STUDIES ON MACHINE LEARNING FOR DATA ANALYTICS IN BUSINESS APPLICATION

FANG FANG

(B.Mgmt.(Hons.), Wuhan University)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF INFORMATION SYSTEMS

NATIONAL UNIVERSITY OF SINGAPORE

2014

DECLARATION

I hereby declare that the thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Fang Fang

22 January 2014

ACKNOWLEDGEMENTS

I would like to thank many people who made this thesis possible.

First and foremost, it is difficult to overstate my sincere gratitude to my supervisor, Professor Anindya Datta. I appreciate all his contributions to my research, as well as his guidance and support in both my professional and personal time. It has been a great honor to work with him. I am also deeply indebted to Professor Kaushik Dutta, who has provided great encouragement and sound advice throughout my research journey.

I thank my fellow students and friends in NUS, especially members of the NRICH group, for providing such a warm and fun environment in which to learn and grow. I will never forget our stimulating discussions, our time when working together, and all the fun we have had.

Last but not least, I would like to thank my parents, for the unconditional support and love. To them I dedicate this thesis.

TABLE OF CONTENTS

SUMMARY

The volume of data produced by the digital world is now growing at an unprecedented rate. Data are being produced everywhere, from Facebook, Twitter, YouTube to Google search records, and more recently, mobile apps. The tremendous amount of data embodies incredible valuable information. Analysis of data, both structured and unstructured such as text, is important and useful to a number of groups of people such as marketers, retailers, investors, and consumers.

In this thesis, we focus on predictive analytics problems in the context of business applications and utilize machine learning methods to solve them. Specifically, we focus on 3 problems that can support a firm's business and management team's decision-making. We follow the Design Science Research Methodology (Hevner and Chatterjee 2010, Hevner et al. 2004) to conduct the studies.

Study I (chapter 2) focuses on cross-domain sentimental classification. Sentiment analysis is quite useful to consumers, marketers, and organizations. One of the tasks of sentiment analysis is to determine the overall sentiment orientation of a piece of text. Supervised learning methods, which require labeled data for training, have been proven quite effective to solve this problem. One assumption of supervised methods is that the training domain and the data domain share exactly the same distribution, otherwise, accuracy drops dramatically. However, in some circumstances, labeled data is quite expensive to acquire. For instance, Tweets and comments in Facebook. Study I addresses this problem and proposes an approach to determine the sentiment orientation of a piece

of text when in-domain labeled data is not available. The experimental results suggest that the proposed method outperforms all existing methods in literature.

Study II (chapter 3) focuses on Industry Classification. Industry analysis, which studies a specific branch of manufacturing, service, or trade, is quite useful for various groups of people. Before industry analysis, we need to define industry boundaries effectively and accurately. Existing schemes like SIC, GICS or NAICS have two major limitations. Firstly, they are all static and assume that the industry structure is stable. Secondly, these schemes assume binary relationship and do not measure the degree of similarity. Study II aims to contribute the literature by proposing an industry classification methodology that can overcome these limitations. Our method is on the basis of business commonalities using the topic features learned by the Latent Dirichlet Allocation (LDA) from firms' business descriptions.The experimental results indicate that the proposed approach is better than the GICS and the baseline.

Study III (chapter 4) focuses on mobile app download estimation. Mobile apps represent the fastest growing consumer product segment of all times. To be successful, an app needs to be popular. The most commonly used measure of app popularity is the number of times it has been downloaded. For a paid app, the downloads will determine the revenue the app generates; for an ad-driven app, the downloads will determine the price of advertising on this app. In addition, research in the app market necessities download numbers to measure the success of an app. Even though the app downloads are quite valuable, it turns out that number of downloads is one of the most closely guarded secrets in the mobile industry – only the native store knows the download number of an app.

Study III intends to propose a model of daily free app downloads estimation. The experimental results prove the effectiveness and accuracy of the proposed model.

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

1.1 BACKGROUND AND MOTIVATION

The volume of data produced by consumer activity is growing at an unprecedented rate. Data are being produced everywhere, from Facebook, Twitter, YouTube to Google search records, and more recently, mobile apps. According to recent research by International Data Corporation (IDC)[1], digital data that can be analyzed by computers will double about every two years from now until 2020 (Gantz and Reinsel 2012). IDC's report estimates that there will be 40,000 exabytes, or 40 trillion gigabytes, of digital data in 2020. Without doubt, the amount of data is huge.

The tremendous amount of data encapsulates much useful information. Analysis of this data, both structured and unstructured, is quite valuable and useful to various constituencies in the business community and critical for business success: (a) Marketers need to use customer profile data to differentiate among customers and then match customers with appropriate product offerings; (b) Retailers need to use transaction data to monitor the sales trends and then optimize inventory. (c) Investors need to use financial statement data to investigate company's competitiveness and then make investment decisions; (d) Consumers need to use text review data to research products and then make the final purchase. In a word, data analytics is extremely valuable.

[1] http://www.idc.com/

Data analytics may be classified into several categories: (1) *descriptive analytics* aims to provide descriptive statistics of data such as mean, average and so on; (2) *explanatory analytics* intends to use statistical methods to explain observed phenomena and explore causal relationships (Shmueli and Koppius 2011); (3) *predictive analytics* aims to use various machine learning techniques for forecasting future or unknown events. In this dissertation, we focus on applying predictive analytics methods to common business problems. Our motivation stems from the ubiquity of "predictive" problems in the business domain, but the relative paucity of work on applying predictive analytics techniques in this area. We explain this below.

The need to predict future events is paramount in many business scenarios: (a) revenue and profit forecasting, (b) predicting/classifying consumer types that would be interested in particular product lines, (c) predicting competitor actions and (d) predicting market reaction to new products, to just name a few. Given the abundance of situations needing "smart" predictions, it would appear that traditional machine learning predictive techniques would be a natural fit.

Machine learning has been extensively applied in a number of domains, mostly Science and Engineering areas such as Bioinformatics (Michiels et al. 2005, Tarca et al. 2007), Cheminformatics (Gehrke et al. 2008, Podolyan et al. 2010), Robotics (Conrad and DeSouza 2010) and so on. However, far less work has been done in business-related areas. In particular, in certain areas like Industry Classification, there is very little work which uses machine learning to address the problem. Recently, there is increasing research interest in the application of machine learning methods for business analytics

(Abbasi and Chen 2008, Rui and Whinston 2011) and results are promising. However, much more needs to be done.

In this thesis, we focus on predictive analytics problems in the context of business applications and utilize machine learning methods to solve them. In particular, we look at three classes of business problems that can support a firm's business and management team's decision-making: (1) *extracting sentiments expressed by users towards products*: the management team is always eager to know how products are received by the consumers and then modify the production plan accordingly. We fulfill this need by using the reviews text data written by consumers and extract their attitude towards the products. (2) *Industry classification*: The management team also likes to identify who the competitors are and adjust the company's business strategy accordingly. We contribute to this by using firms' 10-K forms and identifying firms involved in same business, which are therefore potentially competitors, and (3) *Competitor Sales Estimation*: The management team is also interested in the sales of products from other competitors so as to then adjust their product strategy accordingly. To know the exact sales volume of competitors by product line is quite hard, given the sensitivity of the data. In this thesis, we provide a solution in the mobile app domain due to the availability of data and use sale ranks to estimate the actual sales amount. The three problems chosen are due to their wide application in multiple business scenarios, and of course, each of these problems has received much attention lately in the literature. A brief introduction of the three problems is presented in the next section.

1.2 RESEARCH FOCUS AND POTENTIAL CONTRIBUTIONS

In this section, we will briefly introduce the research problems investigated in the thesis and also discuss potential contributions of each study. The first two studies use text data for analytics: study I aims to detect sentimental orientation embedded in the text and study II aims to classify firms into industries based on text descriptions of firms' business. Study III aims to estimate the sales of products. We select the domain of mobile apps due to the availability of data. In this thesis, we follow the Design Science Research Methodology (Hevner and Chatterjee 2010, Hevner et al. 2004) to conduct the studies.

