection method. In such a task where feature selection does not appear to be important, Naive Bayes did not benefit from augmenting its feature space with bigrams. Performance did not degrade either, which shows that the added features are relevant, given the sensitivity that Naive Bayes has to high-dimensional spaces. SVM gets a small boost of performance by integrating bigrams in the feature space. Using bigrams only did not provide a superior alternative either, as shown in Table 4.4.

In Table 4.5 we see the performance of the feature augmentation method on the WebKB corpus. Here feature selection appears to be more important than in 20Newsgroups for both Naive Bayes and SVMs, even if the vocabulary is much smaller. Adding bigrams offers gains for both Naive Bayes and SVMs. In Table 4.6 we see the performance using bigrams

Table 4.3: 10-fold cross validation mean accuracies using a mix of unigrams and bigrams on the 20Newsgroups corpus. Bigrams are selected according to (4.11). Standard deviations are in 0.2-0.4 range. Horizontal axis is bigrams, vertical unigrams.

		0	**0.5K**	**1K**	**5K**	**10K**	**20K**	**50K**
34658	**NB**	89.16	89.20	89.14	89.31	89.52	89.41	89.52
	SVM	90.13	90.84	90.93	90.86	91.02	91.13	91.08
30K	**NB**	89.72	88.98	89.36	89.70	89.70	89.34	89.52
	SVM	90.73	90.81	91.14	91.05	91.24	91.27	90.84
25K	**NB**	89.34	89.40	89.47	89.41	89.67	89.42	89.39
	SVM	91.04	90.93	91.08	91.05	91.50	91.26	91.21
20K	**NB**	89.02	88.85	89.08	89.38	89.92	89.67	89.50
	SVM	90.49	91.02	91.02	91.20	91.51	91.38	90.95
15K	**NB**	88.66	88.25	88.41	89.06	89.54	89.30	89.05
	SVM	90.35	90.37	90.73	90.63	91.42	90.87	90.81
10K	**NB**	87.73	87.44	88.01	88.45	89.15	88.86	89.11
	SVM	89.23	89.96	90.13	90.40	90.66	90.55	90.34
5K	**NB**	85.67	85.96	85.98	87.04	87.72	87.58	88.11
	SVM	82.30	83.05	86.77	89.13	89.79	89.81	89.77

Table 4.4: 10-fold cross validation mean accuracies using only bigrams on the 20Newsgroups corpus. Bigrams are ranked according to $KL_{w_i w_{i+1}}$. Standard deviations are in the range 0.2-0.4.

	5K	**10K**	**15K**	**20K**	**30K**	**50K**	**100K**	**135K**
NB	80.14	82.08	83.39	84.23	85.42	86.64	87.14	86.14
SVM	N/A	N/A	75.60	81.17	85.03	86.66	87.30	86.75

Table 4.5: 10-fold cross validation mean accuracies using a mix of unigrams and bigrams on the WebKB corpus. Bigrams are selected according to (4.11). Standard deviations are in the 0.6-1.2 range. Horizontal axis is bigrams, vertical unigrams.

		0	**0.5K**	**1K**	**2K**	**5K**	**10K**	**20K**	**50K**
26087	**NB**	85.44	86.02	86.50	87.37	88.01	87.53	87.97	87.70
	SVM	90.12	91.51	91.33	91.10	90.89	91.03	91.26	90.60
20K	**NB**	85.21	86.90	87.47	87.88	87.52	87.95	88.09	87.44
	SVM	90.51	92.00	91.37	90.79	90.75	91.25	90.82	90.58
15K	**NB**	85.61	86.70	86.64	87.47	88.10	87.69	88.53	88.00
	SVM	90.45	91.75	91.31	91.42	91.52	91.18	91.17	91.24
10K	**NB**	84.98	86.57	87.70	87.66	88.12	87.90	88.37	87.72
	SVM	90.91	91.56	91.49	91.61	91.51	92.08	91.74	91.00
5K	**NB**	86.78	89.22	88.65	89.17	88.52	88.59	88.40	88.08
	SVM	91.35	91.71	91.26	91.86	91.68	91.85	91.37	91.21
2K	**NB**	87.25	89.16	89.64	89.47	89.67	89.28	88.64	89.21
	SVM	91.41	91.91	92.08	92.07	92.47	92.28	92.59	91.77
1K	**NB**	87.01	89.61	90.28	90.05	89.77	89.59	89.35	88.67
	SVM	89.79	92.23	92.61	92.84	93.02	93.00	92.06	91.75
0.5K	**NB**	81.75	88.33	89.36	90.10	89.78	89.26	88.69	88.84
	SVM	N/A	N/A	90.95	91.25	91.78	92.17	91.74	91.11

only. Naive Bayes achieves better results than using unigrams only but SVMs performance is about the same. Overall, the best text classification accuracy for WebKB is obtained by augmenting the bag-of-words space with bigrams, from 91.62 to 93.02 with standard deviation being 0.81 for both.

In Table 4.7 a summary of the results is shown. The highest classification accuracies using each one of the three feature construction methods are shown. It should be noted that in practice a scheme to automatically estimate the number of features should be applied. Table 4.7 shows that 5 out of 6 times the augmented space is better than the bag-of-words

Table 4.6: 10-fold cross validation mean accuracies using only bigrams on the WebKB corpus. Bigrams are ranked according to $KL_{w_i w_{i+1}}$. Standard deviations are in the range 0.6-1.2

	1K	**2K**	**3K**	**5K**	**10K**	**20K**	**50K**	**70K**	**110K**
NB	89.22	89.96	90.39	89.95	90.06	90.12	89.51	89.40	88.31
SVM	33.73	65.27	90.70	91.51	91.62	91.41	91.11	91.38	89.14

Table 4.7: Summary results from all corpora. The best accuracies for each feature construction method are shown. Student's t-test is performed to assess the significance of difference. The last two symbols show if the performance of the augmented representation is statistically different than the unigrams-only and bigrams-only representation respectively at the confidence level of 0.95. A (+) symbol means that the augmented representation is better and a (=) symbol means that the difference is not significant.

		Only 1-grams	**Only 2-grams**	**Mix of 1-grams, 2-grams**		
Fisher	**NB**	89.67	90.71	91.88	(+)	(+)
	SVM	91.07	91.02	92.18	(+)	(+)
20Newsgroups	**NB**	89.72	87.14	89.92	(=)	(+)
	SVM	91.04	87.30	91.51	(+)	(+)
WebKB	**NB**	87.25	90.39	90.28	(+)	(=)
	SVM	91.42	91.62	93.02	(+)	(+)

space, and 5 out of 6 times better than the bigrams-only space. In no occasion was the augmented space worse than either of the representations, considering experiments on all three corpora and learning methods. For the SVMs method (which gave the best results), the augmented space is always better than either individual space.

4.5 Discussion

In this chapter, we have shown how an old and previously dismissed filter feature selection method can be cast under new light with small modifications. The measure has the property

of assigning high (but sensible) scores to words that were observed few times, unlike IG which would tend to assign low scores to words that occur a small number of times. This property is crucial for designing a measure that can incorporate word pairs or even higher-order word sequences, since the average number of occurrences of a word pair is much lower than that of a word. Using the new RCKL measure we have shown that incorporating selected word pairs offers improvements over the bag-of-words representation, across a variety of corpora and learning methods. Key to the new representation is that the added bigrams are compensated for redundancy. A word pair is added according to how much more information it brings compared to its words. Therefore, word pairs such as *a holiday*, *the holiday* will not be preferred given that *holiday* is already in the feature set. This work may help dismiss the myth that more complex representations do not help text classification. The implicit assumption was that the bag-of-words representation captures enough of topic information and more complex representations are hard to model, since they considerably increase the dimensionality of the feature space. Moreover, previous attempts to use more complex features were not successful. As a result of this fallacy, research in text classification has mostly focused on learning methods and not on vector representations. The suggested method offers some evidence that design of feature spaces for text classification can be more important than previously considered.

It would be interesting to connect the suggested criterion with the model selection literature. In this chapter we used an ad-hoc way for identifying non-redundant bigrams. Is there an "optimal" compensation term that could be added when considering the redundancy of a bigram, as in the Bayesian Information Criterion (BIC) or Akaike Information Criterion (AIC)? This formulation may help extend this criterion in a natural way to higher order n-grams.

PART II
TOPIC LEARNING IN CONVERSATIONAL SPEECH

The second part of the dissertation is concerned with topic learning in human-human conversations. Classifying or clustering human-human conversations to topics can be an important part in a number of applications ranging from analyzing business meetings to customer call-centers. It should be stressed that topic learning in speech need not be a mere concatenation of existing methodologies in automatic speech recognition and topic learning in text. Although the success of a speech topic learning system depends on the success of the underlying core technologies, conversational speech offers unique research opportunities. Even if the underlying technologies have reached perfection we would still be missing an important part of the information by ignoring the prosodic aspects of the speech signal, i.e. *how* we speak in addition to *what* we say. Moreover, bringing the core technologies to perfection may not actually be needed in many cases. For example, in a call-routing application, a WER of the ASR component of 30% degrades minimally the routing classification performance [64].

Conversational speech brings a number of new and exciting issues in the topic learning problem. In chapter 5, the issue of interaction between content and style is investigated, specifically lexical differences between genders in conversational speech and how they affect topic classification. In chapter 6, the role of linguistic phenomena associated with conversational speech is investigated. Specifically, the impact of disfluencies on topic classification is assessed. In chapter 7, prosody and language are used in conjunction for topic learning. The output of a symbolic prominence classifier is used to design feature subsets for topic classification and clustering of conversations. Finally, in chapter 8, words are clustered according to a confusability measure derived from an ASR system and the derived word clusters are used for topic learning.

Chapter 5

A QUANTITATIVE ANALYSIS OF LEXICAL DIFFERENCES BETWEEN GENDERS IN TELEPHONE CONVERSATIONS

In this chapter, we provide an empirical analysis of differences in word use between genders in telephone conversations, which complements the considerable body of work in sociolinguistics concerned with gender linguistic differences. Experiments are performed on a large speech corpus of roughly 12000 conversations. We employ machine learning techniques to automatically categorize the gender of each speaker given only the transcript of his/her speech, achieving 92% accuracy. An analysis of the most characteristic words for each gender is also presented. Experiments reveal that the gender of one conversation side influences lexical use of the other side. Unfortunately, we were not able to leverage these differences to improve topic classification or speech recognition.

5.1 Introduction

Linguistic and prosodic differences between genders in American English have been studied for decades. The interest in analyzing the gender linguistic differences is two-fold. From the scientific perspective, it will increase our understanding of language production. From the engineering perspective, it can potentially help improve the performance of a number of natural language processing tasks, such as text classification, machine translation or automatic speech recognition by training better language models. Traditionally, these differences have been investigated in the fields of sociolinguistics and psycholinguistics, see for example [33], [45] or http://www.ling.lancs.ac.uk/groups/gal/genre.htm for a comprehensive bibliography on language and gender. Sociolinguists have approached the issue from a mostly non-computational perspective using relatively small and very focused data collections. Recently, the work of [91] has used computational methods to characterize the

differences between book authors' gender in written text, such as literary books. A number of monologues have been analyzed in [148] in terms of lexical richness using multivariate analysis techniques. The question of gender linguistic differences shares a number of issues with stylometry and author/speaker attribution research [152], [41], but novel issues emerge with analysis of conversational speech, such as studying the interaction of genders.

In this chapter, we focus on lexical differences between genders on telephone conversations and use machine learning techniques applied on text categorization and feature selection to characterize these differences. Therefore our conclusions are entirely data-driven. We use a very large corpus created for automatic speech recognition - the Fisher corpus described in [32]. The Fisher corpus is annotated with the gender of each speaker making it an ideal resource to study not only the characteristics of individual genders but also of gender pairs in spontaneous, conversational speech. The size and scope of the Fisher corpus is such that robust results can be derived for American English. The computational methods we apply can assist us in answering questions, such as *"To which degree are gender-discriminative words content-bearing words?"* or *"Which words are most characteristic for males in general or males talking to females?"*.

In section 5.2, we describe the corpus we have based our analysis on. In section 5.3, the machine learning tools are explained, while the experimental results are described in section 5.4 with a specific research question for each subsection. We conclude in section 5.6 with a summary and future directions.

5.2 Data Preparation

The Fisher corpus [32] was used in the gender analysis experiments. Processing of the corpus was described in Chapter 3. It should be noted that about 10% of speakers are non-native, making this corpus suitable for investigating their lexical differences compared to American English speakers.

The gender distribution of the Fisher corpus is 53% female and 47% male. Age distribution is 38% 16-29, 45% 30-49% and 17% 50+. Speakers were connected at random from a pool recruited in a national ad campaign. It is unlikely that the speakers knew their

conversation partner. All major American English dialects are well represented, see [32] for more details. The Fisher corpus was primarily created to facilitate automatic speech recognition research. The subset we have used has about 17.8M words and it is the largest resource ever used to analyze gender linguistic differences. In comparison, [148] has used about 30,000 words for their analysis.

Before attempting to analyze the gender differences, there are two main biases that need to be removed. The first bias, which we term the *topic bias* is introduced by not accounting for the fact that the distribution of topics in males and females is uneven, despite the fact that the topic is pre-assigned randomly. For example, if topic A happened to be more common for males than females and we failed to account for that, then we would be implicitly building a topic classifier rather than a gender classifier. Our intention here is to analyze gender linguistic differences controlling for the topic effect as if both genders talk equally about the same topics. The second bias, which we term *speaker bias* is introduced by not accounting for the fact that specific speakers have idiosyncratic expressions. If our training data consisted of a small number of speakers appearing in both training and testing data, then we will be implicitly modeling speaker differences rather than gender differences.

To normalize for these two important biases, we made sure that both genders have the same percent of conversation sides for each topic and there are 8899 speakers in training and 2000 in testing with no overlap between the two sets. After these two steps, there were 14969 conversation sides used for training and 3738 sides for testing. The median length of a conversation side was 954. It is possible that there is still some degree of topic bias if there is a strong dependence between topic and gender effects in language, this issue will be investigated more in section 5.5.

5.3 Machine Learning Methods Used

The methods we have used for characterizing the differences between genders and gender pairs are similar to what has been used for the task of text classification. We chose two ways for characterizing the differences between gender categories. The first, is to classify the transcript of each speaker, i.e. each conversation side, to the appropriate gender category.

This approach can show the cumulative effect of all terms on the distinctiveness of gender categories. The second approach is to apply feature selection methods, similar to those used in text categorization, to reveal the most characteristic features for each gender.

Classifying a transcript of speech according to gender can be done with a number of different learning methods. We have compared Support Vector Machines (SVMs), Naive Bayes, Maximum Entropy and the tfidf/Rocchio classifier and found SVMs to be the most successful. In addition to classification, we have applied two feature selection techniques, information gain (see equation 2.2) and KL-distance (see equation 4.2), to assess the discriminative ability of each individual feature.

5.4 Analysis of gender differences

Having explained the methods and data that we have used, we set forward to investigate a number of research questions concerning the nature of differences between genders. Each subsection is concerned with a single question.

5.4.1 Given only the transcript of a conversation, is it possible to classify conversation sides according to the gender of the speaker?

The first hypothesis we investigate is whether simple features, such as counts of individual terms (unigrams) or pairs of terms (bigrams) have different distributions between genders. The set of possible terms consists of all words in the Fisher corpus plus some non-lexical tokens such as laughter and filled pauses. One way to assess the difference in their distribution is by attempting to classify conversation sides according to the gender of the speaker. The results are shown in Table 5.1, where a number of different text classification algorithms were applied to classify conversation sides. 14,969 conversation sides are used for training and 3,738 sides are used for testing. No feature selection was performed; in all classifiers a vocabulary of all unigrams or bigrams with 5 or more occurrences is used (20,513 for unigrams, 306,779 for bigrams). For all algorithms, except Naive Bayes, we have used the tf·idf representation. The *Rainbow* toolkit [107] was used for training the classifiers. Results show that differences between genders are clear and the best results are obtained by using

SVMs. The fact that classification performance is significantly above chance for a variety of learning methods shows that lexical differences between genders are inherent in the data.

From Table 5.1 we also observe that using bigrams is consistently better than unigrams, despite the fact that the number of unique terms rises from ∼20K to ∼300K. This suggests that gender differences become even more profound for phrases, a finding similar to [41] for speaker differences.

Table 5.1: Classification accuracy of different learning methods for the task of classifying the transcript of a conversation side according to the gender - male/female - of the speaker.