1.2.1 Study I: Cross-domain Sentimental Classification

Sentiment analysis, which aims to detect the underlying sentiments embedded in texts, has attracted much research interest recently. Such sentiments are quite useful to consumers, marketers, organizations, etc. One of the tasks of sentiment analysis is to determine the overall sentiment orientation of a piece of text and *supervised learning methods*, which require labeled data for training, have been proven quite effective to solve this problem.

One assumption of supervised methods is that the training domain and the data domain share exactly the same distribution, i.e., (a) texts in both data sets are represented in same feature space and (b) features, or words, follow the same distributions in both data sets. The first assumption requires that a similar set of words are used in both domains, while the second assumption demands that the occurrence probability of a word is identical in training and testing domains. If these assumptions do not hold, accuracy drops dramatically (about 10% according to our experiment results). These assumptions do not pose problems when performing sentiment analysis in domains where training data are readily available.

However, in some circumstances, labeled data is quite expensive to acquire. For instance, if we want to detect sentiment from Tweets or comments in Facebook, the only way to get labeled data is by manually labeling and thus, it is prohibitively burdensome and time-consuming.

This is the problem addressed in this study - we want to determine the sentiment orientation of a piece of text when *in-domain* labeled data is not available. Particularly, we would like to contribute the literature by proposing an innovative method that can effectively perform cross-domain sentimental classification.

1.2.2 Study II: LDA-Based Industry Classification

Industry analysis, which studies a specific branch of manufacturing, service, or trade, is quite useful for various groups of people: asset managers, credit analysts, investors, researchers, etc. Before industry analysis, we need to define industry boundaries effectively and accurately. Otherwise, further industry analysis could become impossible, or at least misleading.

There exist a number of *Industry Classification* schemes such as the Standard Industrial Classification (SIC)[2] and the North American Industry Classification System (NAICS)[3]. However, these schemes have two major limitations. Firstly, they are all static and assume that the industry structure is stable (Hoberg and Phillips 2013). Secondly, these schemes assume binary relationship and do not measure the degree of similarity.

In this study, we aim to contribute the literature by proposing an industry classification methodology that can overcome these limitations. Our method is on the basis of business commonalities using the topic features learned by the Latent Dirichlet Allocation (LDA) (Blei et al. 2003) from firms' business descriptions.

1.2.3 Study III: Mobile App Download Estimation

Mobile apps represent the fastest growing consumer product segment of all time (Kim 2012). The production scale of apps is eye-popping as well – approximately 15000 new apps are launched every week (Datta et al. 2012). To be successful, an app needs to be popular. The most commonly used measure of app popularity is the number of times (which we will simply refer to as "downloads") it has been downloaded into consumers' smart-devices. For a paid app, the downloads will determine the revenue the app generates; for an ad-driven app, the downloads will determine the price of advertising on this app. In addition to its huge business value, app download numbers are also quite valuable from a research perspective. The rapid growth of the app market offers an

[2]http://www.census.gov/epcd/www/sic.html [Accessed May 1, 2013]
[3] http://www.census.gov/eos/www/naics/ [Accessed May 1, 2013]

excellent place for studies such as innovation (Boudreau 2011), competitive strategies in hypercompetitive markets (Kajanan et al. 2012). Studies in the app market necessities download numbers to measure the success of an app.

Even though app downloads are quite valuable, it turns out that number of downloads is one of the most closely guarded secrets in the mobile industry – only the native store knows the download number of an app. As a result, in recent times, there has been much interest in estimating app downloads (Garg and Telang 2012). However, the present study only focuses on paid apps. In this study, we intend to fill the gap by proposing a model for estimating daily free app downloads, which complements Garg and Telang (2012).

1.3 MACHINE LEARNING

Machine learning is a highly interdisciplinary field which borrows and builds upon ideas from statistics, computer science, engineering, cognitive science, optimization theory and many other disciplines of science and mathematics (Ghahramani 2004). It aims to construct computer programs/systems that can make decisions regarding unseen instances based on knowledge learnt from the training data. Tom Mitchell provided a widely quoted formal definition: “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E” (Mitchell 1997).

Machine learning methods can be categorized into several classes and two major types are supervised learning methods and unsupervised learning methods. Supervised methods require correct outputs for instances in training data, and their objective is to learn a function from the training data, which can produce a output for instances not in the training data. The output can be a class label for classification tasks and a real number for regression tasks. On the contrary, unsupervised methods do not require instances in training data to have correct outputs, and their purpose is to identify underlying patterns in the training data. One classic example of unsupervised learning is clustering, which aims to group similar instances as a cluster. Another example is topic models, such as the Latent Dirichlet Allocation (LDA) (Blei et al. 2003), whose goal is to discover underlying "topics" in a collection of documents.

Both supervised methods and unsupervised methods are used in this thesis. Specifically, supervised methods are used for cross-domain sentimental classification (study I) and mobile app downloads estimation (study III); unsupervised methods are used for industry classification (study II).

1.4 THESIS ORGANIZATION

The rest of this thesis is organized as follows: chapter 2 presents the study on cross-domain sentiment classification. In chapter 3, we propose a novel method for industry classification and peer identification. Chapter 4 discusses the estimation of mobile app downloads using rankings. Chapter 5 concludes this thesis.

CHAPTER 2 STUDY I: CROSS-DOMAIN SENTIMENTAL CLASSIFICATION USING MULTIPLE SOURCES

2.1 INTRODUCTION

With the explosion of blogs, social networks, reviews, ratings as well as other user-generated texts, *sentiment analysis*, which aims to detect the underlying sentiments embedded in those texts, has attracted much research interest recently. Such sentiments are useful to various constituencies: (a) Consumers can use sentiment analysis to research products or services before making a purchase. (b) Marketers can use this to research public opinion regarding their company and products, or to analyze customer satisfaction. Finally, (c) organizations can also use this to gather critical feedback about problems in newly released products.

One of the tasks of sentiment analysis is to determine the overall sentiment orientation of a piece of text. This problem has been widely investigated and *supervised learning methods*, which require labeled data for training, have been proven quite effective. However, supervised methods assume that the training data domain and the testing data domain share exactly the same distribution, i.e., (a) texts in both data sets are represented in same feature space and (b) features, or words, follow the same distributions in both data sets. The first assumption requires that a similar set of words are used in both domains, while the second assumption demands that the occurrence probability of a word is identical in training and testing domains. If these assumptions do not hold, accuracy drops dramatically (about 10% according to our experiment results). These assumptions do not pose problems when performing sentiment analysis in domains where training data are readily available. An example of such a domain is movie reviews. Each review is typically accompanied by a numerical rating, allowing easy assignment of sentiment to the review. In nearly all previous work, reviews rated 1 and 2 are considered as negative and those rated 4 and 5 are treated as positive. However, in circumstances where user-assigned ratings are not available, labeled data is quite expensive to acquire. For instance, if we want to detect sentiment from Tweets or comments in Facebook, the only way to get labeled data is to manually label it and thus, prohibitively burdensome and time-consuming. Yet, sentiment mining is pervasive enough such that its application is useful in many domains, such as Tweets and Facebook comments, where labeled data are not available.

This is the problem addressed in this study. We want to determine the sentiment orientation of a piece of text when *in-domain* labeled data is not available. A number of methods have been proposed in the literature most of which rely on the idea of applying labeled data from a "source" domain to perform sentiment classification on data in a different "target" domain through domain independent feature called *pivot* features. Following is an illustrative example. Suppose we are adapting from "computers" domain to "cell phones" domain. While many of the features of a good cell phone review are the same as a computer review, such as "excellent" and "awful", many words are totally new, like "reception". In addition, many features which are useful for computers, for instance "dual-core", are not useful for cell phones. The intuition is that even though the phrase “good-quality reception” and “fast dual-core” are completely distinct for each domain, they both have high correlation with “excellent” and low correlation with “awful” on unlabeled data. As a result, we can tentatively align them (Blitzer et al. 2007). After learning a classifier for computer reviews, when we see a cell-phone feature like “good-quality reception”, we know it should behave in a roughly similar manner to “fast dual-core”.

The main drawback of these methods is that the performance is largely dependent on the selection of pivot features. Ideally, pivot features would act similarly in both target and source domains towards sentiment. The problem is that we do not know the sentiment of the data in the target domain, making extremely hard to select those pivot features accurately.

In this study, we propose a hybrid approach that integrates the sentiment information from labeled data of multiple source domains and a set of preselected sentiment words for sentimental domain adaptation, *i.e., cross-domain* sentiment classification. In order to solve the aforementioned limitation caused by difficulty of pivot feature selection, we tackle this task by mapping the data into a latent space to learn an abstract representation of the text. The assumption we make is that texts with the same sentiment label would have similar abstract representations, even though their text representations differ. For instance, in the previous example, the phrase “good-quality reception” and “fast dual-core” are completely distinct for each domain; however, in the latent space, they might corresponds to the same feature. This idea has been used in Titov (2011) and Glorot et al (2011); however, as we will discuss later, our method is distinct enough from them.

Furthermore, in addition to use of *out-domain* data, we also utilize sentiment information from preselected opinionated words. We believe these words could provide certain helpful sentiment information in our classification context. Finally we train our classifiers over the new hybrid representations. The experimental results suggest that our method statistically outperforms the state of the art and even surpasses the *in-domain* method in some cases.

The rest of the chapter is organized as follows: we first review related work in literature. Then we provide the intuition and overview of our method followed by an elaboration of our proposed method. Whereafter, we evaluate our method on a benchmark data set. Finally, we conclude this chapter with a discussion of this study.

2.2 RELATED WORK

In this section, we review related work on in-domain sentiment classification, cross-domain sentiment classification as well as other sentiment analysis tasks.

2.2.1 In-domain Sentiment Classification

One of the most thoroughly studied problems in sentiment analysis is the *in-domain* sentiment classification, which refers to the process of determining the overall tonality of a piece of text and classifying it into several sentiment classes. Two main research directions have been explored, i.e., document level sentiment classification and sentence level sentiment classification.

In document level classification, documents are assumed to be opinionated and all documents are classified as either positive or negative (Liu 2010). This problem can be addressed as either supervised learning problem or unsupervised classification problem. Many of the existing research using supervised machine learning approach have used product reviews as target documents. Training and testing data are very convenient to collect for these documents since each review already has a reviewer-assigned rating, typically 1-5 stars. One representative work would be (Pang and Lee 2008). They employed multiple approaches to the sentiment classification problem and concluded that machine learning methods definitively outperform others.

Due to opinion words being the dominating indicators for sentiment classification, it is quite natural to use unsupervised learning based on such words. This kind of methods has not been studied so much because of its relatively inferior performance compared with supervised methods. The simplest method is to determine sentiment of a document based on the occurrences of positive and negative word. A review could be classified as positive if there are more positive words and categorized as negative otherwise. One representative example the more sophisticated work is Turney (2002). They performed classification based on certain fixed syntactic phrases that are likely to be used to express opinion. They first identified phrases with positive semantic orientation and phrases with negative semantic orientation. The semantic orientation of a phrase was calculated as the mutual information between the given phrase and the word “excellent” minus the mutual information between the given phrase and the word “poor”. A review was classified as positive if the average semantic orientation of its phrases is positive and categorized as negative otherwise.

In sentence level classification, sentences are first classified as subjective or objective. Then subjective sentences are further classified into positive or negative (Liu 2010). Traditional supervised learning methods have been applied here. Representative examples include Wiebet and Bruce (1999), which used a Naïve Bayesian classifier for subjectivity classification. Other learning algorithms are also used in subsequent research (Hatzivassiloglou and Wiebe 2000, Riloff and Wiebe 2003). One of the bottlenecks for this task is the lack of training example. A bootstrapping approach to automatically label training data was proposed in Riloff and Wiebe (2003) to solve this problem.

2.2.2 Cross-domain Sentiment Classification

Most sentiment classification methods assume that training data and testing data share exactly the same distribution. The assumption can be interpreted from two perspectives: (a) documents in both training domain and testing domain are represented using the same set of words; (b) words follow the same distribution. The first perspective necessitates that the same set of words are used in both training domain and testing domain while the second part obliges that the probability of a word occurring in training domain equals that of in testing domain. If these two assumptions are not met, accuracy of the classifier drops dramatically. A number of solutions have been proposed to solve this problem and all of them utilize labeled data from other domains, or source domains. Intuition in most existing research is to map features between the target domain and the source domain making use of domain independent feature known as pivot features. An illustrative example is given in the introduction section. The two kinds of pivot features were explored in literature: words (Blitzer et al. 2007, Bollegala et al. 2011, Pan et al. 2010) and topics (He et al. 2011, Liu and Zhao 2009). We discuss them in turn below.

Blitzer et al. (2007) started the line of research on cross-domain sentiment classification. They selected words as pivot features according to their common frequency and mutual information with the source labels, and then applied Structural Correspondence Learning (SCL) algorithm to obtain *k* new real-valued features. Finally, they augmented the original feature with the *k* new real-valued features in both source domain and target domain, and performed classification over the new feature space. Pan et al. (2010) also proposed a similar method. They selected words with low mutual information between words and domains as pivot features, and then run a Spectral Feature Alignment (SFA) algorithm to align domain-specific words. The classification was performed over the augmented feature space. Bollegala et al. (2011) also used words as pivot features but in a different manner. Instead of selecting a small set of domain-independent features, they treated all features as pivot features. Based on pointwise mutual information, relatedness between any two words was calculated. Then, they expanded the feature representation of a document with those words that are highly related with words in the document and trained classifiers over the new feature space. So far this is the only work that used multiple source domains simultaneously. Multiple source domains can also be used simultaneously in our approach but in a different manner. For example, (a) we use latent space model to learn latent representations; (b) we only rely on the newly learnt features and original word features are discarded in our approach; (c) sentiment information from preselected opinionated words are also utilized in our method.

With the success of topic model, researchers also attempted to use topics as pivot features. Liu and Zhao (2009) observed that customers often use different words to comment on the similar topics in the different domains, and therefore, these common topics can be used as the bridge to link different domain-specific features. They proposed a topic model named Transfer-PLSA to extract the topic knowledge across different domains. Through these common topics, the features in the source domain were mapped to the target domain features, so that the domain-specific knowledge could be transferred across different domains. He et al. (2011) also proposed a similar method using Joint Sentiment-Topic (JST) model which incorporates word polarity priors through modifying the topic-word Dirichlet priors.

All work discussed so far used pivot features and their experimental results suggest that classification accuracies have been improved. However, pivot features have limitation. Ideally, pivot features, or domain-independent features, would act exactly the same way with respect to sentiment labels in both domains. However, it is hard to measure since we do not have labeled data in target domain and performance would largely depend on selection of pivot features. In order to break this limitation, latent space models were introduced for cross-domain sentiment classification. Titov (2011) used a Harmonium Model Smolensky (1986) with a single layer of binary latent variables to cluster features in both domains and ensure that at least some of the latent variables are predictive of the label on the source domain. Such model can be regarded as composed of two parts: a mapping from initial (normally, word-based) representation to a new shared distributed representation, and a classifier in this representation. They combined their model with the baseline *out-domain* model using the product-of-experts combination (Hinton 2002) for classification. Glorot et al. (2011) adopted deep learning, which learns to extract an abstract meaningful representation for each review in an unsupervised fashion. They used Stacked Denoising Auto-encoders (SDA) as the building blocks of the deep network and trained a classifier based on the output of the network. Unlike other research, they only relied on the newly learnt features and did not adopt original word features. Our work also uses latent space model for latent representation learning. The major differences are, we adopt Restricted Boltzmann Machine (RBM) for latent representation learning, and additionally, we perform sentiment classification over a hybrid representation combining both the latent representation and the sentiment features from preselected sentiment words.

There are also a number of works which explored domain adaptation under specific context and worth mentioning here. Peddinti and Chintalapoodi (2011) performed sentiment analysis of Twitter by adaptation data from Blippr and IMDB movie review. They proposed two iterative algorithms based on Expectation Maximization and Rocchio SVM for filtering out noisy data. The experimental results showed that their approach was quite effective with F-score up to 0.9. Mejova and Srinivasan (2012) studied the problem of sentiment analysis across media streams. The authors created dataset consist of data from blogs, reviews, and Twitter and concluded that models trained on some social media sources are generalizable to others and Twitter to be the best sources of training data. Since those work are restricted in a specific context, the approaches might not work in general cases.