	Unigrams	**Bigrams**
Rocchio	76.3	86.5
Naive Bayes	83.0	89.2
MaxEnt	85.6	90.3
SVM	88.6	92.5

5.4.2 Does the gender of a conversation side influence lexical usage of the other conversation side?

Each conversation always consists of two people talking to each other. Up to this point, we have only attempted to analyze a conversation side in isolation, i.e. without using transcriptions from the other side. In this subsection, we attempt to assess the degree to which, if any, the gender of one speaker influences the language of the other speaker. In the first experiment, instead of defining two categories we define four; the Cartesian product of the gender of the target speaker and the gender of the other speaker. These categories are symbolized with two letters: the first characterizing the gender of the target speaker and the second the gender of the other speaker, i.e. FF, FM, MF, MM. The task remains the same: given the transcript of a conversation side, classify it according to the appropriate category. This is a task much harder than the binary classification we had in subsection 5.4.1, because given only the transcript of a conversation side we must make inferences about

the gender of the current as well as the other conversation side. We have used SVMs as the learning method. In their basic formulation, SVMs are binary classifiers (although there has been recent work on multi-class SVMs). We followed the original binary formulation and converted the 4-class problem to six 2-class problems. The final decision is taken by voting of the individual systems. The confusion matrix of the 4-way classification is shown in Table 5.2.

Table 5.2: Confusion matrix for 4-way classification of gender of both sides using transcripts from one side. Unigrams are used as features, SVMs as classification method. Each row represents the true category and each column the hypothesized category.

	FF	**FM**	**MF**	**MM**
FF	1447	30	40	65
FM	456	27	43	77
MF	167	25	104	281
MM	67	44	210	655

The results show that although two of the four categories, FF and MM, are quite robustly detected, the other two, FM and MF, are mostly confused with FF and MM respectively. These results can be mapped to single gender detection, giving accuracy of 85.9% for classifying the gender of the given transcript. This is lower than the 88.6% accuracy in Table 5.1, but this can be attributed to the 4-class task of Table 5.2 compared to the binary classification task of Table 5.1. From Table 5.2 we also obtain a 68.5% accuracy for classifying the gender of the conversational partner. The accuracy of 68.5% is higher than chance (57.8%) showing that genders alter their linguistic patterns depending on the gender of their conversational partner.

In the next experiment, we design two binary classifiers. In the first classifier, the task is to correctly classify FF vs. MM transcripts, and in the second classifier the task is to classify FM vs. MF transcripts. Therefore, we attempt to classify the gender of a speaker given knowledge of whether the conversation is same-gender or cross-gender. For both classifiers 4526 sides were used for training equally divided among each class. 2558 sides were used

for testing of the FF-MM classifier and 1180 sides for the FM-MF classifier. The results are shown in Table 5.3.

Table 5.3: Classification accuracies in same-gender and cross-gender conversations. SVMs are used as the classification method; no feature selection is applied. Chance is 50%.

	Unigrams	**Bigrams**
FF-MM	98.91	99.49
FM-MF	69.15	78.90

It is clear from Table 5.3 that there is a significant difference in performance between the FF-MM and FM-MF classifiers, suggesting that people alter their linguistic patterns depending on the gender of the person they are talking to. In same-gender conversations, almost perfect accuracy is reached, indicating that the linguistic patterns of the two genders, become very distinct. In cross-gender conversations the differences become less prominent since classification accuracy drops compared to same-gender conversations. The difference though is significantly higher than chance. This result, however, does not reveal how this convergence of linguistic patterns is achieved. Is it the case that the convergence is attributed to one of the genders, for example males attempting to match the patterns of females, or is it collectively constructed? To answer this question, we can examine the classification performance of two other binary classifiers FF vs. FM and MM vs. MF. The results are shown in Table 5.4. In both classifiers 4608 conversation sides are used for training, equally divided in each class. The number of sides used for testing is 989 and 689 for the FF-FM and MM-MF classifier respectively.

Table 5.4: Classifying the gender of a speaker given only the transcripts and gender of the conversational partner. Chance is 50%.

	Unigrams	**Bigrams**
FF-FM	57.94	59.66
MM-MF	60.38	59.80

The results in Table 5.4 suggest that both genders equally alter their linguistic patterns to match the opposite gender. It is interesting to see that the gender of a speaker can be detected better than chance given only the transcript and gender of the other speaker, e.g. for FF-FM we are given the transcript of a female speaker and must decide the gender of the conversational partner. The results are better than chance at the 0.0005 significance level.

5.4.3 Are some features more indicative of gender than other?

Having shown that gender lexical differences are prominent enough to classify each speaker according to gender quite robustly, another question is whether the high classification accuracies can be attributed to a small number of features or are rather the cumulative effect of a high number of them. In Table 5.5 we apply the two feature selection criteria, IG and KL, that were described in chapter 4.

Table 5.5: Effect of feature selection criteria on gender classification using SVM as the learning method. Horizontal axis refers to the fraction of the original vocabulary size (∼20K for unigrams, ∼300K for bigrams) that was used.

		1.0	**0.7**	**0.4**	**0.1**	**0.03**
KL	**1-gram**	88.6	88.8	87.8	86.3	85.6
	2-gram	92.5	92.6	92.2	91.9	90.3
IG	**1-gram**	88.6	88.5	88.9	87.6	87.0
	2-gram	92.5	92.4	92.6	91.8	90.8

The results of Table 5.5 show that lexical differences between genders are not isolated to a small set of words. The best results are achieved with 40% (IG) and 70% (KL) of the features; using fewer features steadily degrades the performance. Using the 5,000 least discriminative unigrams and Naive Bayes as the classification method resulted in 58.4% classification accuracy which is not statistically better than chance (this is the test set of Tables 5.1 and 5.2 not of Tables 5.3 or 5.4). Using the 15,000 least useful unigrams resulted in a classification accuracy of 66.4%, which shows that the number of irrelevant features is

rather small, about 5,000 features.

It is also instructive to see which features are most discriminative for each gender. The features that when present are most indicative of each gender (positive features) are shown in Table 5.6. They are sorted using the KL distance and dropping the summation over both genders. Looking at the top 2,000 features for each gender, we observed that a number of

Table 5.6: The 10 most discriminative words for each gender according to KL distance. Words higher in the list are more discriminative.

Male	**Female**
dude	*husband*
shit	*husband's*
fucking	*refunding*
wife	*goodness*
wife's	*boyfriend*
matt	*coupons*
steve	*crafts*
bass	*linda*
ben	*gosh*
fuck	*cute*

swear words appear as most discriminative for males and family-relation terms are often associated with females. For example, the following words are in the top 2,000 (out of 20,513) most useful features for males *shit, bullshit, shitty, fuck, fucking, fucked, bitching, bastards, ass, asshole, sucks, sucked, suck, sucker, damn, goddamn, damned*. The following words are in the top 2,000 features for females *children, grandchild, child, grandchildren, childhood, childbirth, kids, grandkids, son, grandson, daughter, granddaughter, boyfriend, marriage, mother, grandmother*. It is also interesting to note that a number of non-lexical tokens are strongly associated with a certain gender. For example, *[laughter]* and acknowledgments/backchannels such as *uh-huh,uhuh* were in the top 2000 features for females. On the other hand, filled pauses such as *uh* were strong male indicators. Our analysis also

reveals that a high number of useful features are names. A possible explanation is that people usually introduce themselves at the beginning of the conversation. In the top 30 words per gender, names represent over half of the words for males and nearly a quarter for females. Nearly a third were family-relations words for females, and 17% were swear words for males.

In Table 5.7, the 10 most discriminative bigrams (word pairs) for each gender are displayed. For some of the bigrams it is intuitively clear that they are strongly associated with a single gender, such as *my wife* or *my husband*. It is interesting to note that for some words you need context to associate them with a gender. The word pair *hello hey* is strongly associated with females but the pair *hey what's* is strongly associated with males.

Table 5.7: The 10 most discriminative word pairs for each gender according to KL distance. Word pairs higher in the list are more discriminative.

Male	**Female**
my wife	*my husband*
my wife's	*my husband's*
what's up	*hello hey*
how's it	*my boyfriend*
yeah man	*husband and*
I'm John	*husband was*
hey what's	*husband is*
hey how	*my goodness*
wife is	*uh pretty*
is Mike	*my gosh*

When examining cross-gender conversations, the discriminative words were quite substantially different. We can quantify the degree of change by measuring $KL_{SG}(w) - KL_{CG}(w)$, where $KL_{SG}(w)$ is the KL measure of word w for same-gender conversations. The analysis reveals that swear terms are highly associated with male-only conversations, while family-relation words are highly associated with female-only conversations.

From the traditional sociolinguistic perspective, these methods offer a way of discovering rather than testing words or phrases that have distinct usage between genders. For example, in a recent paper [86] the word *dude* is analyzed as a male-to-male indicator. In our study, the word *dude* emerged as a male feature. As another example, our observation that some acknowledgments and backchannels (*uh-huh*) are more common for females than males while the reverse is true for filled pauses asserts a popular theory in sociolinguistics that males assume a more dominant role than females in conversations [33]. Males tend to hold the floor more than women (more filled pauses), and females tend to be more responsive (more acknowledgments/backchannels).

5.4.4 Are gender-discriminative features content-bearing words?

Do the most gender-discriminative words contribute to the topic of the conversation, or are they simple fill-in words with no content? Since each conversation is labeled with one of 40 possible topics, we can rank features with IG or KL using topics instead of genders as categories. In fact, this is the standard way of performing feature selection for text classification. We can then compare the performance of classifying conversations to topics using the top-N features according to the gender or topic ranking. The results are shown in Table 5.8.

Table 5.8: Topic classification accuracies using topic- and gender-discriminative words, sorted using the information gain criterion. Naive Bayes with Laplace prior is used as the classification method. When randomly selecting 5000 features, 10 independent runs were performed and numbers reported are mean and standard deviation. Using the bottom 5000 topic words resulted in chance performance (∼5.0)

	Top 5K	**Bottom 5K**	**Random 5K**
Gender ranking	78.51	66.72	74.99±2.2
Topic ranking	87.72	-	74.99±2.2

From Table 5.8 we can observe that gender-discriminative words are clearly not the most relevant nor the most irrelevant features for topic classification. They are slightly

more topic-relevant features than topic-irrelevant but not by a significant margin. The bottom 5000 features for gender discrimination are more strongly topic-irrelevant words.

These results show that gender linguistic differences are not merely isolated in a set of words that would function as markers of gender identity but are rather closely intertwined with semantics.

5.5 Applications

Are the observed gender linguistic differences valuable from an engineering perspective as well? In other words, can a natural language processing task benefit from modeling these differences? In this section, we attempt to leverage gender lexical differences for two tasks, automatic speech recognition and topic classification.

5.5.1 Can gender lexical differences be exploited to improve automatic speech recognition?

In this subsection, we train gender-dependent language models and compare their perplexities with standard baselines. An advantage of using gender information for automatic speech recognition is that it can be robustly detected using acoustic features. The perplexities of different gender-dependent language models are shown in Tables 5.9 and 5.10. The SRILM toolkit [153] was used for training the language models using Kneser-Ney smoothing [88]. The perplexities reported include the end-of-turn as a separate token. 2300 conversation sides are used for training each one of {FF,FM,MF,MM} models of Table 5.9, while 7670 conversation sides are used for training each one of {F,M} models of Table 5.10. In both tables, the same 1678 sides are used for testing.

In Tables 5.9 and 5.10 we observe that we get lower perplexities in matched than mismatched conditions in training and testing. This is another way to show that different data do exhibit different properties. However, the best results are obtained by pooling all the data and training a single language model. Therefore, despite the fact there are different modes, the benefit of more training data outweighs the benefit of gender-dependent models, when using a simple data partitioning training strategy. Interpolating ALL with F and ALL with M resulted in insignificant improvements (81.6 for F and 89.3 for M).

Table 5.9: Perplexity of gender-dependent bigram language models. Four gender categories are used. Each column has the perplexities for a given test set, each row for a train set.

	FF	**FM**	**MF**	**MM**
FF	**85.3**	91.1	96.5	99.9
FM	85.7	**90.0**	94.5	97.5
MF	87.8	91.4	**93.3**	95.4
MM	89.9	93.1	94.1	**95.2**
ALL	**82.1**	**86.3**	**89.8**	**91.7**

Table 5.10: Perplexity of gender-dependent bigram language models. Two gender categories are used. Each column has the perplexities for a given test set, each row for a train set.

	F	**M**
F	**82.8**	94.2
M	86.0	**90.6**
ALL	**81.8**	**89.5**

5.5.2 Can gender lexical differences be exploited to improve topic classification?

In this subsection, gender information is incorporated into a topic classification system with the goal of improving the topic classification performance. There are two possible steps where gender information can prove useful: the topic models, i.e. $p(word|topic)$, and feature selection. Gender-dependent topic models were estimated, i.e. $p(word|topic, gender)$ as well as gender-dependent feature selection, i.e. split the data according to gender and estimate separate feature selection. The results are shown in Table 5.11, and we can see that topic classification performance is not benefited by the gender information, neither in the topic models nor in feature selection. The best numbers for each row is shown in bold.

Table 5.11: Topic classification accuracy using different topic models (TM) and feature selection (FS). GI stands for gender-independent and GD for gender-dependent. Naive Bayes with Laplace prior is used as the classification method, IG as the feature selection method.

TM,FS	**20513**	**15K**	**10K**	**5K**	**2K**	**0.75K**
GI,GI	85.97	86.71	87.38	87.72	**88.29**	88.19
GD,GI	82.14	84.04	85.86	87.08	87.97	**88.21**
GI,GD	85.97	86.19	87.20	87.48	**88.14**	87.97
GD,GD	82.14	83.94	86.06	87.03	87.79	**87.85**

5.6 Conclusions

We have presented evidence of linguistic differences between genders using a large corpus of telephone conversations. We have approached the issue from a purely computational perspective and have shown that differences are profound enough that we can classify the transcript of a conversation side according to the gender of the speaker with accuracy close to 93%. Our computational tools have allowed us to quantitatively show that the gender of one speaker influences the linguistic patterns of the other speaker. Specifically, classifying same-gender conversations can be done with almost perfect accuracy, while evidence of some convergence of male and female linguistic patterns in cross-gender conversations was observed. An analysis of the features revealed that the most characteristic features for males are swear words while for females are family-relation words. Leveraging these differences in simple gender-dependent language or topic classification models is not a win, but this does not imply that more sophisticated training methods cannot help. For example, instead of conditioning every word in the vocabulary on gender we can choose to do so only for the top-N, determined by KL or IG. The probability estimates for the rest of the words will be tied for both genders.

Chapter 6

THE ROLE OF DISFLUENCIES IN TOPIC CLASSIFICATION OF HUMAN-HUMAN CONVERSATIONS

In this chapter[1], we investigate the impact of disfluencies on the task of classifying natural human-human conversations into topics. Disfluencies are distinctive to spoken language, and their effect on a number of spoken language understanding tasks, including spoken language classification, remains largely unknown. We use a subset of Switchboard-I annotated for disfluencies and topics, and investigate the effect of different disfluency categories with both true and automatically generated transcripts. We show that under the popular bag-of-words representation, even perfect disfluency filtering has a minimal impact on topic classification performance on hand-transcribed data. Differences are somewhat larger with more complex representations (e.g. bigrams) and for some classifiers operating on recognizer transcripts. However, we find that proper choice of classifier is more important than disfluency removal.

6.1 Disfluencies

Disfluencies occur amply in spoken language [143], and although at the surface they appear to interrupt the flow of information, human listeners typically have little trouble understanding disfluent speech. For automatic language processing though, disfluencies falsely increment the counts of words, and since the most prevalent representation for topic classification is the bag-of-words, they can potentially have an adverse effect on conversation classification. In the past, attempts have been made to detect disfluencies in conversations [102]. Removal of disfluencies has been shown to increase the readability of conversation transcripts [84] and detecting and removing repetitions, a certain type of disfluency, has been

[1] Jeremy G. Kahn contributed to the content of this chapter by providing scripts that extract the disfluent segments of text and by useful discussions.

used to produce more natural summaries of spoken dialogues [180]. In addition, handling of disfluencies is important at the grammar component of an SLU system [170].

We decompose disfluencies to five categories, similar to [143], and study the effect of different groups of them. The five categories, with an example for each, are shown below. In edit disfluencies, the + sign marks the starting point of the correction or new sentence.

- **Fillers**. *Uh, well like, one week she'll work three days and I'll work two.* In this example, we see two kind of fillers: filled pauses (*uh*) and discourse markers (*well*, *like*).