2.2.3 Other Sentiment Analysis Tasks

Some other sentiment analysis tasks were also investigated in existing literature and worth mentioning in the context of this particular research. For example, Ding et al. (2008), Hu and Liu (2004) and Liu et al. (2005) studied the problem of feature-based sentiment analysis, which first discovers the targets on which opinions have been expressed in a sentence, and then determines whether the opinions are positive, negative or neutral Liu (2010). Jindal and Liu (2006), Li et al (2010) and Xu et al (2011) examined the problem of comparative opinion mining. Jindal and Liu (2008) explored the problem of opinion spam. Lastly, Pang and Lee (2008) provided a comprehensive review of work in sentiment analysis.

2.3 SOLUTION OVERVIEW

We are interested in determining text sentiment orientation when *in-domain* labeled data is unavailable. The major obstacle for simply borrowing labeled data from other domains is the word distribution discrepancies between domains. The domain that provides labeled data is often referred as source domain, while target domain is the domain on which we would like to perform sentiment classification. However, this obstacle can be overcome if we could map text in the source domains and the target domain into a common space where those discrepancies vanish, or reduce, to a great extent. *Latent space model*, e.g., Restricted Boltzmann Machine (RBM), could serve this purpose. The assumption we make is that the latent representations would be similar for texts with the same sentiment label, even though their word representations differ.

In addition to borrow labeled data from other domain, unsupervised learning methods, where labeled data are unneeded, can be applied. The unsupervised method relies on preselected opinionated words and underperforms the *in-domain* supervised methods (Turney 2002). However, our intuition is combination of preselected opinionated words along with cross domain latent representation would improve the accuracy of existing approaches.

Furthermore, the selection of source domain classification plays a significant role for cross-domain. However, it has been rarely mentioned in the literature. In this research, we propose two approaches: (1) Intelligent Single Source Domain (ISSD) method and (2) Multiple Source Domain (MSD) method. The former one refers to automatically select the most similar domain as the source domain while the latter one uses all domains.

At a high level, our method combines two sources of information: (a) sentiment information from other domains, referred to as source domains, and (b) sentiment information from a hand-picked opinionated word list. We first learn latent space representations for texts where inter-domain distribution variations disappear, or at least reduce to a great extent. Restricted Boltzmann Machine (RBM) is adopted for this purpose due to its recent prominent performance in text related tasks (Larochelle and Bengio 2008). Unlabeled data from source domains and target domain are required for representation learning but they are readily collectable. Next, we identify opinionated words and calculate positive ratio and negative ratio in each document taking advantage of a preselected opinionated word list. Finally, we combine the two features accounting for positive and negative proportions along with the newly learnt latent space representations and train classifiers over this hybrid feature space.

Our approach has several key characteristics that make it quite different from the existing cross domain classification approaches: (a) We only use unigrams while all previous work selected both unigrams and bigrams, also we lemmatize the words before feed them to our system. Pang et al (2002) suggest that unigram information turned out to be the most effective. The unigram features makes our approach more efficient in terms of performance, whereas the lemmatization reduces the sparseness in the data. (b) We use sentiment information from a preselected opinionated word list in addition to labeled data from source domains and construct hybrid feature representations for classification while nearly all of the existing works on cross-domain sentiment classification rely on out-domain labeled data alone. (c) Unlike most of existing work, we rely only on newly learnt features. (d) We adopt the Restricted Boltzmann Machine for latent representation learning and experimental results demonstrate its superiority.

2.4 SOLUTION DETAILS

In this section, we describe the architecture of our system, and the details of each component in the architecture. We will use the piece of text “iPhone has good reception and excellent display” as an example for illustrative purpose throughout the rest of this chapter.

2.4.1 System Architecture

The overall architecture of our approach is depicted in Figure 2.1. In the preprocessing step, we perform routine text processing procedures, including lemmatization and unigrams extraction. The domain selection refers to choose the appropriate domain as source domain. Feature construction aims to build the features for classification. It contains 3 components: (1) the latent features learning aims to learn latent representation; (2) the opinionated features expansion is responsible for building sentiment words features; (3) the hybrid features construction combines these two set of features. Lastly, we detect sentiment orientation using supervised machine learning methods. We describe each of these components in detail below.

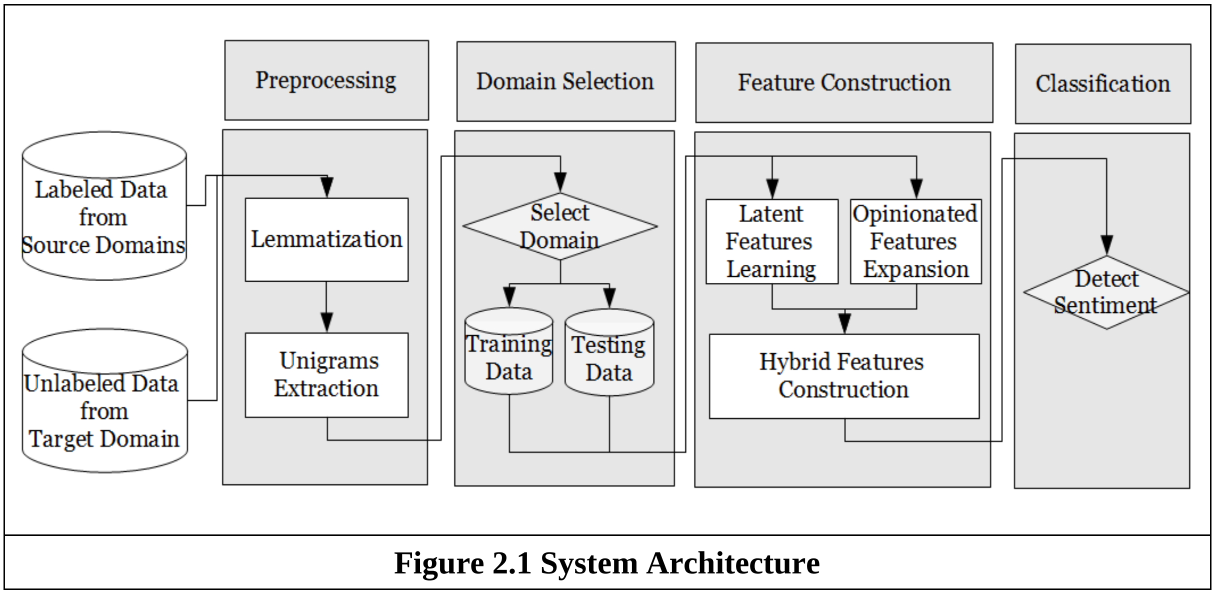

Figure 2.1 System Architecture

2.4.2 Preprocessing

This section introduces the text processing procedure before inputting the data into the system.

Lemmatization

Before feeding the text data into our system, we first carry out lemmatization on each document using Stanford Core Natural Language Processing (NLP) toolkit [4] on both labeled data from multiple source domains and test data from the target domain. Lemmatization, which transfers inflected forms to base form, or lemma, reduces the sparseness of the data and has been shown to be effective in text classification (Joachims 1998). For example, "runs", "ran" and "running" will be all converted into "run". Lemmatization is closely related to stemming. The difference is that stemming operates on a single word *without* knowledge of the context. For instance, the word "meeting" can either be a base form of a noun or an inflected form of a verb. Lemmatization will determine this based on the contextual Part-of-Speech (POS) information, and thus, it is more appropriate for our classification context.

Unigrams Extraction

In this work, we select only unigrams as training features, while all previous research considered both unigrams and bigrams. Experimental results of Pang et al (2002) suggest that unigram information turned out to be the most effective and none of the alternative features, e.g. bigrams, provides consistently better performance. With less features, our system can run more efficiently, especially for latent representation learning which is computationally expensive. We consider only the presence/absence of a word; the frequency of the word is not under consideration. The former achieves better results as

[4] http://nlp.stanford.edu/downloads/corenlp.shtml

shown in Pang et al (2002). Furthermore, stop words, such as “a”, “do”, “be”, are excluded since they are not helpful for our classification task.

Following the example in consideration, we will have “iPhone”, “good”, “reception”, “excellent” and “display” after this preprocessing step.

2.4.3 Source Domain Selection

Selection of source domains plays an important role in domain adaptation. In this study, we propose two approaches: (1) Intelligent Single Source Domain (ISSD) method and (2) Multiple Source Domain (MSD) method. The former one refers to automatically select the most similar domain as the source domain while the latter one uses data from all domains. So this step is only for the ISSD method, since data from all domains will be used for the MSD method. We will discuss which approach of using source domain is better in the evaluation section.

As we discussed before, the reduction of the accuracy is because of the discrepancy between source domain and target domain. So we believe that the classification would be higher if the discrepancy is less. Kullback–Leibler Divergence (KLD) (Kullback and Leibler 1951) is widely used to calculate the divergence between two probability distributions. It can be calculated as follows:

$$D_{KL}(S \,||\, T) = \sum_{w_i} S(w_i) \times \log \frac{S(w_i)}{T(w_i)} \qquad \text{Eq. 2.1}$$

where $S(w_i)$ is the probability of word $w_i$appearing in the source domain and $T(w_i)$ is the probability of word $w_i$appearing in the target domain. However, KL divergence is asymmetric and undefined if $T(w_i) = 0$. In order to overcome these limitations, we adopt the Jensen–Shannon Divergence (JSD) (Lin 1991) to measure the similarity between the source domain and the target domain. It is a symmetric and measures the KLD between S, T and the average of those two distributions:

$$JSD(S \,||\, T) = \frac{1}{2} D_{KL}(S \,||\, M) + \frac{1}{2} D_{KL}(T \,||\, M) \qquad \text{Eq. 2.2}$$

Where $M = \frac{1}{2}(S + T)$. The domain which has the lowest JSD with the target domain will be selected as the source domain.

2.4.4 Feature Construction

In this section, we elaborate the procedure of feature construction.

Latent Features Learning

Any joint probability model that uses vectors of latent variables to abstract away from hand-crafted features whose format is designed by human, e.g., bigrams, would work for our latent representation learning step. The assumption is, the texts with the same sentiment label would have similar abstract representations where cross-domain distribution variation disappears, or at least will be reduced to a great extent, even though their text representations differ. Through the training, different words with the same sentiment from different domain, like “compact” (electronic domain) and “realistic” (video game domain), would correspond to the same latent variable. Therefore, the sentimental information is “transferred” from source domain to target domain. By using the newly learned representation, the feature representation discrepancy between source and target domain is reduced which improves the classification performance.

In this research, we choose to use Restricted Boltzmann Machine (RBM) to learn latent and more abstract representations due to its recent prominent performance in text related task (Larochelle and Bengio 2008). RBM is an energy-based graphic model which associates a scalar energy to each configuration of the variables of interest and learning the parameters corresponds to modifying the energy function so that it has desired properties, e.g., we would like to have desirable configurations to have low energy. RBM consists of a layer of hidden units and a layer of visible units. A RBM with 3 hidden units and 4 visible units is shown in Figure 2.2.

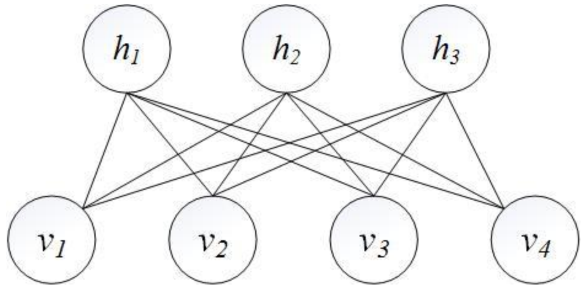

Figure 2.2 A RBM with 3 hidden units and 4 visible units

Suppose that a RBM models a distribution between n hidden units $\boldsymbol{h} = (h_1, h_2, \dots h_n)$ and d-dimension input visible units $\boldsymbol{v} = (v_1, v_2, \dots v_d)$. The energy function of the RBM is defined as:

$$E(\boldsymbol{v}, \boldsymbol{h}) = -\boldsymbol{h}^T \boldsymbol{W} \boldsymbol{v} - \boldsymbol{b}^T \boldsymbol{v} - \boldsymbol{c}^T \boldsymbol{h} \qquad \text{Eq. 2.3}$$

where $\boldsymbol{W}$ represents the weights connecting hidden and visible units, $\boldsymbol{b}$ and $\boldsymbol{c}$ are the offsets of the visible and hidden units respectively.

Because of the specific structure of RBM, visible and hidden units are conditionally independent given one-another. In addition, both hidden units $\boldsymbol{h}$ and visible units $\boldsymbol{v}$ are binary in our context. So we can write transition probability between visible layer and hidden layer as follows:

$$P(\boldsymbol{h} = 1|\boldsymbol{v}) = sigm(\boldsymbol{c} + \boldsymbol{W}\boldsymbol{v}) \qquad \text{Eq. 2.4}$$

$$P(\boldsymbol{v} = 1|\boldsymbol{h}) = sigm(\boldsymbol{b} + \boldsymbol{W}^T\boldsymbol{h}) \qquad \text{Eq. 2.5}$$

where $sigm$ is the sigmoid function defined as follows:

$$f(a) = \frac{1}{1 + e^{-a}} \qquad \text{Eq. 2.6}$$

The probability of a specific configuration is:

$$P(\boldsymbol{v}, \boldsymbol{h}) \propto e^{-E(\boldsymbol{v},\boldsymbol{h})} \qquad \text{Eq. 2.7}$$

which allows us to write:

$$P(\boldsymbol{v}) = \sum_{\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})} \qquad \text{Eq. 2.8}$$

RBM can be trained by minimizing the empirical negative log-likelihood of the training data and the cost function is:

$$c(\boldsymbol{v}) = -\log P(\boldsymbol{v}) \qquad \text{Eq. 2.9}$$

Stochastic gradient descent is properly applied in the training process. However, in this research, we use Contrastive Divergence which can train RBM much more efficiently (Carreira-Perpinan and Hinton 2005). RBM is trained in unsupervised manner, thus only unlabeled data are needed and they are readily collectable. Unlabeled data from both multiple source domains and the target domain are required. They are processed according to the procedures in the previous section before feeding into RBM training.

After learning the parameters, we convert the text representation of a document into a latent representation. Each visible variable represents a word with binary values, that is, “1” stands for presence and “0” otherwise. Using the learnt parameters and equation 2.2, we can calculate the probabilities of each hidden variable being “1”. Here we have two ways of constructing latent features. First, we can sample a value for each hidden variable given its probability and then take all hidden unit values as the feature vector to represent a specific document. Second, we can directly use the values of probabilities as latent representation. Either way will produce the same classification accuracy. In this study, we choose the second way. For instance, if we choose the size of latent representation to be 5, the previous example would be covert into the likes of (“0.24”, “0.79”, “0.41”, “0.94”, “0.31”).

Opinionated Features Expansion

Sentiment orientation can also be identified in an unsupervised manner. One simply example would be identifying orientation based on the ratio of the number of positive vs.

the number of negative words. If the ratio exceeds 1, the document might be positive; and vice versa.

We would like to combine this opinionated word feature with our latent space representation. Two features accounting for opinionated words in a document – (i) the ratio of the number of positive words vs. the number of the total opinionated words and (ii) the ratio of the number of negative words vs. the number of the total opinionated words. We use a list of positive and negative opinion words for calculation of these two ratios. The list is compiled over many years starting from 2004 by author of Liu (2010) and consists of approximately 6800 words[5]. Use of two ratios may seem a little duplicated since either value can be inferred by the other one. However, two-feature representation is necessary. Suppose we have only positive ratio feature. The following two occasions would have the same value "0": (a) no opinionated word exists; (b) all the opinionated words are negative. Clearly these two cases are different and need to be distinguished. However, if we use two features, this will not be a problem. The first case is represented as (0, 0) while the latter one is (0, 1). In addition, if number of positive equals that of negative words, this representation will be 0.5 for both positive and negative features.

There are, of course, more sophisticated uses of opinionated words in literature. We only use the simplest one here and it is enough for performance improvement as will be shown in the experiment. For our example, it has two positive words ("*good*" and "*excellent*")

[5] http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar

and no negative words, so the opinionated word feature is (“1”, “0”) where the former value is the positive ratio and the latter one corresponds to the negative ratio.