- **Restarts**. *I have to plan way in advance, because, + or, what I've done is found like doctors' and dentists' office with extended hours.*

- **Repairs**. *And, uh, I called you know from [that, + the] T I Data Base Calling Instructions.*

- **Repeats**. *Plus, I bet it [cuts, + cuts] down on your absenteeism.*

- **Word Fragments**. *Yeah, but I can [rem-, + remember] back growing up.*

It is possible that some categories may have no effect on topic classification performance, while others negatively impact performance. For example, fillers, such as filled pauses and discourse markers, are very frequent in a conversation, so their relevance (or lack thereof) may be robustly estimated using the text with disfluencies. However, filler removal may cause word-pair features to be more useful. Restarts and repairs represent a more interesting category since the intention of the speaker changes (or is repaired) and this may adversely affect performance. Repeats distort counts, but the majority are on very frequent words *I I am sure...*, so the argument for the filler category may apply. However, repeats are easiest to detect, so it would be good news if they dominate any performance differences due to being the most frequent category. In addition, although word fragments can be mapped to a single token when the true transcripts are used, they will be erroneously recognized as a full word or deleted when using an ASR system, possibly impacting the neighboring words

as well. Finally, there exists the possibility that disfluencies could help topic classification, through repetition of key words, though we conjecture that the disruptions outweigh the useful cases (as supported by the results of section 6.5).

6.2 Research Questions

There are four main questions that we answer in this chapter:

- **Does the removal of disfluencies lead to a better document representation for topic classification?** We include experiments using a variety of classifiers to verify that there is a consistent improvement of performance. In addition, we investigate the effect of disfluencies on the bag-of-words and bag-of-word-pairs representations, since the impact of disfluencies may depend on the choice of representation. Finally, we look at different classes of disfluencies, since some are easier to automatically detect than others.

- **How do disfluencies interact with feature selection?** Many of the words in a disfluent segment are high frequency and not closely-associated with any topic, such as *I mean*, *um* etc. It is possible that feature-selection methods remove most of the words from disfluent regions of the text. Alternatively, it may be that removing disfluencies before feature selection leads to better results.

- **Can feature selection be improved by first removing disfluencies?** In standard feature selection methods, all occurrences of a word are removed from the data. Therefore, if a word is irrelevant in one context and relevant in another, it will still be removed if the aggregate statistics deem it irrelevant. For example, a very common word within a disfluency is *mean*. It can be the case that if the word *mean* is found outside a disfluency it can be relevant — the speaker may be talking about *arithmetic mean* or how *mean* a person is.

- **Do disfluencies impact true transcripts differently than ASR-generated transcripts?** It is possible that disfluencies will have a different impact on topic

classification performance when using an ASR system. A word fragment will never be recognized as such using an ASR system. In addition, a disfluency — even without word fragments — may be more challenging to correctly recognize, because the language context is disrupted.

The outcome of these experiments can suggest new directions for further research. For example, if removing disfluencies is important for topic classification, then the results provide added motivation for automatic disfluency detection research.

6.3 Corpus & Task

For all our experiments we have used the Switchboard-I corpus [59]. A subset of the Switchboard-I corpus annotated for disfluencies is converted from the older TB3 data [115] to the more recent LDC V5.0 [157], maintaining the correction information. There are in total 1126 conversations or 2252 conversation sides annotated for disfluencies, consisting of about 1.45M term occurrences. The annotation defines three parts for each edit disfluency, the deletable portion, the interruption point and the correction. The deletable portion is the disfluent part of the utterance and the one that gets deleted, the interruption point marks the boundary between the deletable portion and the correction which can involve an editing term or no terms at all, and the correction is the fluent part of the utterance and the one that is retained. An example annotation is shown below, where the deletable portion (DEL) is within square brackets, the interruption point (IP) is marked with the plus sign and the correction (CORR) is within curly braces. The editing term of the correction is shown as EET (explicit editing term).

$$qualifications \underbrace{[that]}_{DEL} \underbrace{+}_{IP} \underbrace{you\ know}_{EET} \underbrace{\{that\}}_{CORR}\ you\ have$$

It should be noted that edit disfluencies can be overlapping or nested. The annotation methodology does not distinguish between repairs, restarts or repeats. To distinguish the three categories we applied the following simple rules, shown below, fillers were explicitly marked in the text. Note that here are multiple categories of fillers annotated, and that in this work only filled pauses and discourse markers consisted of the *"fillers"* category.

```
1: if CORR == ∅ then
2:    DISFLUENCY=RESTART;
3: else if DEL == CORR then
4:    DISFLUENCY=REPEAT;
5: else if DEL != CORR then
6:    DISFLUENCY=REPAIR;
7: end if
```

The task is to classify a conversation side to one of 67 possible topics. In all experiments, words with 2 or more occurrences in the entire corpus (train and test) have been retained. This resulted in vocabularies of 13866 and 13192 terms when using text before and after removal of disfluencies respectively. Instead of choosing a specific train and test set, we performed a 10-fold cross validation test and report the average and standard deviation of results. This allows us to observe the sensitivity of the results to different train/test data.

6.4 Methods

We have used two toolkits that are publicly available for research purposes and have implementations of six different text classifiers. The Bow toolkit [107] was used for training five out of six classifiers and the *SVMLight* toolkit (http://svmlight.joachims.org/) was used for training the Support Vector Machines classifier. Both toolkits are popular within the text classification community and have been extensively used in the past. The six classifiers we have used are:

- **Maximum Entropy** (MaxEnt) [122]
- **k Nearest Neighbors** (kNN) [106]
- **Support Vector Machines** (SVM) [80]
- **Naive Bayes with shrinkage** (NB) [111]
- **tfidf/Rocchio** (Rocchio) [78]

- **Probabilistic Indexing** (PrIndex) [54].

We will very briefly describe the last three, lesser-known text classifiers. Naive Bayes with shrinkage is the Naive Bayes classifier with an alternative way of smoothing. Instead of using Laplace smoothing, i.e. if $N(w, c)$ is the count of word w in topic c, we set $\tilde{N}(w, c) = N(w, c) + 1$, the topic-specific word distributions are smoothed with the word distribution in the whole training corpus, i.e. $\tilde{p}(w|c) = \lambda p(w|c) + (1 - \lambda)p(w)$. Therefore the probability of observing document $\vec{d}$ is given by:

$$p(\vec{d}) = \sum_{c=1}^{C} p(c) \prod_{k=1}^{N^d} (\lambda p(w_k = w|c) + (1 - \lambda)p(w_k = w))^{N_w^d} \tag{6.1}$$

where N_w^d is the number of occurrences of word w in document $\vec{d}$ and N^d is the number of unique words of document $\vec{d}$. The tfidf/Rocchio classifier represents each document m with a weight vector whose k-th element is given by:

$$d_k^m = \frac{f_k^m \log(N_D/n_k)}{\sum_{j=1}^{V} f_j^m \log(N_D/n_j)} \tag{6.2}$$

where N_D is the number of documents, n_k the number of documents in which the indexing term appears, and f_k^m is the frequency of term k in document m. The representation of class c is then constructed as:

$$\vec{u}_c = \frac{\alpha}{|R_c|} \sum_{m \in R_c} \vec{d}^m - \frac{\beta}{|\bar{R}_c|} \sum_{m \in \bar{R}_c} \vec{d}^m \tag{6.3}$$

where R_c is the set of training documents of class c and $\bar{R}_c$ is the set of training documents of every class but c. The parameters α and β are tuned either by hand or using cross validation. During testing, a new document $\vec{d}$ is assigned to the class with the maximum cosine similarity:

$$\hat{c} = \underset{c}{\operatorname{argmax}}\, cos(\vec{d}, \vec{u}_c) \tag{6.4}$$

The probabilistic indexing classifier is a statistical classifier where a new document d is classified to class $\hat{c}$ according to:

$$\hat{c} = \underset{c}{\operatorname{argmax}} \sum_{w} p(c|w)p(w|d) \tag{6.5}$$

Table 6.1: Topic classification accuracy of various classifiers using unigrams as features and reference transcripts.

	Keep All	**Remove All**	**Remove Fillers**	**Remove Restarts**	**Remove Repairs**	**Remove Repeats**	**Remove Fragments**
MaxEnt	78.0±0.9	79.0±0.9	78.4±0.8	78.0±0.8	78.0±0.9	78.3±0.9	78.2±0.8
kNN	83.9±0.6	84.7±0.8	84.0±0.7	84.5±0.5	84.6±1.2	84.4±0.7	84.2±0.5
SVM	83.0±0.4	83.4±0.6	83.4±0.6	83.0±0.5	83.1±0.9	82.6±0.7	83.8±0.6
PrIndex	82.8±2.3	85.6±1.8	84.2±1.1	84.4±1.6	83.5±2.1	84.7±2.0	83.9±1.7
NB	91.4±0.9	91.9±0.8	91.8±0.6	91.4±0.4	91.6±0.6	91.6±0.5	91.6±0.5
Rocchio	92.4±2.2	93.1±0.6	93.2±0.6	92.3±1.9	91.4±2.6	92.3±2.2	91.5±2.3

and $p(c|w)$ is evaluated through Bayes Rule.

For all our experiments the default settings for each classifier have been used. For example, the smoothing coefficient in NB was set to $\lambda = 0.6$ and for kNN $k = 30$. For the SVM experiments, we used the one-vs-one approach for multi-class recognition using SVMLight (see chapter 4). For the feature selection experiments, we have used the IG measure (see chapters 2 and 4).

6.5 Experiments

We have distinguished seven cases in our data. Using the original text with all the disfluencies which we denote on the tables with **Keep All**, removing all five categories of disfluencies (**Remove All**) and then individually removing each one of the five categories.

6.5.1 Effect of disfluencies on the BOW representation

We begin the experiments using the standard bag-of-words representation. In Table 6.1, we see the topic classification accuracy across different classifiers, and also by individually removing each disfluency category. The standard deviation of all classification experiments is also reported.

Table 6.2: Relative reduction of word counts from removing different disfluency categories.

Keep All	**Remove All**	**Remove Fillers**	**Remove Restarts**	**Remove Repairs**	**Remove Repeats**	**Remove Fragments**
-	11.9%	6.6%	0.8%	2.8%	2.6%	0.9%

From Table 6.1, we can see that overall there is a small but consistent difference by removing all disfluencies. Looking at the top 3 results, we find that the differences between **Keep All** and **Remove All** are significant for PrIndex and NB ($p < 10^{-3}$) and marginally significant for PrIndex ($p = 0.11$), but not significant for Rocchio, using a Student's t-test on 50 cross-validation subsets in each case. Removing individual disfluency categories provides classification accuracies within the range of two extremes (**Remove All** and **Keep All**). Since the difference between the two extremes is small, it is hard to say what is the relative influence of each one of the categories, but it is certainly the case that it is not a single category that accounts for all of the difference. In Table 6.2, the relative reduction of word occurrences compared to retaining all disfluencies is shown. In Table 6.2, the biggest category is fillers, which can explain why the impact of removing only fillers appears to be slightly bigger than other categories in Table 6.1. Note also that since disfluencies can be nested, the sum of words removed from each one of the categories can be higher than the words removed from all categories.

6.5.2 Effect of disfluencies on the BOWP representation

The next question we attempt to answer is whether more complex representations can benefit more from removing disfluencies. In chapter 4, we showed that bigrams can perform better than unigrams for Switchboard-like conversations, when enough training data are available [22]. A reason for this is that bigrams can capture expressions that are inadequately modeled with unigrams. For example, for the topic *"reality shows"* a relevant bigram is *"big brother"*. But neither *"big"* or *"brother"* as individual words can capture this. If disfluencies can disrupt the sequence of such relevant bigrams, for example *big uh, um, brother* then

Table 6.3: Topic classification accuracy of various classifiers using bigrams as features and reference transcripts.

	Keep All	**Remove All**	**Remove Fillers**	**Remove Restarts**	**Remove Repairs**	**Remove Repeats**	**Remove Fragments**
MaxEnt	63.4±1.1	65.4±1.3	64.1±0.8	63.5±1.2	63.6±0.7	63.0±0.7	62.1±0.8
kNN	72.8±1.1	79.2±0.7	78.0±1.0	72.8±0.8	74.0±1.0	74.1±0.9	73.1±1.0
SVM	49.6±1.4	50.8±1.5	50.4±1.8	49.8±0.9	50.2±1.1	50.3±0.9	50.0±1.2
PrIndex	94.3±0.4	94.5±0.7	94.4±0.5	94.2±0.6	94.5±0.4	94.0±0.5	94.2±0.5
NB	81.5±0.4	83.4±0.7	82.6±0.7	82.0±1.0	81.8±0.8	81.8±0.7	81.4±0.5
Rocchio	85.2±0.6	86.4±1.0	86.0±0.7	85.3±0.8	85.8±0.7	85.4±0.7	85.5±0.9

they can effectively weaken the representational capacity of bigrams. In Table 6.3 we report the results of using bigrams (bag-of-word-pairs or BOWP) as the representation method. Overall, we notice that the difference between the **Keep All** and **Remove All** cases is increased, compared to unigrams (except for PrIndex). This difference (between **Keep All** and **Remove All**) is significant for the Rocchio classifier ($p < 6 \times 10^{-3}$) and for NB-Shrinkage, but not significant for the best-performing classifier (probabilistic indexing). For all classifiers, except probabilistic indexing, it appears that using bigrams degrades the performance considerably compared to unigrams. Surprisingly, probabilistic indexing gets a significant boost, offering the best result over all classifiers and over all representations. Another interesting observation is that the kNN classifier benefits significantly by removing disfluencies when using bigrams as features.

6.5.3 Feature selection and disfluencies

The next two questions we explore are a) whether the negative contribution of disfluencies can be mitigated with feature selection (e.g. words frequently associated with disfluencies are removed in the feature selection process) and, alternatively, b) whether feature selection is more effective when disfluencies are removed. We performed feature selection, training

Table 6.4: The effect of feature selection on original text (reference transcripts) with and without disfluencies, using the top 5K unigrams selected with information gain.

	Keep All	**Remove All**
MaxEnt	-1.1	-1.1
kNN	-0.2	-0.7
SVM	-1.9	-1.8
PrIndex	+3.3	+1.0
NB	-0.2	-0.9
Rocchio	+0.4	+0.1

and testing of classifiers on disfluent text and on text with disfluencies removed. Table 6.4 shows the difference in average classification performance between these two experiments and the first two columns of Table 6.1. In Table 6.4, column 1, we can see that keeping only the top 5K IG words does not improve the results compared to using all the word features, except for the case of PrIndex which records a significant boost. This IG-based feature selection removed 60% of the disfluency word types and 74% of the disfluency tokens, so it appears that there are some topically important words in disfluency regions. To answer the second question, we can compare the columns of Table 6.4, which show that IG-based feature selection is not improved by using text with disfluencies removed.

6.5.4 ASR-generated transcripts and disfluencies

We have used the SRI Decipher ASR system [154] to decode all the 2252 conversation sides. We have used only the first step of the entire decoding process which consists of using bigram language models with unadapted MFCC acoustic models to perform the decoding. Since these data have been part of the training data of the SRI Decipher system, continuing for subsequent decoding steps would result in an unrealistically low word error rate (WER). The WER using only the first step is 30.2%. Since we have available the time segments of each disfluency and the SRI Decipher system outputs the time segments for each word, we

Table 6.5: Topic classification on the ASR transcripts using unigrams as features.

	Keep All	**Remove All**
MaxEnt	76.6±1.1	78.1±0.8
kNN	83.9±0.3	83.1±0.7
SVM	81.7±0.6	82.4±0.3
PrIndex	81.7±1.5	84.3±1.0
NB	89.9±0.5	89.9±0.7
Rocchio	85.1±1.4	91.5±0.9

can remove all words that fall mostly within a disfluency. Here, "mostly" refers to more than half of the word's duration being within a disfluency. This process is not perfect: words that should not have been removed will be removed, and words that should have been removed will not be removed, but it is fairly accurate. The experiments approximate the oracle case where disfluency removal is almost perfect.

The results are shown in Table 6.5 and show a mixed picture. The best classifier in other experiments, Rocchio, degrades substantially with ASR errors (comparing with Table 6.1) and clearly benefits from removing disfluencies. The second best classifier, NB, degrades somewhat with ASR but is not helped by disfluency removal. For most classifiers, there is little degradation due to ASR (despite a 30% WER) and a small benefit to disfluency removal, but it may be that any benefit is lost (or worse) with automatic disfluency detection. Hence, proper choice of classifier is more important than disfluency removal with ASR transcripts.