Hybrid Features Construction

In order to take the advantage of both representations, we combine the two sets of features, i.e., latent features and opinionated word features, and form the hybrid feature representations. Following our example, after this step, we will have (“0.24”, “0.79”, “0.41”, “0.94”, “0.31”, “1”, “0”) as the final representation.

2.4.5 Classification

At this stage we have hybrid representations for both training and testing data. The standard supervised machine learning methods can be applied easily. In this study, we select Support Vector Machine (SVM) (Press et al. 2007) for sentiment classification; however, other classifiers can also be applied here. We first use the multiple source domain labeled data to train the SVM model on the basis of this hybrid representation. After which, we use the hybrid representation of the target domain to classify the target documents as positive or negative sentiment.

2.5 EVALUATION

In this section, we first describe our dataset and evaluation metrics, and then discuss our experimental results.

2.5.1 Experimental Setting

The Multi-Domain Sentiment Dataset [6] is used in all existing work and we will also use this dataset for ease of comparison. The dataset is collected by authors of Blitzer et al (2007). The Multi-Domain Sentiment Dataset contains product reviews taken from Amazon.com for many product types (domains). Each domain has 1000 positive, 1000 negative reviews, and a number of unlabeled reviews - some domains (books and DVDs) have hundreds of thousands while others (musical instruments) have only a few hundreds. Each review consists of rating (0-5 stars), reviewer's name, reviewer's location, product name, review title, date, and the review text. Reviews with rating > 3 were labeled positive, those with rating < 3 were labeled negative, and the rest discarded because their polarity was ambiguous. In addition, a number of unlabeled reviews are also available for each domain.

Domain	Number of Reviews		
	Positive	Negative	Unlabeled
Books	1000	1000	4465
DVDs	1000	1000	3586
Electronics	1000	1000	5681
Kitchen	1000	1000	5945

Table 2.1 Data Statistics

[6] http://www.cs.jhu.edu/~mdredze/datasets/sentiment/

All existing cross-domain sentiment classification research selected four domains: books, DVDs, electronics and kitchen appliances. For ease of comparison, we will also evaluate our method over these four domains. The data statistics are listed in Table 2.1.

Similar to the previous research (Blitzer et al. 2007), we randomly select 200 positive reviews and 200 negative reviews as test data for each domain and the remaining 1600 labeled reviews in each domain are used as training data. All unlabeled data are used for latent representation learning. For computational reason, only top 5000 frequent unigrams in the dataset are selected as features for latent space representation learning.

Parameter	Value
Learning rate	{0.1, 0.01, 0.001}
Epochs	{10, 15, 20, 25, 30}
Hidden units	{5000}

Table 2.2 Parameter Range

Restricted Boltzmann Machine was implemented using Matlab[7]. In latent space model learning, we tried an extensive set of learning parameters and the following combination gave us the best results: hidden unites: 5000, learning rate 0.1, epochs: 30. Support Vector Machine (SVM) implemented in Weka (Hall et al. 2009) was selected as our classifier. When training SVMs, we chose the Radial Basis Function (RBF) kernel (Buhmann 2003) since we found that it consistently outperformed other counterparties in our classification context.

[7] http://www.mathworks.com/products/matlab/

2.5.2 Evaluation Metrics

We use two metrics to evaluate our method. The first one is accuracy which captures the percentage of all reviews that are classified correctly. It can be computed as follows:

$$Accuracy = \frac{number\ of\ reviews\ correctly\ classified}{number\ of\ reviews\ in\ the\ test\ set} \qquad \text{Eq. 2.10}$$

Accuracy is a widely used metric in literature and offers us a direct performance of the classification. However, it incorporates the contribution of the classifier as well. In order to eliminate the effect of the classifier in the evaluation and assess the transfer efficiency more precisely, we adopt transfer loss which equals the reduction of accuracy compared with *in-domain* classification. This is quite necessary when we compare cross-domain sentiment classification methods using different classifiers. Let $e(S,T)$ be the error obtained by a method trained on the source domain S, or a combination of multiple source domains, and tested on the target domain T and $e(T,T)$ be the error of a method both trained and tested on target domain T using the same classifier, i.e., the *in-domain* method. Transfer loss can be calculated as follows:

$$L(S,T) = e(S,T) - e(T,T) \qquad \text{Eq. 2.11}$$

Transfer loss has been used in previous work (Blitzer et al. 2007, Glorot et al. 2011) and a lower value signifies a better performance.

2.5.3 Single Domain Method

This section presents the experimental results of cross-domain sentiment classification using a single source domain and validate our statement that use of similar domain as source domain would offer better results.

Domain Similarity

Each domain is represented by a 5000-dimension vector and each dimension is valued by the probability of the corresponding word appearing in the domain. We calculated the probabilities based on the dataset we used in the experiment. If a certain word does not appear in the domain, its probability is set to be 0. The Jensen–Shannon Divergences between each pair of domains are then calculated and presented in the following table. A lower value indicates less divergent, that is, more similar.

	Books	DVD	Electronics	Kitchen
Books		0.029730	0.361194	0.419062
DVD			0.196915	0.244196
Electronics				0.003937

Table 2.3 Domain Similarity

All values are in 10^{-4}

From the above table, we can see that Electronics and Kitchen are quite similar since the divergence is quite small. According to the results, we would select DVD as the source domain for Book and vice versa; and choose Kitchen as the source domain for Electronics and vice versa.

Accuracy

In the current research, there are three kinds of features: (1) unigrams, (2) latent (RBM), and (3) opinionated words ratios. The purpose of the current research is to propose a hybrid method for cross-domain sentiment classification that combines latent features and lexicon features. We can see the effectiveness of the latent features by comparing results of unigrams (1) and latent features (2) and show the effectiveness of the opinionated words features (3) by comparing results of latent features (2) and hybrid features (2+3). Thus, we only show the results for models trained on (1), (2), and (2)+(3) since we believe it is sufficient to achieve our research purpose.

Classification accuracies using only single source domain are presented in Table 2.4. The values of classification accuracy using the training data from the domain selected in the last section, i.e., the ISSD method, are in bold. From the table 2.4, we can see that, in average, hybrid method outperforms RBM method and RBM method is superior to the unigrams method. These results demonstrate the effectiveness of our new representation.

Source / Target	In-Domain	Unigrams				RBM				Hybrid			
		B	D	E	K	B	D	E	K	B	D	E	K
B	83.00		77.25	69.00	70.00		80.50	73.00	73.50		81.75	76.75	75.25
D	81.50	74.25		70.50	73.00	77.25		75.25	77.25	78.75		79.50	81.00
E	81.75	72.75	76.00		78.00	70.00	74.50		83.25	72.25	76.50		83.75
K	87.25	74.75	76.25	85.00		80.50	79.75	87.00		81.50	81.50	88.50	
Average	83.38	74.73				77.65				79.75			

Table 2.4 Classification Accuracy using Single Source Domain

All values are in percentages
ISSD results are in bold
B: Books; D: DVD; E: Electronics; K: Kitchen

The values in the first column of the table are the results of *in-domain* method where both training and testing data are from the same domain. The results of this method are considered as the gold standard for comparison. In the previous section, we select source domain for each target domain and the corresponding results are better than their counterparties with only one exception (when in hybrid representation, using Kitchen as the source domain can provides better results than using Books for DVD domain). These results suggest that use of similar domain as source domain could provide better results and Jensen–Shannon Divergences can effectively measure the similarity.

We ran a series of t-tests to check if our latent space features are statistically more effective than those using word representations for the ISSD method. The p-values are as shown in Table 2.5.

Method	Unigram	RBM	in-domain
RBM	0.0079***		
Hybrid	0.0011***	0.0076***	0.8756+
in-domain	0.0000***	0.3551+	

Table 2.5 P-values of Accuracy Significant Test for ISSD method

* p<0.1; ** p<0.05; *** p<0.01
+ two-side test conducted

From the above table we can see that the RBM method statistically outperformed unigram method at 0.01 level and Hybrid method is significantly better than RBM method and unigram method at 0.01 level. In addition, Hybrid and RBM are not significantly different with the in-domain method, indicating that those two methods are as good as the in-domain method.