6.6 Discussion

In this chapter, we have explored the impact of disfluencies on topic classification of natural human-human conversations. Overall, we can say that removing disfluencies has a small impact on the bag-of-words representation but appears to have a bigger impact on the bag-of-word-pairs representation for several classifiers. It is worth mentioning that the best topic

classification results are obtained using the PrIndex classifier with bigrams and disfluencies removed. Another observation from our experiments was that feature selection can not effectively remove disfluencies. In addition, feature selection does not appear to be greatly improved by first removing disfluencies. Lastly, we have explored the effect of disfluencies on ASR-generated transcripts. On ASR transcripts, we have found that the effect of both word errors and disfluency removal is highly dependent on the classification method, with the greatest benefit from disfluency removal coming for the classifier most sensitive to errors. For both true and ASR transcripts the best performance is achieved by removing disfluencies, but the relative gain is not large and may be lost with automatic disfluency detection. Overall, we find that choice of classifier has a much bigger effect than disfluency removal. With current classifiers and the bag-of-word representation, there appears to be little need for disfluency removal, though this could change if future developments make more use of word sequence patterns.

The conclusions of this study need to be interpreted with the caveat that the corpus was explicitly designed to include dialogs on a single topic. In multi-topic and.or multi-party speech, disfluencies may play a more important role, and similarly for more fine-grained topic labeling. In addition, there are other tasks in language processing where disfluencies might be informative (rather than interpreted as noise), such as speaker and topic segmentation.

Chapter 7

USING SYMBOLIC PROMINENCE IN FEATURE SELECTION FOR TOPIC CLASSIFICATION AND CLUSTERING

In this chapter, we use the output of a symbolic prominence classifier rather than acoustic cues of prominence, to improve the tasks of clustering and classification of spontaneous conversations to topics. In our experiments, we combine the output of a prominence classifier with lexical feature selection and combination methods to build improved feature subsets. Evaluated for the task of topic classification on a subset of Switchboard-I, the combination method offered a 11% relative reduction of classification error compared to using lexical-only feature selection methods; similar gains are reported for clustering.

7.1 Prosody in Spoken Language Processing

Various aspects of prosody have been successfully integrated in a number of spoken language processing tasks such as dialog act classification in conversational speech [144] and dialogue systems [125], discourse segmentation [145], error detection in dialogue systems [3] and voicemail summarization [92]. A less explored avenue to improve such tasks is prominence. Prominence, also referred to as pitch accent in English, is phrase-level emphasis given to one or more syllables of a word that goes beyond word-level strong/weak syllable differences associated with lexical stress. Prominence can be in combination but is not the same as intonation marking phrase or sentence boundaries. Some of the acoustic correlates of prominence are particularly high (or low) F0 targets, duration lengthening and increased energy of a syllable. Prominence has long been linked with information structure in numerous ways such as to contrast new vs. old information [20, 72] and to give local focus. Despite the wealth of literature on the role of prominence for human understanding, few attempts have been made to integrate prominence in spoken language processing tasks. In

[125], stress is used to disambiguate between words. In [30], a spoken document retrieval system is augmented with acoustic features such as duration and energy but no F0 information was used, which is an important feature for detecting prominence. Links between prominence and simple measures of word saliency have been established in [129, 147]. In [92], a number of acoustic and lexical features have been used to learn which words to extract for voicemail summarization. Acoustic cues that correlate with prominence have been found to be important. Most related with the current work is [18], where prosody is used to discriminate content from function words, but the approach was not integrated in a topic classification or detection system.

In this chapter, we use the output of a prominence detection system to facilitate classification and clustering of topics in human-human conversations. We show that prominence can be integrated with standard techniques to design better feature subsets. Incorporating prominence into the semantic characterization of speech has the major advantage that it can be equally useful in supervised and unsupervised cases. Traditional lexical measures of word saliency rely on annotated data in terms of topics. These measures become less reliable as the number of annotated examples decreases and do not apply at all in unsupervised cases. In contrast, prominence may be useful in cases of lack of supervision, e.g. in the absence of topic labels. Moreover, it is likely to be less sensitive to ASR errors, so it may be more useful when hand-transcripts are not available, although in this chapter we have experimented only with the true transcripts. Importantly, training an automatic prominence detection system with accuracy around 80% does not require a large amount of hand-annotated data. The prominence classifier used in this chapter was trained on 124 Switchboard-I conversations, where each word was hand-annotated with a binary value (prominent or not).

7.2 Leveraging Prominence for Topic Detection

The problem of feature selection and combination is at the core of text and spoken language classification. Typical vocabulary sizes are on the order of tens of thousands and only some of these features are deemed relevant for classification. Traditional techniques for determining which words should be removed rely on lexical information only. For example, one of the

best performing feature selection measures is IG [49], described in chapter 4. All words in the vocabulary are ranked according to IG and the top N words are selected. Prominence can be used to complement measures such as IG. Here, two alternatives are investigated.

First, the prominence classifier can be used to produce a score for every word occurrence in the dataset. The score will be the probability of each occurrence being prominent. Words can then be ranked according to their average prominence, i.e. the average value of the prominence scores of all occurrences of a word. Having two alternative ranked lists - one from IG and another from prominence - the objective is to merge them so as to maximize topic classification/clustering performance. One way of combining the lists is to cascade them. First, the N_p words with the lowest average prominence are eliminated, the remaining words are ranked according to IG and the top N_l words selected. This scheme will be most successful when the two lists produce their best results in different regions. As shown in section 7.3 their effects are complementary; prominence can robustly identify irrelevant words but not the most relevant, while IG can identify quite well the most relevant words but not the most irrelevant words.

A second way of leveraging prominence is to use the prominence scores of each occurrence rather than their average. An appealing characteristic of prominence scoring is that different occurrences of a word will have different prominence scores, whereas the common bag-of-words representation treats all occurrences of a word uniformly. Conceptually, this scheme is a generalization of the bag-of-words representation in that a word counter is conditionally incremented based on the prominence value. One combination method is to only count the word occurrences above a certain threshold and then calculate IG.

7.3 Experiments

Experiments were performed on 648 conversations or 1296 conversation sides from the Switchboard-I corpus [59]. A single topic from a list of 64 is associated with each conversation. The true transcripts of the conversations were used for all experiments. Each word of the transcript was automatically annotated with a score between 0 and 1, indicating the posterior probability that the word is prominent, given local acoustic and text cues. The

total number of word occurrences was approximately 800K, and keeping words with 5 or more occurrences, resulted in a vocabulary of 5211.

The prominence classifier described in [172] was used in all our experiments. All words of 124 conversation sides were annotated with binary values (prominent or not) and C4.5 decision trees were used for training. The prominence model predicts the posterior probability of prominence for each word using a decision tree with a combination of prosodic and text features. The prosodic features include: various F0 statistics within and across words (normalizing for speaker mean and variance), normalized duration statistics, silence duration, and energy statistics over a word. The text features included part-of-speech labels of the target word and its neighbors. The model was trained using a weakly supervised approach, where an initial model is trained on a small set of labeled data (roughly 4 hrs from 124 conversations), and then the EM algorithm is used to incorporate additional data that does not have prosodic mark-up but does have syntactic parses that can be used as additional features. For reference, the error rate of the classifier when used in predicting prominence was 21.3%, though hard decisions were not used in this chapter.

7.3.1 Classification experiments

The Bow toolkit [107] was used for all the classification experiments. The 1296 conversation sides were equally split in train and test datasets and a 10-fold cross validation methodology is used. The standard deviation of all classification experiments are also reported using error bars in figures.

In the first experiment, we remove words from the vocabulary according to their average prominence, comparing three criteria: removing words with the highest average prominence, lowest average prominence and removing words at random. In Figure 7.1, the three different sets of results are shown. In all cases the Naive Bayes with Laplacian prior is used as the learning method. Removing words with average prominence probability 0.45 or lower resulted in a very significant gain in performance over using all features; the classification accuracy rose from .548 when using all 5211 words to .712 when using words with average prominence 0.45 or higher (4337 words). Removing words at random consistently degraded

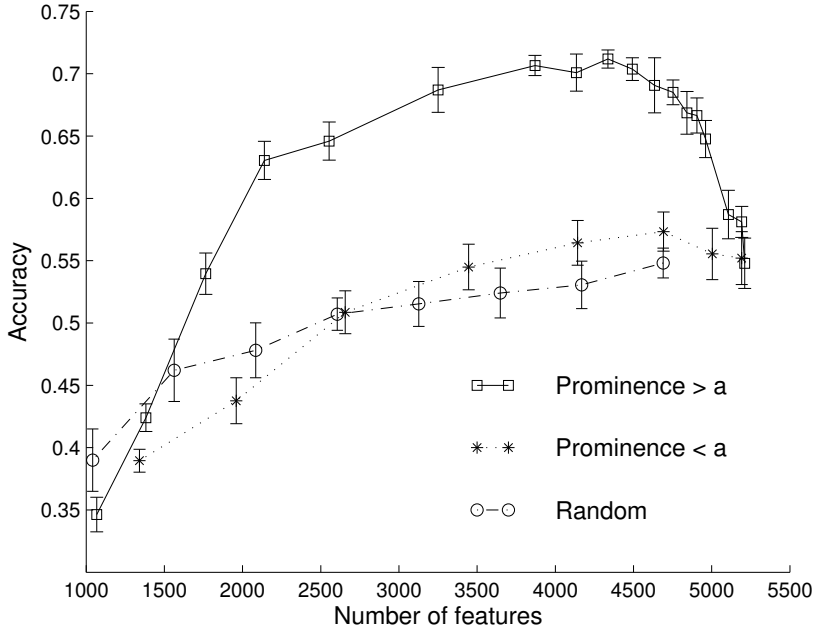

Figure 7.1: Eliminating words according to their average prominence. 10-fold cross validation is used and Naive Bayes with Laplace prior as the learning method.

the performance, while removing words with the highest prominence was not much different from random. These results suggest that prominence can quite robustly identify the least important words but not the most important words. If the most important words could be identified with prominence then removing words with the maximum average prominence should have resulted in worse results than removing words at random.

In the second experiment, we combine the IG measure with prominence. We compare two methods to select words. The first method is to rank all 5211 words according to IG and select the N highest and the second method is to remove words with average prominence 0.45 or higher, rank the remaining 4337 words according to IG and select the N highest. The results are shown in Figure 7.2 where it can be seen that using only IG improves classification accuracy substantially compared to using all 5211 features, but combining prominence and IG offers additive gains, for every number of final features explored.

It is also instructive to see some of the words that are removed using prominence. In

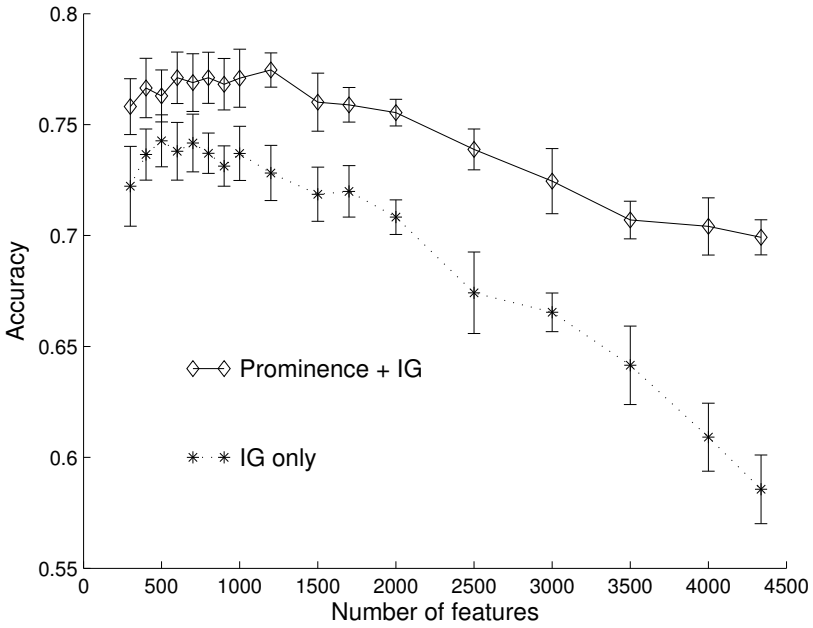

Figure 7.2: Feature subsets determined by IG only and prominence combined with IG. 10-fold cross validation is used and Naive Bayes with Laplace prior as the learning method.

Table 7.1 we see the 16 words with the least average prominence and their corresponding ranking in the IG list. We observe that for many common words, IG fails to identify them as irrelevant, as intuition would suggest. Using a default stopword list would capture some of these words, but not all. For example, the word *bye-bye* is specific to the task at hand and would not be included in a default stopword list.

In Figure 7.3 we compare against another baseline, using a list of human-created stopwords and then ranking the remaining features with the IG measure. The stopword list consists of 484 words and is the same as the one used in the CLUTO toolkit. We can observe that when using 2000 features or less the prominence+IG scheme is better than the stopwords+IG scheme. When using more than 2000 features the two schemes converge.

The results in Figures 7.1, 7.2 and 7.3 were obtained with the Naive Bayes model. To make sure that prominence can help build better feature spaces, independent of the choice of classifier, we repeated the experiments of Figure 7.2 for a number of successful learning

Table 7.1: The 16 words with the least average prominence and their position in the IG list (out of 5211 words, smaller numbers are less important).

1-8		**9-16**	
thinner	349	shall	946
to	116	of	56
bye-bye	4368	from	4642
than	3413	at	4214
an	4686	with	4232
till	3477	within	4254
a	4	and	900
the	300	should've	1735

methods for text classification. The results are given in Table 7.2, which shows the relative reduction in classification error for the lowest achievable classification errors using IG only and prominence+IG for a variety of learning methods.

Table 7.2: Relative reduction of classification error using prominence+IG compared to IG only for various learning methods.

NB	**Rocchio**	**Prind**	**SVM**
11%	12%	15%	7%

The four different classifiers used are Naive Bayes with Laplace smoothing (NB), Rocchio (Rocchio) [78], Probabilistic indexing (Prind) [54] and Support Vector Machines (SVMs) [80]. The results of Table 7.2 show consistent gains in performance using the combined feature selection method, reinforcing our hypothesis that prominence helps design better feature spaces irrespective of the classifier. Even for SVMs, that are known to provide state-of-the-art results in text classification and are considered robust to irrelevant features, the proposed method still offers gains.

Up to this point we have only experimented with the average prominence of a word. In

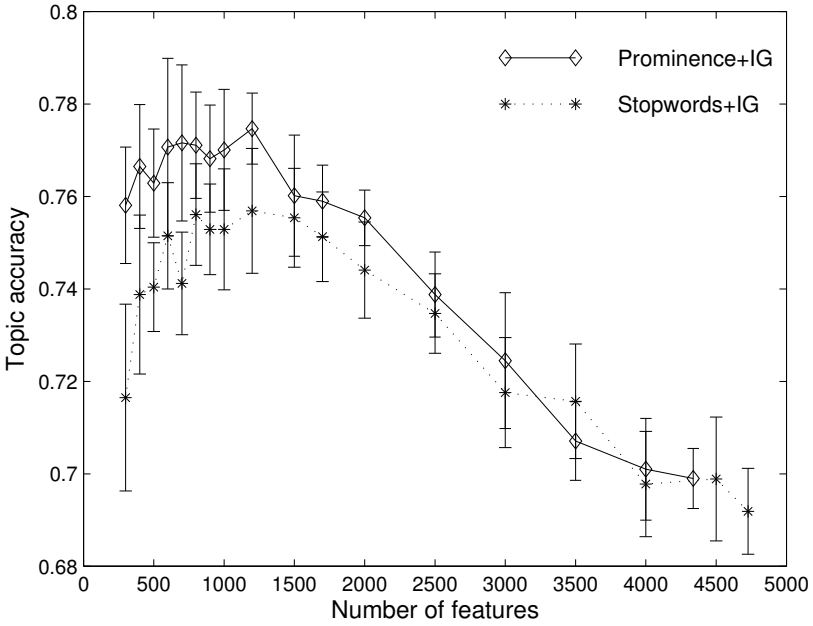

Figure 7.3: Feature subsets determined by stopwords+IG and prominence combined with IG. 10-fold cross validation is used and Naive Bayes with Laplace prior as the learning method.

Figure 7.4 we select word occurrences rather than words, according to their prominence. The results are shown in Figure 7.4 and show that this is not as good of a selection mechanism as with average prominence. This can be due to two main reasons. First, using the average prominence reduces the variability of the prominence prediction; in other words, using the average reduces the "noise" in the data. Second, if we remove a high percentage of an irrelevant word, but not every occurrence then the remaining few occurrences may be biased towards a specific topic, therefore the classifier will mistakenly train this word as relevant.

7.3.2 Clustering experiments

An important characteristic of utilizing symbolic prominence is that it does not require any supervised data in terms of topics, while all lexical word saliency measures do. IG is not applicable in an unsupervised scenario. Designing feature subsets for text clustering

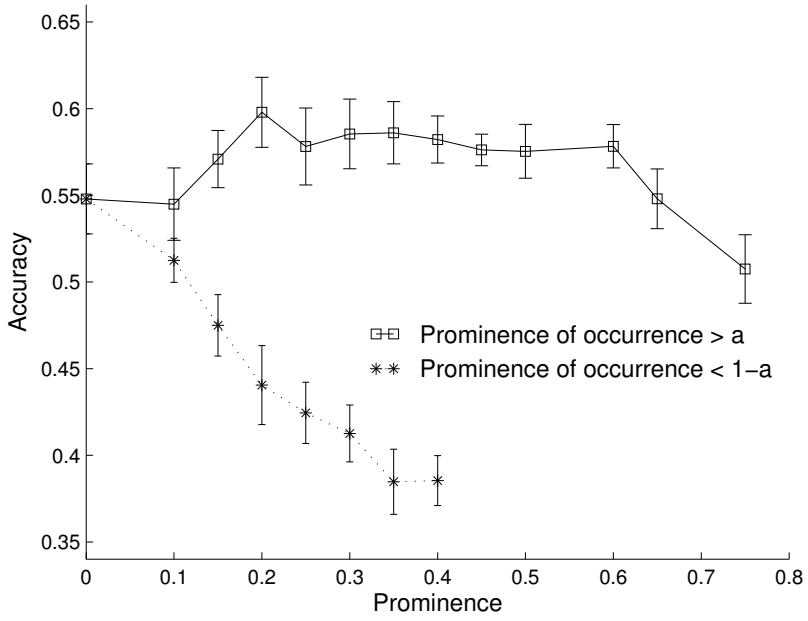

Figure 7.4: Eliminating word occurrences according to their prominence. 10-fold cross validation is used and Naive Bayes with Laplace prior as the learning method.

is usually done with feature correlation methods, such as Latent Semantic Analysis (LSA) [35]. In this respect, using prominence to select irrelevant features is complimentary to feature combination/correlation approaches such as LSA. A natural way to combine LSA and prominence is to first remove words according to prominence and then combine the remaining ones with LSA. We used the CLUTO toolkit[1][182], a software package for clustering in high-dimensional spaces, to cluster the 1296 conversations with the number of topics (64) assumed to be known a priori. We used the default values of CLUTO to perform clustering. The objective function to maximize is intra-cluster cosine similarity, ten random restarts are performed and the one with the highest objective function is retained. Since the final result depends on the initial conditions we performed 10 trials and also report standard deviations. Note that the standard deviations here do not have the same interpretation as in classification, since here the same dataset is used for clustering. The results were evaluated

[1]http://www-users.cs.umn.edu/~karypis/cluto/

using the adjusted Rand index [73] (see chapter 2) a common measure to evaluate clustering solutions. The adjusted Rand index is the fraction of pairs of points that were correctly clustered together and correctly clustered in different classes, adjusted to zero for chance results. Figure 7.5 shows the clustering results, where we see that for a variety of features combining prominence with LSA is better than using LSA only. Words with average prominence 0.4 or higher were selected. LSA is performed on the tf·idf conversation-side word matrix.

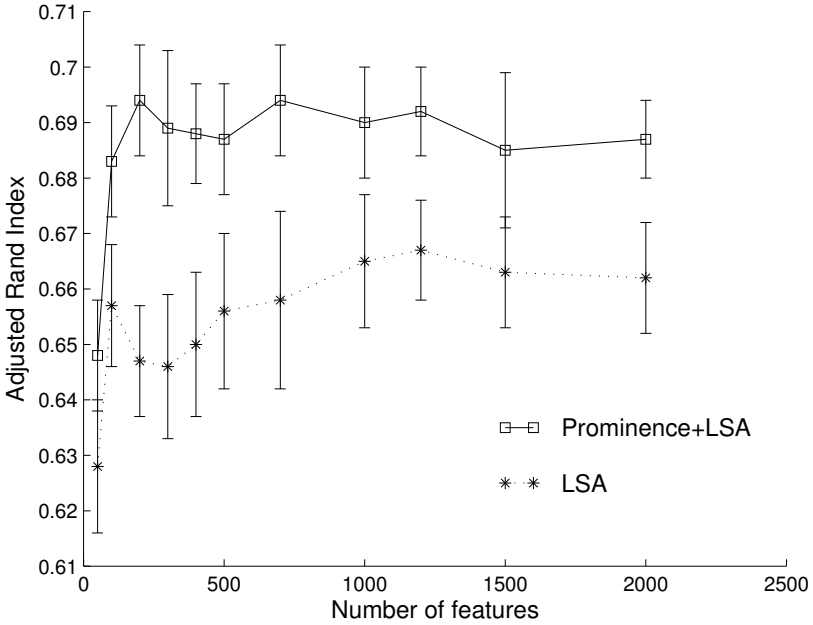

Figure 7.5: Effect of LSA only and prominence combined with LSA on clustering performance.

Using all 5211 features resulted in an adjusted Rand index of .667 with standard deviation of .016. Using LSA alone did not offer gains over the baseline, while using the combined scheme resulted in an adjusted Rand index of .694.

7.4 Discussion

We have demonstrated how automatically detected symbolic prominence can be used to design better feature subsets for semantic characterization of natural human-human conversations. Despite the fact that the prominence classifier had an error rate of about 21.3% it was shown to be useful for the tasks of topic classification and clustering. Specifically, words with very low and high average prominence were shown to be mostly irrelevant for classification purposes. This suggests that prominence may have a bigger role in SLU systems by filtering out uninteresting areas rather than detecting areas of high content. Further, prominence was shown to lift the gains from other common feature selection and combination methods, such as IG and LSA. In the future, it will be interesting to examine the words with high prominence and low IG, since this may be a way to detect discourse markers.

In addition, prominence can potentially be useful when using the ASR transcriptions as well. Using an ASR system adds "noise" to the word sequence, and therefore lexical measures of word saliency may be affected. Prominence detection accuracy may also be affected, but because the feature selection algorithm used here leverages average statistics and the prominence detection algorithm relies on prosody as well as word identity, it is likely that the increased prominence detection error will have minimal impact on feature selection. Moreover, the prominence detection algorithm used here can easily be improved upon since it does not take advantage of sequential dependencies.

Chapter 8

CONFUSABILITY-DRIVEN WORD CLUSTERS FOR TOPIC CLASSIFICATION

In this chapter, we propose a method to cluster the words in the vocabulary according to their confusability, as determined by an automatic speech recognizer. By clustering highly confusable words we aim at reducing the vocabulary to tokens that are robustly identified. The method is completely unsupervised in the sense that it does not need to have access to the correct transcripts or topic labels. For the task of topic classification, the derived clusters reduced the topic classification error by 10% compared to using the 1-best hypotheses of the ASR system. Similar gains can be obtained by applying Porter's stemming algorithm on the 1-best transcripts or utilizing multiple ASR hypotheses through confusion networks. In addition, it is shown that methods for coping with errors introduced by the ASR system are more important when there are small amounts of topically labeled data, a situation that is likely to arise in practical applications.

8.1 Introduction

An important issue in topic learning in speech is that the ASR system that is employed to convert speech to text, introduces a number of errors. Typical state-of-the-art conversational speech ASR systems exhibit word error rates around 15% [127], although they can be higher if the acoustic conditions degrade, e.g. multi-speaker overlap or highly disfluent speech [155]. The impact of word errors in topic classification performance of human-human conversations is not well understood, although tasks that share many issues such as call-routing and spoken document retrieval exhibit a relative drop in performance that is smaller than the percentage of word errors introduced. The purpose of this chapter is to assess the degree to which ASR errors impact topic classification performance in conversational speech and

provide a method to increase topic classification accuracy given the ASR transcripts. For a review of past methods on handling ASR errors for spoken language processing and also for approaches other than using a large vocabulary ASR system, e.g. using a phone recognizer for spoken document retrieval, the reader is referred to chapter 2, section 2.5.2.

8.2 ASR errors and Topic Classification

First, we need to determine the impact of ASR errors in topic classification accuracy of human-human conversations. We have used the SRI Decipher system [154] to decode 4484 conversation sides of Switchboard-I corpus [59]. Only the first step of the entire decoding process was used, which consists of using bigram language models with unadapted MFCC acoustic models to perform the decoding. Since these data have been part of the training data of the SRI Decipher system, continuing for subsequent decoding steps would result in an unrealistically low word error rate (WER). The WER using only the first step is 30.2%. To obtain a second reference point, the 4484 conversation sides were also decoded using a tighter beam width resulting in a WER of 40.1%.

In Figure 8.1, topic classification accuracies for the true transcripts and 1-best ASR hypothesized transcripts are shown where training is matched to the testing condition. Naive Bayes with shrinkage is used as the learning method, since it was shown to be one of the best performing methods. 10-fold cross validation is applied, and mean and standard deviation are reported for each experiment. Different amounts of training data are used to investigate the impact of training set size on the results. Figure 8.1 shows that using ASR transcripts leads to an increase in topic classification error that depends on the amount of training data. For a 10/90 train/test split the increase is 16% relative to the true transcripts. For a 90/10 split the relative increase is 2%, a non-significant difference according to Student's t-test. When the WER rises to 40.1% the topic classification error increases significantly more. For a 10/90 split, a 44% relative increase is recorded, while a 20% for a 90/10 split. These results show that the ASR errors are important for the topic classification task, and that the impact is not linear as a function of WER. Increasing the WER from 30.2% to 40.1% increased the relative difference of classification error compared to the true transcripts, from

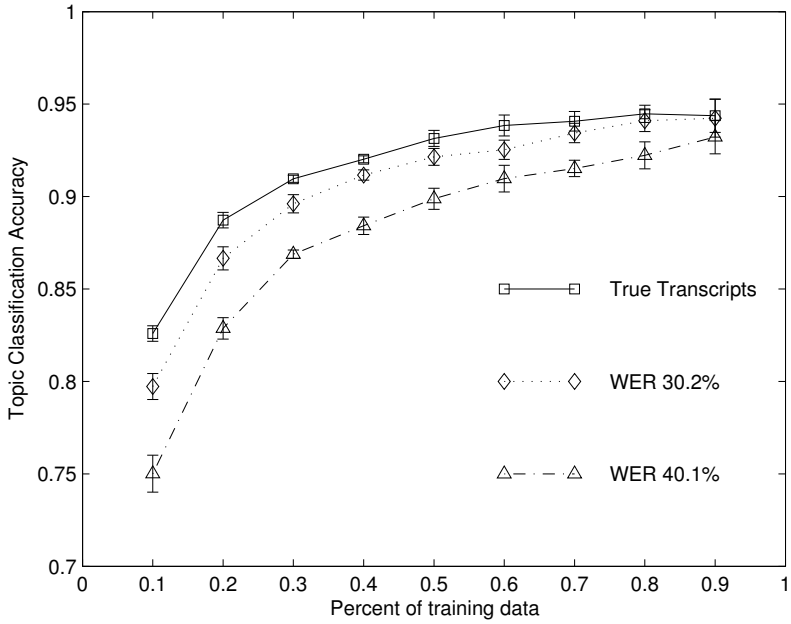

Figure 8.1: Effect of ASR errors on topic classification accuracy on the Switchboard-I corpus.

16% to 44% (for a 10/90 train/test split). In addition, the impact of word errors is lessened when there is more training data.

In Table 8.1 we see the effects of matched and mismatched conditions during training and testing. We can observe that using the true transcripts in either training or testing results in higher topic accuracy compared to not using true transcripts. For example using 1-best for both training and testing resulted in mean accuracy of 79.7 while using true transcripts for training and 1-best for testing resulted in mean accuracy of 80.5. Similar results are obtained when we use 1-best transcripts for training and true transcripts for testing. This shows that a way to improve topic performance is to have more human-annotated transcripts in both training and/or testing, rather than having matched training/testing conditions.

Table 8.1: Effect of matched and mismatched conditions in training and testing. **True** refers to using true transcripts while **1-best** refers to using the 1-best ASR hypotheses with 30.2% WER. Naive Bayes with shrinkage is used as the topic learning method and a 10/90 train/test split is used. Numbers shown are topic classification accuracies with standard deviations using 10-fold cross validation.

		TEST	
		True	**1-best**
TRAIN	**True**	82.6±.4	80.5±.7
	1-best	81.8±.3	79.7±.7

8.3 Clustering ASR-confusable words

Since the results of Figure 8.1 showed that the impact of word errors is more important under sparsity of topically annotated data, the focus will be in the 10/90 train/test split. Under small amounts of data, a major problem for any classification task is the variance of the estimates of the class-conditioned feature probabilities. Using maximum a posteriori estimation can be helpful, but the variance of the estimators remains a major issue, and can help explain why more training data can significantly help classification performance, as shown in Figure 8.1. One common approach to reducing the variance of the estimation is to group similar features together, therefore the estimator for the group will utilize the counts of all the features that belong in it. This basic observation is utilized in a wide array of learning problems and motivates the method proposed in this chapter. A key decision about the feature grouping process is to define a distortion metric. Given that the goal is to address problem of word errors, one method is to group together words that are highly confusable with each other, according to the ASR system. A supervised approach to this method is to align human-annotated transcripts with ASR 1-best hypotheses, and calculate the co-occurrence of words in the correct and hypothesized transcripts. The main problem with such an approach is that it requires the availability of human-annotated transcripts, making its use for new tasks especially hard.

In this chapter, we use an unsupervised approach to clustering words according to their

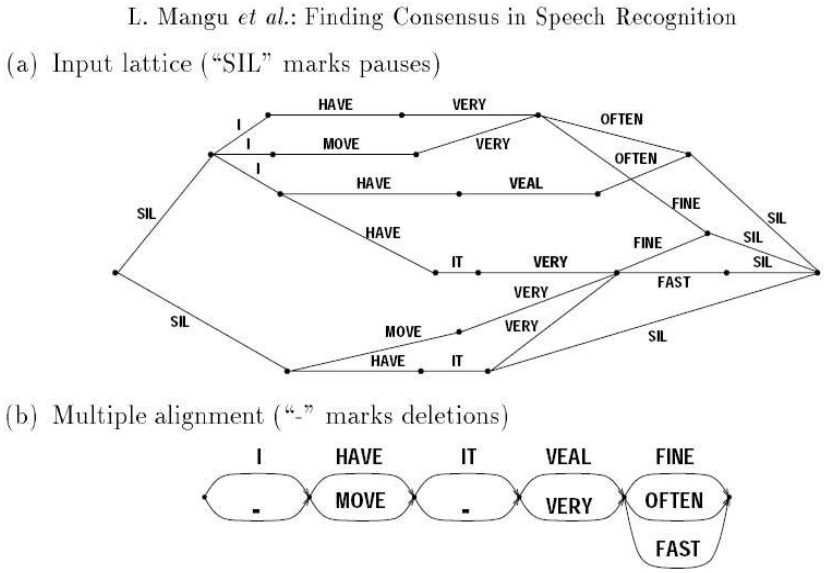

Figure 8.2: ASR Lattice and corresponding confusion network, reprinted with permission from [105].

confusability, in the sense that it does not require knowledge of the correct transcripts or topic labels. We use the confusion networks [105], to define the confusability of words. Confusion networks are a way to align the multiple hypotheses of an ASR system such that at position k all competing words are output. An example confusion network is shown in Figure 8.2. Each word w in position k is also associated with a probability $p(W_k = w|X)$ where X is the acoustic signal. This quantity combines the acoustic and language scores in a single number. Therefore, if two words are acoustically similar but linguistically distinct, their confusability will not be very high. Confusion networks have been used before for topic classification, as word confidence estimators in call routing [164]. They have also been used for modeling multiple hypotheses for sentence boundary detection [70] and building word discriminative models [166]. For topic classification, confusion networks allow us to decouple the co-occurrence of words due to two distinct factors, the ASR confusability and the topic similarity, while the 1-best output would mix those factors together. With

confusion networks we can represent a word w with a vector $\vec{R}_w$ of dimension L where the k-th dimension is $p(W_k = w|X)$. L is the number of word occurrences in the entire corpus and can be in the order of millions. Hopefully, we do not need to store all the dimensions since the vast majority will be zero. With this representation we use the cosine distance as a way to measure similarity between words and apply average linkage agglomerative clustering to group words. The procedure is shown in algorithm (2). Despite V being in the order of

Algorithm 2 Grouping ASR confusable words

Each word w is represented with the L-dimensional vector $\vec{R}_w$ where the k-th dimension is $p(W_k = w|X)$ and L is the total number of word occurrences

Define a $V \times V$ symmetric matrix S, V being the vocabulary size, with the (w, v) entry:

$$S_{w,v} = \cos(\vec{R}_w, \vec{R}_v) = \frac{\vec{R}_w \cdot \vec{R}_v^t}{\|\vec{R}_w\|\|\vec{R}_v\|} \tag{8.1}$$

where $\|\vec{R}_w\| = \sqrt{\vec{R}_w \cdot \vec{R}_w^t}$

Set number_of_clusters = V

Place each word in a separate cluster: $C = \{w_1, \ldots, w_V\}$

while number_of_clusters $> T$ **do**

Find clusters c_1 and c_2 such that:

$$(\hat{c}_1, \hat{c}_2) = \underset{c_1,c_2 \in C}{\operatorname{argmax}} S_{c_1,c_2} \tag{8.2}$$

Delete $\hat{c}_2$ from the set of available clusters: $C \leftarrow C - \{\hat{c}_2\}$

Update similarities of c_1 with any remaining cluster $c \in C$:

$$S_{\hat{c}_1,c} \leftarrow \frac{\eta_1 S_{\hat{c}_1,c} + \eta_2 S_{\hat{c}_2,c}}{\eta_1 + \eta_2} \tag{8.3}$$

where η_i is the number of words clustered in group i

Set number_of_clusters = number_of_clusters - 1

end while

tens of thousands, performing agglomerative clustering remains manageable, both in terms of memory and computational requirements, since the similarity matrix is typically very

sparse. Large amounts of speech can be decoded, and confusability can be derived from the ASR hypotheses. The only requirement is that the ASR confidence estimator must operate in a certain range. An ASR system that is always very confident about its output, i.e. $p(W_k = w|X) \approx 1 \ \forall k$ will not be very useful nor does a system that outputs a very high number of potential words at each step, i.e. $p(W_k = w|X) \approx \epsilon \ \forall k$. For most ASR systems, where there is a reasonable match between training and testing conditions, the above should not be a big issue.