Transfer Loss

Transfer losses of single source domain method are reported in Table 2.6. We follow the same structure as Table 5 and the transfer losses of ISSD method are in bold. The average transfer losses for unigrams, RBM and hybrid are 8.65, 5.73 and 3.62 respectively, which indicates that our representation learning could effectively reduce the reduction of accuracy. Transfer loss is less when domain with less divergence is used as source domain. One exception is using Kitchen as the source domain can provides better results than using Books for DVD domain for hybrid representation. The results further confirm our statement that use of similar domain as source domain would provide better results.

Source / Target	Unigrams				RBM				Hybrid			
	B	D	E	K	B	D	E	K	B	D	E	K
Books		5.75	14.00	13.00		2.50	10	9.5		1.25	6.25	7.75
DVD	7.25		11.00	8.5	4.25		6.25	4.25	2.75		2	0.5
Electronics	9.00	5.75		3.75	11.75	7.25		-1.5	9.5	5.25		-2
Kitchen	12.5	11.00	2.25		6.75	7.5	0.25		5.75	5.75	-1.25	
Average	8.65				5.73				3.62			

Table 2.6 Transfer Loss using Single Source Domain

All values are in percentages
ISSD results are in bold
B: Books; D: DVD; E: Electronics; K: Kitchen

2.5.4 Multiple Domains Method

This section presents the experimental results of cross-domain sentiment classification using a multiple source domains and compares Intelligent Single Source Domain method and Multiple Source Domains method.

Accuracy

Classification accuracies for various methods are presented in Table 2.7. Each row corresponds to results that one of the four domains serves as target domain. For instance, the first row presents results where Books is the target domain. All values in the table are in percentages.

The first column of the table shows results of method using opinionated words are used as the sole source. It classified the review as positive if number of positive words surpass number of negative words and negative otherwise. When these two numbers equal, we set it as positive. The accuracies range from 70.35% to 76.85% with an average of 73.50%.

Method / Target Doman	Opinionated Words	Intelligent Single Source Domain (ISSD)			Multiple Source Domains (MSD)		
		Unigrams	RBM	Hybrid	Unigrams	RBM	Hybrid
Books	70.35	77.25	80.50	81.75	75.25	82.00	84.25
DVD	73.75	74.25	77.25	78.75	77.75	83.50	84.50
Electronics	73.05	78.00	83.25	83.75	81.00	82.75	84.25
Kitchen	76.85	85.00	87.00	88.50	82.75	86.25	87.75
Average	73.50	78.63	82.00	83.75	79.19	83.69	85.19

Table 2.7 Classification Accuracy

All values are in percentages

The middle part of the table corresponds to the Intelligent Single Source Domain (ISSD) method, that is, intelligently select the domain which is most similar with the target domain as the source domain. As we can see from the table, classification accuracy ranges from 74.25% to 85% with an average of 78.63% when unigrams are used as features. The average accuracy goes up to 82% when latent features are used and further increases to 83.75% when the two opinionated word features are included.

Results of multiple source domain method are presented in the right part of the table. As we can see that the average accuracy of multiple source domain method is higher than that of intelligent single source domain method whatever feature is used. We postulate that the reason might be: (a) we can collect more data and a larger number of training instances would benefit classification; (b) word distributions in different domains vary and combination of multiple source domains will increase the probability that words in test set behave less discordantly with respect to those in the training set.

Classification accuracy ranges from 75.25% to 82.75% with an average of 79.19% when unigrams are used as features. When using latent representations learnt by RBM, the classification accuracy raise to 83.69% in average. In addition, it outperforms the *in-domain* method in DVD domain and Electronics domain. This conclusively demonstrates the effectiveness of our latent representation learning. Finally, we train our classifiers over the hybrid representations, which combine the latent representations, and opinionated words features. The accuracy further steps up to 85.19% in average and ranges from 84.25% to 87.75%. It produces better results than the *in-domain* method in all the four domains.

One interesting point is that MSD is inferior to ISSD when "Kitchen" is treated as the target domain, no matter what set of features are used. We postulate the reason might be that the source domain of ISSD, “Electronics”, is quite similar with the “Kitchen” (their JSD is quite close to 0 as reported in Table 3) and thus, ISSD provides good out-domain results. As we can see from Table 2.6, the transfer loss using unigrams is only 2.25, indicating that the out-domain result is close to the in-domain result. When we use the MSD, the “Books” domain, which is quite different from the “Kitchen” domain, is added for training and the accuracy is reduced. This result indicates that when we have a source domain which is quite similar with the target domain, it is better to use that domain as the sole source domain instead of use multiple source domain data simultaneously. From the series of results with “Kitchen” as the target domain, we can see that our latent feature learning effectively reduces the discrepancy between source domain and target domain. From Table 2.7 we could see that when “Kitchen” is the target domain, the MSD is 2.25% lower than the ISSD for unigrams representation; whereas, when the RBM and Hybrid representations are used, the MSD is only 0.75% lower than ISSD. This further proves the effectiveness of our latent feature learning.

We also ran a series of t-tests to check if our latent space learning results are statistically better than those using word representations for the MSD method. For example, we want to know whether the superior of RBM method over Unigram method is statistically significant. We calculated the increase of accuracy for each domain and then ran a one-tail t-test to see whether the difference is statistically greater than 0. The p values are reported in Table 2.8.

Method	Unigram	RBM	in-domain
RBM	0.0073***		
Hybrid	0.0080***	0.0045***	0.0250**
in-domain	0.0310**	0.7610^{+}	

Table 2.8 P-values of Accuracy Significant Test for MSD method

* p<0.1; ** p<0.05; *** p<0.01
$^{+}$ Two-tail test result is reported

From the results in Table 2.8 we can conclude that RBM and hybrid methods are statistically better than multiple sources with unigram representation method at a 0.01 level. The hybrid method is statistically superior to the RBM method at a 0.01 level. The results in Table 3 suggest that our hybrid approach of combining the latent space learning and opinionated word features are effective.

In addition, the *in-domain* method is statistically better than Multiple Sources method in a 0.05 level. The one-tail t-test results for the *in-domain* method over RBM are not statistically significant, indicating that the *in-domain* method is not statistically better. So we report the two-tail t-test results to see if there is any difference between the *in-domain* method and the RBM method statistically, i.e., if the accuracy difference is equal to zero. Results in the table show that the both differences are not statistically significant, indicating that the RBM method is statistically as good as *in-domain* method. Furthermore, the results suggest that our hybrid method statistically outperforms the *in-domain* method at a 0.05 level.

Besides checking whether our new features are statistically effective, we also would like to know whether the superior of multiple sources domain method to the intelligent single source domain method is statistically significant. Again, we run a one-tail t-test and the p value is 0.059 which indicates that the multiple sources domain method is statistically better than the intelligent single source domain method at a 0.1 level.

We also report the average accuracies of the RBM and Hybrid multiple source domains methods under different parameter setting are shown in Figure 2.3. From the figure we can see that the curves of hybrid methods are always above the curves of RBM methods with the same learning rate. In addition, the curves are upwards-sloping with few exceptions, that is, accuracies typically go up as the epochs increase. However, the slop is decreasing gradually. For example, in the curve of hybrid method with learning rate 0.1, the line between epoch 25 and 30 is nearly flat. For the sake of space, we do not report graphs for single source domain method.