In Figure 8.3, the effect of using different numbers of confusability-driven word clusters is shown. A 10/90 train/test split is used since it is under small amounts of training data,

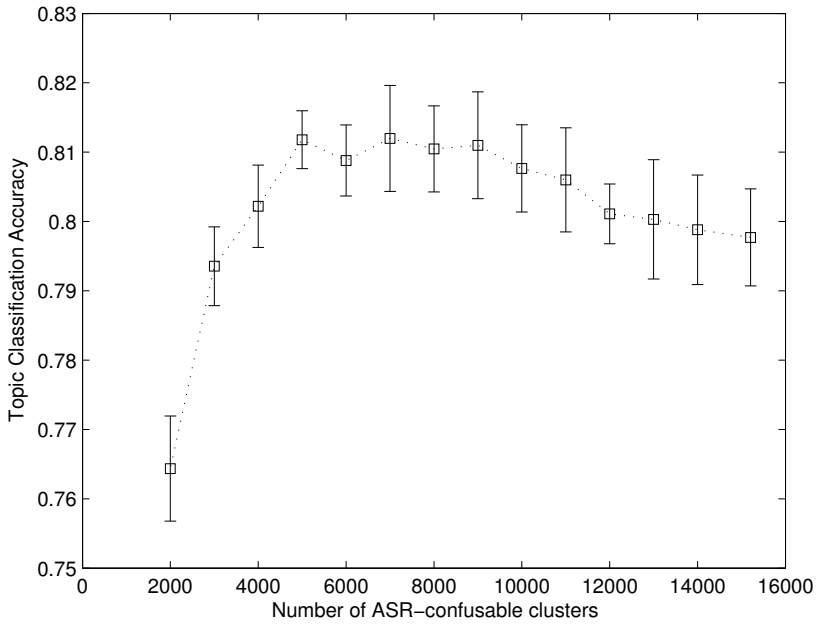

Figure 8.3: Topic classification accuracy for different numbers of confusability-driven word clusters.

where classification performance degrades the most. Using 15208 clusters is the same as using words for classification. From Figure 8.3 it can be observed that using up to 7000 confusability-driven clusters improves topic classification over using words. Further pursuing clustering starts to degrade the performance and after 4000 clusters, performance degrades

sharply. Using 7000 clusters resulted in a 81.2% topic classification performance vs. 79.7% for using words only. This difference is statistically significant ($p < 10^{-3}$) using Student's t-test and 50 cross-validation subsets.

It is instructive to see what kind of words are clustered together using the confusability-driven method. The left column of Table 8.2 shows the 20 words with the highest information gain, while the right column shows the 20 (out of 7000) word clusters with the highest information gain. We can observe that it is often the case that word clusters contain inflectional variants of a lemma but there are a number confusable words that are topically different, e.g. *praying* and *spraying* or *flour* and *flower*. Table 8.2 offers an explanation for the performance improvement that the scheme offers. Morphological variants will tend to have high confusability and since they are associated with the same topics, clustering them will be helpful when there are few training data.

In Table 8.3 the performance of the ASR-confusable clusters with a number of other baselines are also reported. The **Conf Nets** column refers to using the raw scores as output from the confusion networks. Therefore, instead of having integral word counts as in 1-best or true transcripts, fractional word counts are used. Confusion networks have been used to improve call routing performance in [164].

From Table 8.3 we observe that although the proposed method can offer gains compared to using the 1-best transcripts, using confusion networks or applying Porter's stemmer [132] on the 1-best transcripts produces comparable results. Overall, about 50% of the topic classification difference between the true and 1-best transcripts is recovered. The fact that both **Conf Nets** and **7K ASR clusters** produced similar results shows that the topic classification performance is not very sensitive to the values assigned to word counts.

Even though the ASR clusters gave only a small improvement (or no gain) in performance relative to other approaches, there are still a number of advantages of the proposed scheme. Compared to confusion networks, there are computational advantages for using the word ASR confusability-driven clusters. For confusion networks, the number of unique words in each conversation side is 2.2 times relative to 1-best, even if word occurrences with score 0.01 or less are skipped. Coupled with the fact that generating the confusion networks incurs added computations compared to 1-best, processing each side (during testing) can take at

Table 8.2: Top 20 words ranked according to Information Gain when using only words and using 7000 confusability-driven word clusters.

Words	**Using 7000 ASR-confusable clusters**
movie	*boil boiled boils oil*
movies	*dumbest government government's governments*
trees	*campinas camping*
american	*movie movies moving*
oil	*america american americans america's*
vote	*yard yards yarn*
government	*exercise exercised exercises exercising*
camping	*hobbies hobby hopping*
exercise	*water watered water's waters*
drug	*akeem keen team teen team's teams teens tina*
water	*boating voting devoting*
countries	*drug drugs*
favorite	*car cart card cards carts car's cars kara's*
rain	*flour flower flowers*
car	*service services surfaces*
taxes	*music music's*
yard	*recycle recycled recycles*
jury	*praying spraying spring sprained spray sprayed springs sprays*
music	*retreat retrieve treason tree's trees treat tree treats*
team	*credit credits*

Table 8.3: Comparing the ASR-confusable clusters with a number of other baseline methods. A 10-fold cross validation 10/90 train/test split of 4484 conversation sides is used and Naive Bayes with shrinkage is the classification method. The 1-best transcripts have a 30.2% WER.

True transcripts			**1-best transcripts**		
stemmed	**words**	**Conf Nets**	**7K ASR clusters**	**stemmed**	**words**
83.4±.5	82.6±.4	81.3±.6	81.2±.7	80.9±.6	79.7±.7

least a factor of 3 than with 1-best transcripts. This can be an issue depending on the classifier used and the requirements on total processing time. On the other hand, the ASR-confusable clusters can be generated offline and then applied on the 1-best transcripts. Since the number of clusters used will be about 2.5 times less than the original vocabulary, further computational savings can be realized. Comparing with the stemmed 1-best transcripts, the proposed approach can have an advantage when applied in a new and resource impoverished language, where no morphological analyzer exists. Although, we have not validated that the proposed approach can actually offer gains in other languages, it remains a strong possibility. Further, it may be that confusable words are more powerful than stemming in other languages where the inflections may be more acoustically distinct.

Finally, we have compared our approach with standard feature selection. The motivation is that if two topically distinct words are highly confusable then using feature selection can filter them out. Applying the KL feature selection on a 10/90 train/test split resulted in a 80.3% classification accuracy when using 14000 words and the accuracy quickly degraded after that. The topic classification accuracy with using all words is 79.7%. A major shortcoming of feature selection techniques is that they rely on data annotated with topic information therefore they are less likely to be helpful under small training set sizes.

8.4 Summary

A method that clusters ASR confusable words without requiring hand transcribed data is presented. The method is based on confusion networks and clusters words that highly co-occur in the same positions. The proposed method was shown to offer gains compared

to using the 1-best ASR hypotheses. An analysis of the words that are clustered together revealed that morphologically similar words are among the most confusable. This result can help explain why a WER of 30.2% causes disproportionally lower degradation of performance. If the erroneously decoded words are still about the same topics then the cost for topic classification is zero. In the future, it will be interesting to see if there are similar findings when the algorithm is applied to other languages. The proposed method is most useful when there are small amounts of topically annotated data. One of the findings of this chapter was that the effect of word errors is diminished as the amount of topically annotated data increases.

Chapter 9

SUMMARY AND FUTURE DIRECTIONS

This chapter summarizes the main conclusions of the dissertation, comments on the impact of this work outside the areas it was applied, and suggests directions for future research.

9.1 Main Conclusions and Impact

Combining multiple clustering partitions can improve clustering performance and offer a partition that is not as sensitive to the initial parameter values. Combining clusterers is a problem that is analogous to classifier combination, but with the added difficulty that the correspondence between clusters of different systems is not known. Three clustering combination algorithms were introduced and the results show that significant gains, up to 40% relative in some cases, can be realized when combining multiple, diverse partitions. In addition, a cluster validity measure is proposed that uses the agreement across different systems to assess the degree to which a cluster maps to a real entity. The cluster validity measure is shown to be effective, achieving a correlation coefficient with the F-measure of 0.66.

Selectively adding word pairs to the bag-of-words representation can improve text classification performance. Word pairs with information measure that is very different than the sum of the information measures of the individual words can be very helpful. For example, the feature "second hand" was deemed useful for the topic "smoking" since "second hand" is more informative than "second" and "hand" in isolation. Another example is the feature "big brother" for the topic "TV reality shows" since again "big brother" is more informative than "big" and "brother". The suggested measure offered gains across 3 different corpora and 2 classifiers which shows that it is not genre- or classifier- specific.

We were able to demonstrate the presence of profound lexical differences between gen-

ders in conversations. Classifying a male-only vs female-only conversation can be achieved with almost perfect accuracy using an SVM and word pairs as features. Furthermore, we were able to demonstrate that the gender of one conversation side influences the lexical patterns of the other conversation side. From the engineering perspective, these differences can be potentially leveraged to improve a number of language processing tasks such as language modeling and dialog act classification, although we found that gender-dependent topic models or stopword lists do not contribute to an improved topic classification performance.

Disfluencies do not considerably impact the performance of a system that maps spoken conversations to topics. Using a subset of Switchboard-I annotated for disfluencies we have found that removing disfluencies does not considerably impact the topic classification performance, when using the bag-of-words representation. Somewhat bigger differences were observed when using word pairs as features. Feature selection techniques can partly compensate for disfluencies, since words commonly occurring inside a disfluency are words that can be robustly determined that are not associated with any topic. Also, removing disfluencies from the ASR transcripts does not appear to significantly improve topic classification performance. The best result, though, was obtained with the probabilistic index classifier, and using bigrams on disfluency-removed text.

We were able to demonstrate that automatically detected prominence can facilitate systems that map spoken conversations to topics. Our work showed that even the imperfect output of a prominence detection system, with a classification error of approximately 20%, is very useful in determining which words are less important for topic classification and clustering. Unlike previous work that attempts to link prosodic patterns with keywords, we found that it is the absence of prominence that reliably indicates the least important words. Therefore prominence can best be used to filter non-keywords rather than select keywords. An exciting result was that the gains of combining prominence and existing lexical measures of word saliency were additive and also that the gains were not specific to the classification method. An advantage of using prominence information is that it can be equally useful under no topically annotated data, in contrast to lexical measures that degrade with less annotated data. Another important finding, was that the average prominence on all occurrences of a word was more useful than the prominence value of

individual occurrences. These results should be interpreted under the caveat that all the experiments are on conversational speech, and the conclusions for other types of speech, such as broadcast news, may be different, because accent patterns are different and automatic accent detection is probably more reliable

The impact of ASR errors on topic classification was also investigated. It was shown that a WER of 30% (40%) causes a 15% (45%) drop in topic classification performance irrespective of the amount of training data. A method to cluster ASR confusable words was suggested that does not rely on any supervision, both in terms of correct transcripts or topic labels. The proposed method makes use of confusion networks to cluster words that highly co-occur and was found to offer improvement over using the ASR 1-best hypotheses. An interesting finding was that morphologically similar words tend to be confusable but also tend to be about the same topics, so the effect of word clusters is similar to that of a stemmer. This would explain why a WER of 30% introduces a disproportionally lower degradation in topic classification performance.

Some of these contributions are applicable in areas outside topic classification/clustering of human-human conversations. For example, the clustering combination methods can be applied, in principle, in any clustering problem, whether the attributes are continuous or discrete. In addition, all the challenges of performing topic classification/clustering in human-human conversations are applicable in spoken document retrieval. The role of prominence, the impact of disfluencies and errors introduced by the ASR system, are all issues that are important for spoken document retrieval as well.

9.2 Future Directions

The approaches presented in this dissertation can be extended in a number of ways. Below we briefly mention future directions.

The clustering combination methods presented in Chapter 3 were found to benefit some clustering algorithms more than others. For example, the gains from combining different partitions of mixture of multinomials were consistently higher than other algorithms. One possible explanation for this is that the diversity of partitions is a factor; the higher the

diversity the more beneficial the combination will be. However, defining diversity of partitions is not straightforward. Finding measures that can predict the gains of combining partitions can lead to meta-learning paradigms, where the parameters of the clustering algorithm are set so as not to maximize the objective function in a single run, but to maximize the performance of the combined partition.

The feature augmentation method as presented in Chapter 4, is applicable only to the case of augmenting an initial set of unigrams with selected bigrams. Extending the measure for higher order n-grams can be done in a number of ways. One approach is to subtract from the relevance measure of the n-gram sequence the relevance measures of all the possible subsequences of length $k < n$. Also, the proposed measure can be used for non-contiguous word pairs, although in such a case the number of possible pairs will be much higher than the number of unique bigrams.

The analysis in Chapter 5 on gender differences in conversations can be enhanced by integrating a number of discourse-related features. In our analysis we mentioned a number of non-lexical features that were found to be useful in our analysis, mainly laughter, backchannel/acknowledgment and filled pauses. It is possible that features derived from turn-taking and speech overlaps exhibit different patterns between genders. Other discourse features such as agreement/disagreement, as quantified in [69] may also be useful. In addition, it will be interesting to use other sociolinguistic variables such as age or social background. For example, the list of the most discriminative words shown in Table 5.6 will probably be different when conditioning in specific age groups. Also, although we were not able to show that gender information can be used to improve language modeling, it is possible that more sophisticated approaches in integrating gender information can offer improvements. An obvious shortcoming of our attempt is data sparsity, i.e. by conditioning every bigram according to gender the training data are split, increasing the variance of the estimations. Smoothing techniques may partly alleviate the data sparsity problem. For example, one approach would be to condition only the top N bigrams according to KL and tie the estimates of the remaining ones. Gender information may also prove useful in dialogue act tagging [156], where the priors of categories such as backchannels, yes/no answers and turn exits may depend on gender.

In Chapter 6, it was demonstrated that removing disfluencies has a marginal impact on the topic classification performance, even when the disfluent segments can be identified without errors. Therefore, investigating the effect that automatic disfluency detection systems have on topic classification performance is not very meaningful. However, disfluencies may be important for topic classification if more complex representations are used. For example, disfluencies are likely to impact syntactic parsing, therefore a representation that is based on parsing, as in [159] where head-modifier tuples are extracted, is likely to be impacted. In addition, other linguistic phenomena that are more profound in conversational speech, such as the use of pronouns, can be important for topic classification. Pronouns are much more likely to occur in conversational speech than in written text. In [137] it is reported that 14% of word in spoken language text are pronouns vs. 2% in written text. Since pronouns substitute for nouns or noun phrases that are generally considered to convey semantic information, they may have a negative impact on topic clustering or classification performance. It would be of interest to investigate whether pronouns have a significant impact and if so whether co-reference information could be used to improve topic classification for conversational speech.

The use of prominence for topic classification, as demonstrated in Chapter 7, can be improved in a number of ways. Prominence detection can be casted as a regression problem rather than a classification problem. Using soft outputs from the prominence classifier is an implicit way of doing so, but directly modeling it as a regression problem may be more appropriate. Some modes of prominence, such as emphatic prominence, may be detected using unsupervised methods, by measuring the difference of the prosodic pattern with the average pattern. Recently, approaches that view prominence as a continuous phenomenon rather than a binary one, and are trained with no supervision have found some success [169]. Integrating prominence with the output of the ASR system is also of interest, our work has only investigated the output using the true transcripts.

The method presented in Chapter 8 may be improved by using a combination of confusion networks and clustering. Instead of using verbatim the values that are output from confusion networks or creating hard clusters of confusable words, a better strategy may be to use a linear combination of the two. For example, at each position k, confusion networks

produce a vector $\vec{W}_k^{CONF}$. By clustering vectors $\vec{W}_k^{CONF}$, $\forall k$ we can obtain centroids of clusters $\vec{W}_c^{CLUST}$, where $c = g(\vec{W}_k^{CONF})$ is the cluster index. Instead of using only $\vec{W}_k^{CONF}$ or only $\vec{W}_c^{CLUST}$ we can use a linear combination of the two. Also, the method of Chapter 8 can be used for the unsupervised learning of morphology. Based on the assumption that morphologically similar words have high ASR confusability and also are topically similar, we can design an algorithm that given a new language with the above constraints will cluster morphological similar words. Given the clusters, a number of rule-learning methods can be applied to learn the suffixes of a new language. One possible modification to the algorithm would be to cluster ASR confusable words only if their topic similarity is above a threshold. A similar approach is described in [6] where instead of using an ASR confusability measure, the string edit distance between two words is calculated. In [6] it is empirically demonstrated that if two words have small string edit distance and have similar semantic representations then they are very likely to be morphological variants of the same stem. In [6], the semantic representation of a word w is defined as a vector with the counts of the words that appear in a window of 50 around each occurrence of w. A shortcoming of using string edit distance is that all characters are equally considered. Possibly, a scheme that uses syllables instead of characters may be more successful, therefore using the ASR confusability may be more advantageous. A shortcoming of ASR confusability measure is that it will be less likely that two words stemming from the same lemma but being of different part-of-speech categories will have high confusability.

BIBLIOGRAPHY

[1] R Agrawal and R. Srikant. Privacy-preserving data mining. In *Proc. of the ACM SIGMOD Conference on Management of Data*, pages 439–450. ACM Press, May 2000.

[2] H. Alshawi. Effective utterance classification with unsupervised phonotactic models. In *Proc. of Human Language Technologies/North American Chapter of the Association for Computational Linguistics conference (HLT/NAACL)*, pages 1–7, 2003.

[3] J. Ang, R. Dhillon, A. Krupski, E. Shriberg, and A. Stolcke. Prosody-based automatic detection of annoyance and frustration in human-computer dialog. In *Proc. of International Conference on Spoken Language Processing (ICSLP)*, volume 3, pages 2037–2040, 2002.

[4] F. Azuaje. A cluster validity framework for genome expression data. *Bioinformatics*, 18(2):319–320, 2002.

[5] A. Banerjee, S. Merugu, Inderjit Dhillon, and J. Ghosh. Clustering with Bregman divergences. In *Proc. of SIAM International Conference on Data Mining (SDM)*, pages 234–245, 2004.

[6] M. Baroni, J. Matiasek, and H. Trost. Unsupervised discovery of morphologically related words based on orthographic and semantic similarity. In *Proc. of 6th workshop on ACL Special Interest Group in Computational Phonology (SIGPHON)*, pages 48–57, 2002.

[7] S. Basu, A. Banerjee, and R. Mooney. Semi-supervised clustering by seeding. In *Proc. of International Conference on Machine Learning (ICML)*, pages 19–26, 2002.

[8] S. Basu, M. Bilenko, and R. Mooney. A probabilistic framework for semi-supervised clustering. In *Proc. of ACM Special Interest Group on Knowledge Discovery in Databases (SIGKDD)*, pages 59–68, 2004.

[9] E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36(1-2):105–139, 1999.

[10] D. Beeferman, A. Berger, and J. Lafferty. Statistical models for text segmentation. *Machine Learning*, 34(1-3):177–210, 1999.

[11] R. Bekkerman, R. El-Yaniv, N. Tishby, and Y. Winter. Distributional word clusters vs. words for text categorization. *Machine Learning Research*, 3:1183–1208, 2003.

[12] W. Belfield and H. Gish. A topic classification system based on parametric trajectory mixture models. In *Proc. of Eurospeech03*, pages 1269–1272, 2003.

[13] J. Bellegarda. Large vocabulary speech recognition with multi-span statistical language models. *IEEE Transactions on Speech and Audio Processing*, 8(1):76–84, 2000.

[14] A. Ben-Hur, A. Elissee, and I. Guyon. A stability based method for discovering structure in clustered data. In *Pacific Symposium on Biocomputing*, pages 6–17, 2002.

[15] K. Bessho, K. Ohtsuki, N. Hiroshima, S. Matsunaga, and Y. Hayashi. Topic structure extraction for meeting indexing. In *Proc. of International Conference on Spoken Language Processing (ICSLP)*, pages 1713–1716, 2004.

[16] M. Bilenko, S. Basu, and R.J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Proc. of the International Conference on Machine Learning (ICML)*, 2004.

[17] J. Bilmes. *Natural Statistical Models for Automatic Speech Recognition*. PhD thesis, University of California, Berkeley, EECS department, 1998.

[18] J.-M. Blanc and P.F. Dominey. Using prosody to discriminate between function and content words. In *Proc. of International Conference on Speech Prosody*, 2004.

[19] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Machine Learning Research*, 3:993–1022, 2003.

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

[21] C. Boulis and M. Ostendorf. Combining multiple clustering systems. In *8th European conference on Principles and Practice of Knowledge Discovery in Databases (PKDD), LNAI 3202*, pages 63–74, 2004.

[22] C. Boulis and M. Ostendorf. Text classification by augmenting the bag-of-words representation with redundancy-compensated bigrams. In *Proc. of the International Workshop in Feature Selection in Data Mining*, pages 9–16, 2005.

[23] P. Bradley and U. Fayyad. Refining initial points for K-Means clustering. In *Proc. 15th International Conf. on Machine Learning, (ICML)*, pages 91–99, 1998.

[24] P. Brown, V. DellaPietra, P. deSouza, J. Lai, and R. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, 1992.

[25] I. Bulyko, M. Ostendorf, and A. Stolcke. Getting more mileage from web text sources for conversational speech language modeling using class-dependent mixtures. In *Proc.of the Human Language Technologies/North American Chapter of the Association for Computational Linguistics conference (HLT-NAACL)*, pages 7–10, 2003.

[26] W. Byrne, D. Doermann, M. Franz, S. Gustman, J. Hajič, D. Oard, M. Picheny, J. Psutka, B. Ramabhadran, D. Soergel, T. Ward, and W-J Zhu. Automatic recognition of spontaneous speech for access to multilingual oral history archives. *IEEE Transactions on Speech and Audio Processing*, 12(4):420–435, 2004.

[27] B. A. Carlson. Unsupervised topic clustering of Switchboard speech messages. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 315–319, 1996.

[28] P. Cheeseman and J. Stutz. Bayesian classification (AutoClass): Theory and results. In *Advances in Knowledge Discovery and Data Mining*. AAAI Press/MIT Press, 1996.

[29] C. Chelba and A. Acero. Discriminative training of n-gram classifiers for speech and text routing. In *Proc. of Eurospeech03*, volume 4, pages 2777–2780, 2003.

[30] B. Chen, H.-M. Wang, and L.-S. Lee. Improved spoken document retrieval by exploring extra acoustic and linguistic cues. In *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech 2001)*, volume 1, pages 299–302, 2001.

[31] J. Chu-Carrol and B. Carpenter. Vector-based natural language call routing. *Computational Linguistics*, 25(3):361–388, 1999.

[32] C. Cieri, D. Miller, and K. Walker. The Fisher corpus: a resource for the next generations of speech-to-text. In *Proc. of the 4th International Conference on Language Resources and Evaluation, LREC*, pages 69–71, 2004.

[33] J. Coates, editor. *Language and Gender: A Reader*. Blackwell Publishers, 1997.

[34] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to extract symbolic knowledge from the World Wide Web. In *Proceedings of the 15th meeting of the American Association for Artificial Intelligence (AAAI-98)*, 1998.

[35] S. Deerwester, S.T. Dumais, G.W Furnas, T.K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.

[36] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, Series B, 39(1):1–38, November 1977.

[37] I. Dhillon, Y. Guan, and B. Kullis. Kernel k-means, spectral clustering and normalized cuts. In *Proc. of ACM Special Interest Group on Knowledge Discovery in Databases (SIGKDD)*, 2004.

[38] I. Dhillon, S. Mallela, , and R. Kumar. A divisive information-theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Research(JMLR)*, 3:1265–1287, 2003.

[39] I. Dhillon and D. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1/2):143–175, 2001.

[40] E. Dimitriadou, A. Weingessel, and K. Hornik. A combination scheme for fuzzy clustering. *Inter. J. of Pattern Recognition and Artificial Intelligence*, 16(7):901–912, 2002.

[41] G. Doddington. Speaker recognition based on idiolectal differences between speakers. In *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech 2001)*, pages 2251–2254, 2001.

[42] S. Dudoit and J. Fridlyand. A prediction-based resampling method to estimate the number of clusters in a dataset. *Genome Biology*, 3(7):1–21, 2002.

[43] S. Dudoit and J. Fridlyand. Bagging to improve the accuracy of a clustering procedure. *Bioinformatics*, 19(9):1090–1099, 2003.

[44] J. Dy and C. Brodley. Feature selection for unsupervised learning. *Machine Learning Research*, 5:845–889, 2004.

[45] P. Eckert and S. McConnell-Ginet, editors. *Language and Gender*. Cambridge University Press, 2003.

[46] X. Fern and C. Brodley. Random projection for high dimensional data: A cluster ensemble approach. In *Proc. of the 20th International Conf. on Machine Learning, (ICML)*, pages 186–193, 2003.

[47] B. Fischer and J. Buhmann. Bagging for path-based clustering. *IEEE Trans. on Patern Analysis and Machine Intelligence*, 25(11):1411–1415, 2003.

[48] J. Foote, S. Young, G. Jones, and K. Sparck-Jones. Unconstrained keyword spotting using phone lattices with application to spoken document retrieval. *Computer Speech and Language*, 11:207–224, 1997.

[49] G. Forman. An extensive empirical study of feature selection metrics for text classification. *Machine Learning Research*, 3:1289–1305, 2003.

[50] C. Fraley and A. Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631, 2002.

[51] A. Fred and A. Jain. Data clustering using evidence accumulation. In *Proc. of the International Conference on Pattern Recognition*, pages 276–280, 2002.

[52] D. Frossyniotis, M. Pertselakis, and M. Stafylopatis. A multi-clustering fusion algorithm. In *Proc. of the 2nd Hellenic Conference on Artificial Intelligence*, pages 225–236, 2002.

[53] J. Frunkranz, T. Mitchell, and E. Riloff. A case study in using linguistic phrases for text categorization on the WWW. In *Working Notes of the AAAI/ICML Workshop on Learning for Text Categorization*, 1998.

[54] N. Fuhr. Models for retrieval with probabilistic indexing. *Inf. Process. Manage.*, 25(1):55–72, 1989.

[55] H. Gabow, Z. Galil, T. Spencer, and R. Tarjan. Efficient algorithms for finding minimum spanning trees in undirected and directed graphs. *Combinatorica*, 6(2):109–122, 1986.

[56] J. Garofolo, C. Auzanne, and E. Voorhees. The TREC spoken document retrieval track: A success story. In *Proc. of the Recherche d'Informations Assiste par Ordinateur: ContentBased Multimedia Information Access Conference*, 2000.

[57] D. Gildea and T. Hofmann. Topic-based language models using EM. In *Proc. of Eurospeech*, pages 2167–2170, 1999.

[58] S. Godbole, A. Harpale, S. Sarawagi, and S. Chakrabarti. Document classification through interactive supervision of document and term labels. In *Proc. of the European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pages 185–196, 2004.

[59] J. Godfrey, E. Holliman, and J. McDaniel. Switchboard: Telephone speech corpus for research development. In *Proc. of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 517–520, 1992.

[60] D. Gondek and T. Hofmann. Non-redundant data clustering. In *Proc. of IEEE International Conference on Data Mining (ICDM)*, pages 75–82, 2004.

[61] J. Goodman. A bit of progress in language modeling. Technical Report MSR-TR-2001-72, Microsoft Research, 2001.

[62] J. Goodman. Classes for fast maximum entropy training. In *Proc. of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2001.

[63] J. Goodman and J. Gao. Language model size reduction by pruning and clustering. In *Proc. of International Conference on Spoken Language Processing (ICSLP)*, 2000.

[64] A.L. Gorin, G. Riccardi, and J.H. Wright. How may I help you? *Speech Communication*, 23:113–127, 1997.

[65] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Machine Learning Research*, 3:1157–1182, 2003.

[66] D. Hakani-Tür, G. Tür, M. Rahim, and G. Riccardi. Unsupervised and active learning in automatic speech recognition for call routing. In *Proc. of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 429–432, 2004.

[67] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2-3):107–145, 2001.

[68] V. Hatzivassiloglou, L. Gravano, and A. Maganti. An investigation of linguistic features and clustering algorithms for topical document clustering. In *Proc. of ACM Special Interest Group on Information Retrieval (SIGIR)*, pages 224–231, 2000.

[69] D. Hillard, M. Ostendorf, and E. Shriberg. Detection of agreement vs. disagreement in meetings: Training with unlabeled data. In *Proc. of the Human Language Technologies/North American Chapter of the Association for Computational Linguistics conference (HLT/NAACL)*, pages 34–36, 2003.

[70] D. Hillard, M. Ostendorf, A. Stolcke, Y. Liu, and E. Shriberg. Improving automatic sentence boundary detection with confusion networks. In *Proc. of Human Language Technologies/North American Chapter of the Association for Computational Linguistics conference (HLT-NAACL)*, pages 69–72, 2004.

[71] T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of 15th Conference on Uncertainty in Artificial Intelligence, UAI'99*, pages 289–296, 1999.

[72] M. Horne, P. Hansson, G. Bruce, and J. Frid. Prosodic correlates of information structure in Swedish human-human dialogues. In *Proc. of Eurospeech*, pages 29–32, 1999.

[73] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, pages 193–218, 1985.

[74] R. Iyer, H. Gish, and D. McCarthy. Unsupervised training techniques for natural language call routing. In *Proc. of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3090–3093, 2002.

[75] R. Iyer and M. Ostendorf. Modeling long distance dependencies in language: Topic mixtures versus dynamic cache model. *IEEE Trans. Speech and Audio Processing*, 7(1):30–39, 1999.

[76] A. Jain and R. Dubes, editors. *Algorithms for Clustering Data*. Prentice Hall, 1988.

[77] A. Jain, M. Murty, and P. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):267–323, 1999.

[78] T. Joachims. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *Proc. of International Conference on Machine Learning (ICML)*, 1997.

[79] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proc. of 10th European Conference on Machine Learning (ECML)*, 1998.

[80] T. Joachims. *Making large-Scale SVM Learning Practical*. MIT-Press, 1999.

[81] T. Joachims. Transductive inference for text classification using support vector machines. In *Proc. of the International Conference on Machine Learning (ICML)*, 1999.

[82] T. Joachims. *Learning to Classify Text Using Support Vector Machines*. PhD thesis, University of Dortmund, 2002.

[83] G.H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *Proceedings of the 11th International Conference on Machine Learning (ICML)*, pages 121–129, 1994.

[84] 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. of Eurospeech*, pages 1585–1588, 2003.

[85] H-K. Juo and C-H. Lee. Discriminative training of natural language call routers. *IEEE Trans. on Speech and Audio*, 11(1):24–35, 2003.

[86] S. Kiesling. Dude. *American Speech*, 79(3):281–305, 2004.

[87] D. Klein, S.D. Kamvar, and C.D. Manning. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *Proc. of the International Conference on Machine Learning (ICML)*, 2002.

[88] R. Kneser and H. Ney. Improved backing-off for m-gram language modeling. In *Proc. Intl. Conf. on Acoustics, Speech and Signal Processing International Conference on Acoustics, Speech, and Signal Processing (ICASSP))*, pages 181–184, 1987.

[89] R. Kohavi and G.H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[90] D. Koller and M. Sahami. Toward optimal feature selection. In *Proceedings of 16th International Conference on Machine Learning (ICML)*, pages 284–292, 1996.

[91] M. Koppel, S. Argamon, and A.R. Shimoni. Automatically categorizing written texts by author gender. *Literary and Linguistic Computing*, 17(4):401–412, 2002.

[92] K. Koumpis and S. Renals. The role of prosody in a voicemail summarization system. In *Proc. of ISCA Workshop on Prosody in Speech Recognition and Understanding*, pages 87–92, 2001.

[93] H.W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, pages 83–97, 1955.

[94] K. Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning (ICML)*, pages 331–339, 1995.

[95] T. Lange, V. Roth, M. L. Braun, and J. M. Buhmann. Stability-based validation of clustering solutions. *Neural Computation*, 16(6):1299 – 1323, June 2004.

[96] M. Law, M. Figueiredo, and A. Jain. Simultaneous feature selection and clustering using mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1154–1166, 2004.

[97] M.H. Law, A. Topchy, and A.K. Jain. Model-based clustering with probabilistic constraints. In *Proc. of SIAM Data Mining conference*, pages 641–645, 2005.

[98] C. Lee and G.G. Lee. MMR-based feature selection for text categorization. In *Proc. of the Human Language Technologies/North American Chapter of the Association for Computational Linguistics conference (HLT/NAACL*, pages 5–8, 2004.

[99] G-A. Levow. Prosody-based topic segmentation for Mandarin broadcast news. In *Proc. of the Human Language Technologies/North American Chapter of the Association for Computational Linguistics conference (HLT/NAACL)*, pages 137–140, 2004.

[100] D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In *Proceedings of the 15th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, 1992.

[101] L. Li and W. Chou. Improving latent semantic indexing classifier with information gain. In *Proc. of International Conference on Spoken Language Processing (ICSLP)*, volume 2, pages 1141–1144, 2002.

[102] Y. Liu, E. Shriberg, and A. Stolcke. Automatic disfluency identification in conversational speech using multiple knowledge sources. In *Proc. of Eurospeech*, pages 957–960, 2003.

[103] B. Logan, D. Goddeau, and J.M. Van Thong. Real-world audio indexing systems. In *Proc. of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1001–1004, 2005.

[104] J.B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. of the Fifth Symposium on Math, Statistics and Probability*, pages 281–297, 1967.

[105] L. Mangu, E. Brill, and A. Stolcke. Finding consensus in speech recognition: Word error minimization and other applications of confusion networks. *Computer Speech and Language*, 14(4):373–400, October 2000.

[106] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, 1999.

[107] A. McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/ mccallum/bow, 1996.

[108] A. McCallum. Multi-label text classification with a mixture model trained by EM. In *Proc. of AAAI Workshop on Text Learning*, 1999.

[109] A. McCallum and K. Nigam. A comparison of event models for Naive Bayes text classification. In *Proceedings of AAAI-98 Workshop on Learning for Text Categorization*, 1998.

[110] A. McCallum and K. Nigam. Text classification by bootstrapping with keywords, EM and shrinkage. In *Proc. of the workshop on Unsupervised Learning in Natural Language Processing*, pages 52–58, 1999.

[111] A. McCallum, R. Rosenfeld, T. Mitchell, and A. Ng. Improving text classification by shrinkage in a hierarchy of classes. In *Proc. of International Conference on Machine Learning (ICML)*, 1998.

[112] J. McCarley and M. Franz. Influence of speech recognition errors on topic detection. In *Proc. of ACM Special Interest Group on Information Retrieval (SIGIR)*, 2000.

[113] J. McDonough, K. Ng, P. Jeanrenaud, H. Gish, and J. R. Rohlicek. Approaches to topic identification on the Switchboard corpus. In *Proc. of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 385–390, 1994.

[114] M. Meilă and D. Heckerman. An experimental comparison of model-based clustering methods. *Machine Learning*, 42(1-2):9–29, 2001.

[115] M. Meteer and et al. Disfluency annotation stylebook for the Switchboard corpus. In *Linguistic Data Consortium*, 1995.

[116] P. Mitra, C. Murthy, and S. Pal. Unsupervised feature selection using feature similarity. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(3):301–312, 2002.

[117] S. Monti, P. Tamayo, J. Mesirov, and T. Golub. Consensus clustering: a resampling-based method for class discovery and visualization of gene-expression microaray data. *Machine Learning*, 52(1–2):91–118, 2003.

[118] A. Moschitti and R. Basili. Complex linguistic features for text classification: A comprehensive study. In *Proceedings of the 26th European Conference on Information Retrieval (ECIR)*, 2004.

[119] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Proc. of Neural Information Processing Systems (NIPS)*, 2002.

[120] K. Ng. *Subword-based Approaches for Spoken Document Retrieval*. PhD thesis, MIT, Electrical Engineering and Computer Science Dept., 2000.

[121] K. Ng and V. Zue. Subword-based approaches for spoken document retrieval. *Speech Communication*, 32(3):157–200, October 2000.

[122] K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *Proc. of the IJCAI-99 Workshop on Machine Learning for Information Filtering*, pages 61–67, 1999.

[123] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Learning to classify text from labeled and unlabeled documents. In *Proc. of AAAI*, pages 792–799, 1998.

[124] K. Nigam, A.K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2-3):103–134, 2000.

[125] E. Nöth, A. Batliner, A. Kießling, R. Kompe, and H. Nietmann. VERBMOBIL: The use of prosody in the linguistic components of a speech understanding system. *IEEE Trans. on Speech and Audio Proc.*, 8(5):519–532, 2000.

[126] P. Nowell and R. Moore. The application of dynamic programming to non-word based topic spotting. In *Proc. of Eurospeech95*, volume 2, pages 1355–1358, 1995.

[127] M. Ostendorf, E. Shriberg, and A. Stolcke. Human language technology: Opportunities and challenges. In *Proc. of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 5, pages 949–952, 2005.

[128] D. Palmer and M. Ostendorf. Improving information extraction by modeling errors in speech recognizer output. In *Proceedings of Human Language Technolgies conference (HLT)*, pages 1–5, 2001.

[129] S. Pan and K. McKeown. Word informativeness and automatic pitch accent modeling. In *Proc. of the Joint SIGDAT Conference on EMNLP and VLC*, pages 148–157, 1999.

[130] C. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala. Latent semantic indexing: A probabilistic analysis. In *Proc. of PODS*, pages 159–168, 1998.

[131] F. Peng, D. Schuurmans, and S. Wang. Language and task independent text categorization with simple language models. In *Proc.of the Human Language Technologies/North American Chapter of the Association for Computational Linguistics conference (HLT/NAACL)*, pages 110–117, 2003.

[132] M. Porter. An algorithm for suffix stripping. In *Proc. of Multimedia Information And Systems Series, Readings in information retrieval*, pages 313–316, 1997.

[133] B. Raskutti, H. Ferra, and A. Kowalczyk. Second order features for maximizing text classification performance. In *Proc. of the 12th European Conference on Machine Learning (ECML)*, 2001.

[134] G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24:513–523, 1988.

[135] R.E. Schapire and Y. Singer. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168, 2000.

[136] B. Schölkopf, A. Smola, and K. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[137] S. Schwarm, I. Bulyko, and M. Ostendorf. Adaptive language modeling with varied sources to cover new vocabulary items. *IEEE Trans. on Speech and Audio Processing*, 12(3):334–342, May 2004.

[138] R. Schwartz, T. Imai, F. Kubala, L. Nguyen, and J. Makhoul. A maximum likelihood model for topic classification of broadcast news. In *Proc. of Eurospeech97*, pages 1455–1458, 1997.

[139] S. Scott and A. Matwin. Text classification using wordnet hypernyms. In *Proc. of the workshop on the Usage of Wordnet in Natural Language Processing Systems*, pages 45–51, 1998.

[140] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[141] K. Seymore and R. Rosenfeld. Using story topics for language model adaptation. In *Proc. of Eurospeech*, pages 1987–1990, 1997.

[142] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[143] E. Shriberg. *Preliminaries to a theory of speech disfluencies*. PhD thesis, University of California, Berkeley, 1994.

[144] E. Shriberg and et al. Can prosody aid the automatic classification of dialog acts in conversational speech? *Language and Speech*, 41(3-4):439–487, 1998.

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

[146] M. Siegler and M. Witbrock. Improving the suitability of imperfect transcriptions for information retrieval from spoken documents. In *Proc. of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 505–508, 1999.

[147] R. Silipo and F. Crestani. Prosodic stress and topic detection in spoken sentences. In *Proc. of SPIRE*, pages 242–252, 2000.

[148] S. Singh. A pilot study on gender differences in conversational speech on lexical richness measures. *Literary and Linguistic Computing*, 16(3):251–264, 2001.

[149] S. Sista, R. Schwartz, T. Leek, and J. Makhoul. An algorithm for unsupervised topic discovery from broadcast news stories. In *Proc. of Human Language Technologies conference (HLT)*, pages 99–103, 2002.

[150] N. Slonim. *The information bottleneck: Theory and applications*. PhD thesis, Hebrew University of Jerusalem, 2003.

[151] N. Slonim and N. Tishby. The power of word clusters for text classification. In *Proc. of 23rd European Colloquium on Information Retrieval Research (ECIR)*, 2001.

[152] E. Stamatatos, N. Fakotakis, and G. Kokkinakis. Automatic text categorization in terms of genre and author. *Computational Linguistics*, 26(4):471–495, 2000.

[153] A. Stolcke. An extensible language modeling toolkit. In *Proc. Intl. Conf. on Spoken Language Processing International Conference on Spoken Language Processing (ICSLP)*, pages 901–904, 2002.

[154] A. Stolcke. Speech-to-text research at SRI-ICSI-UW. In *Presentation at the NIST Rich Transcription Workshop, http://nist.gov/speech/tests/rt/rt2003/spring/presentations/sri+-rt03-stt.pdf*, 2003.

[155] A. Stolcke, X. Anguera, K. Boakye, O. Cetin, F. Grezl, A. Janin, A. Mandal, B. Peskin, C. Wooters, and J. Zheng. Further progress in meeting recognition: The ICSI-SRI spring 2005 speech-to-text evaluation system. In *Proc. of NIST MLMI Meeting Recognition Workshop*, 2005.

[156] A. Stolcke, E. Shriberg, R. Bates, N. Coccaro, D. Jurafsky, R. Martin, M. Meteer, K. Ries, P. Taylor, and C. Van Ess-Dykema. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–373, 2000.

[157] S. Strassel. Simple metadata annotation specification version 5.0 http://www.ldc.upenn.edu/projects/MDE/guidelines/SimpleMDE_v5.0.pdf, 2003.

[158] A. Strehl and J. Ghosh. Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Machine Learning Research*, 3:583–617, December 2002.

[159] T. Strzalkowski and S. Jones. NLP track at TREC-5. In *Proc. of Text Retrieval Conference*, 1996.

[160] C.-M. Tan, Y.-F. Wang, and C.-D. Lee. The use of bigrams to enhance text categorization. *Information Processing and Management*, 38(4):529–546, 2002.

[161] J-M. Van Thong, P. J. Moreno, B. Logan, B. Fidler, K. Maffey, and M. Moores. SPEECHBOT: An experimental speech-based search engine for multimedia content in the Web. Technical Report CRL-2001/06, Cambridge Research Laboratory, 2001.

[162] A. Topchy, A. Jain, and W. Punch. A mixture model for clustering ensembles. In *Proc. of SIAM Conference on Data Mining*, 2004.

[163] G. Tür, D. Hakkani-Tür, A. Stolcke, and E. Shriberg. Integrating prosodic and lexical cues for automatic topic segmentation. *Computational Linguistics*, 27:31–57, 2001.

[164] G. Tür, J. Wright, A. Gorin, G. Riccardi, and D. Hakkani-Tür. Improving spoken language understanding using word confusion networks. In *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, pages 1137–1140, 2002.

[165] N. Ueda and K. Saito. Parametric mixture models for multi-labeled text. In *Proc. of Neural Information Processing Systems (NIPS)*, 2002.

[166] V. Venkataramani and W. Byrne. Lattice segmentation and support vector machines for large vocabulary continuous speech recognition. In *Proc. of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 817–820, 2005.

[167] R. Vilalta, T. Stepinski, M. Achari, and F. Ocegueda-Hernandez. A quantification of cluster novelty with an application to martian topography. In *8th European conference on Principles and Practice of Knowledge Discovery in Databases (PKDD), LNAI 3202*, pages 434–445, 2004.

[168] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. In *Proc. of the International Conference on Machine Learning (ICML)*, pages 577–584, 2001.

[169] D. Wang and S. Narayanan. An unsupervised quantitative measure for word prominence in spontaneous speech. In *Proc. of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, pages 377–380, 2005.

[170] Y.Y. Wang. A robust parser for spoken language understanding. In *Proc. of Eurospeech*, volume 5, pages 2055–2058, 1999.

[171] M. Weintraub. Keyword-spotting using SRI's DECIPHER large-vocabularyspeech-recognition system. In *Proc. of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 463–466, 1993.

[172] D. Wong, M. Ostendorf, and J. Kahn. Using weakly supervised learning to improve prosody labeling. Technical Report UWEETR-2005-0003, University of Washington, Electrical Engineering Department, 2005.

[173] E. Xing, A. Ng, and M.Jordan. Distance metric learning with application to clustering with side information. In *Proc. of the Neural Information Processing Systems 15(Neural Information Processing Systems (NIPS))*, pages 505–512, 2003.

[174] J. Xu and W. Croft. Cluster-based language models for distributed retrieval. In *Proc. of ACM Special Interest Group on Information Retrieval (SIGIR)*, 1999.

[175] Y. Yang and J. Pedersen. A comparative study on feature selection in text categorization. In *Proc. of the 17th International Conference on Machine Learning (ICML)*, pages 412–420, 1997.

[176] K.Y. Yeung and R. Bumgarner. Multiclass classification of microarray data with repeated measurements: application to cancer. *Genome Biology*, 4(12):1–19, 2003.

[177] K.Y. Yeung, D. Haynor, and L. Ruzzo. Validating clustering for gene expression data. *Bioinformatics*, 17:309–318, 2001.

[178] K.Y. Yeung and W. Ruzzo. Principal component analysis for clustering gene expression data. *Bioinformatics*, 17(9):763–774, 2001.

[179] L. Yu and H. Liu. Efficient feature selection via analysis of relevance and redundancy. *Machine Learning Research*, 5:1205–1224, 2004.

[180] K. Zechner and A. Waibel. DIASUMM: Flexible summarization of spontaneous dialogues in unrestricted domains. In *Proceedings of Computational Linguistics conference (COLING)*, pages 968–974, 2000.

[181] Y. Zeng, J. Tang, J. Garcia-Frias, and G. Gao. An adaptive meta-clustering approach: Combining the information from different clustering results. In *Proc. IEEE Computer Society Bioinformatics Conference*, pages 276–281, August 2002.

[182] Y. Zhao and G. Karypis. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning*, 55(3):311–331, June 2004.

[183] I. Zitouni, H-K. Kuo, and C-H. Lee. Boosting and combination of classifiers for natural language call routing systems. *Speech Communication*, 41(4):647–661, 2003.

VITA

Constantinos Boulis was born in Athens, Greece and obtained a Diploma from the department of Electronics and Computer Engineering in the Technical University of Crete, Greece, in 1998. His Diploma thesis was on speaker adaptation for automatic speech recognition systems. He continued studies in the area of automatic speech recognition and obtained the Postgraduate degree in 2000, from the same department. He continued his graduate studies in the Electrical Engineering department in the University of Washington, Seattle, USA, where he initially worked on problems involving source and channel coding for distributed automatic speech recognition. He then conducted his doctoral dissertation on topic learning in text and conversational speech. He was awarded the PhD degree in August 2005.