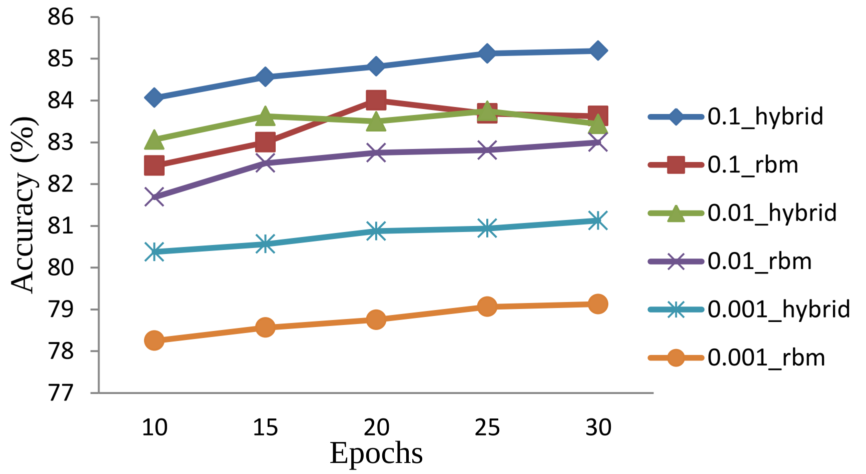

Figure 2.3 Accuracy Curve

Transfer loss

Next, we report the transfer loss, which captures the reduction of accuracy due to use of *out-domain* source, to assess the transfer efficiency. The results are shown in Table 9. We follow the same structure as Table 2.7 where the first column presents results of opinionated words method, the middle left part shows results of single source domain method and the right part illustrates accuracies of multiple source domain approaches.

As we can see from Table 2.9, the transfer loss averaged 9.88 with range from 7.75 to 12.65 when opinionated words are used as the sole source.

The transfer losses reduce dramatically when we use the most similar domain as source domain. Average transfer losses are 4.75, 1.38 and -0.38 for unigrams, RBM and hybrid features respectively. When using latent features learnt by RBM, the transfer loss for Electronics domain is below 0 which indicates that the accuracy is higher than that of *in-domain* method. Furthermore, there are two domains with negative transfer when the hybrid representation are used, i.e., Electronics and Kitchen.

Method Target Doman	Opinionated Words	Intelligent Single Source Domain (ISSD)			Multiple Source Domains (MSD)		
		Unigrams	RBM	Hybrid	Unigrams	RBM	Hybrid
Books	12.65	5.75	2.5	1.25	7.75	1.00	-1.25
DVD	7.75	7.25	4.25	2.75	3.75	-2.00	-3.00
Electronics	8.70	3.75	-1.5	-2	0.75	-1.00	-2.50
Kitchen	10.40	2.25	0.25	-1.25	4.50	1.00	-0.50
Average	9.88	4.75	1.38	0.19	4.19	-0.25	-1.81

Table 2.9 Transfer Loss

Notes: All values are in percentages

When multiple source domains are used, the average transfer losses further reduce. The transfer loss is 4.19 for unigrams representation. When we use latent representations learnt by RBM, the average transfer loss drops significantly to -0.25 with values of two domains being below 0. Furthermore, the average transfer loss reduces to -1.81 when the hybrid representations are adopted and values of all four domains are lower than 0. A value of average transfer loss less than zero suggests that the overall performance is even better than the in-domain method.

We do not report the significant test for transfer loss as it would fall in the same range as in Table 2.7.

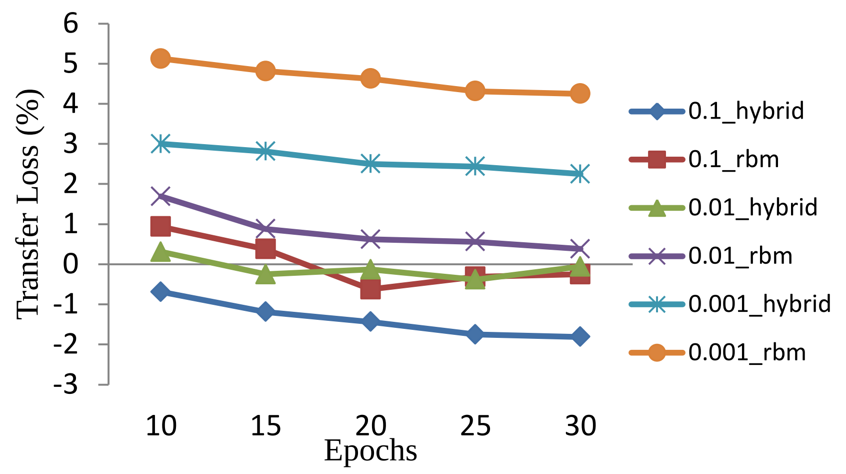

Figure 2.4 Transfer Loss Curve

We also report the average transfer loss for different sets of parameters for multiple source domains method in Figure 2.4. The figure suggests that average transfer losses tend to decrease as the epochs increases with several exceptions. However, the improvement is relatively small, which is around 1% for each curve from epoch 10 to 30. Furthermore, under the same learning rate, curve of hybrid method always lies below that of RBM method. For the sake of space, we do not report graphs for single source domain method.

It is also interesting to compare our work with previous ones, where the same dataset has been used. From the previously reported results, we calculate the average transfer loss for the following previous research: Blitzer et al. (2007); Pan et al. (2010); He et al. (2011); Bollegala et al. (2011); Titov (2011); and Glorot et al. (2011). The results of previous method as well as our Intelligent Single Source Domain (ISSD) and Multiple Source Domain (MSD) method using hybrid features are shown in Figure 2.5. From the figure we could see that both of our methods outperform all compared methods. The results conclusively demonstrate the superiority of our methods over all existing work.

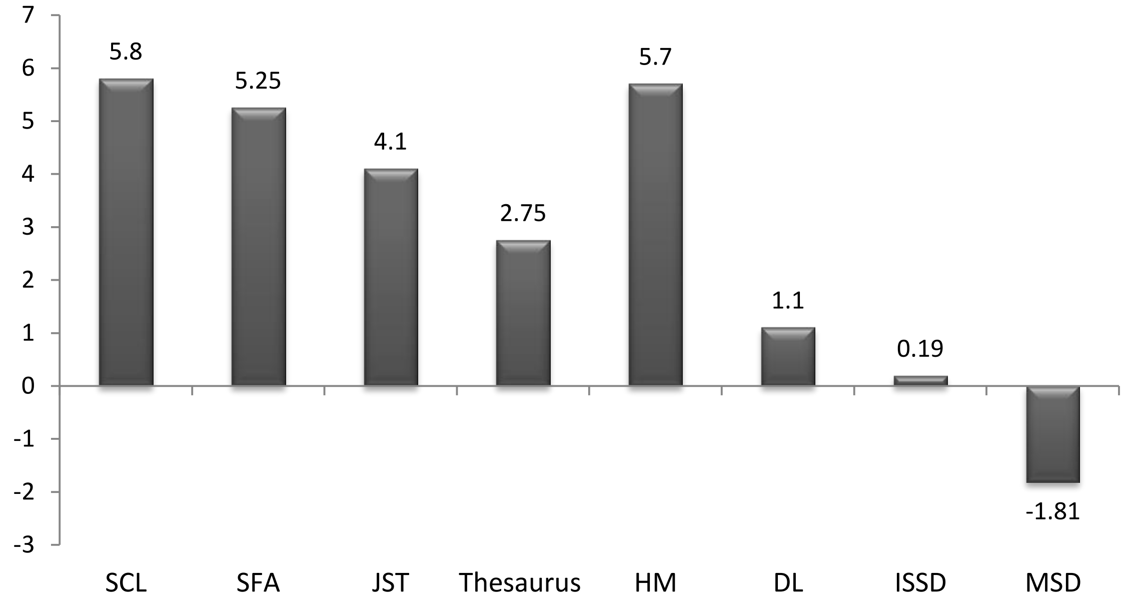

Figure 2.5 Transfer Loss across Methods

2.6 CONTRIBUTIONS AND LIMITATIONS

Our work has the following major contributions: (1) we utilized labeled data from source domain and opinionated words. To our best knowledge, this research represents the first attempt to combine sentiment information from source domain labeled data and hand-picked opinionated words together for the cross-domain sentiment classification task; (2) we propose to use Jensen–Shannon Divergences to measure domain similarity and use similar domains as source domains; (3) our experimental results show that our methods, both Intelligent Single Source Domain (ISSD) and Multiple Source Domain (MSD), statistically outperform the existing work addressing the same problem.

There are several limitations in this study. Firstly, we only tested our method over 4 domains. An extensive evaluation over a larger dataset could further validate the effectiveness of our method. Secondly, the opinionated word list we used in this study only contains correctly spelled words. However, in tweets or Facebook comments, words are usually spelled incorrectly or irregularly. In addition, some emoticons are also used. In this case, an extended word list is needed to successfully handle these situations.

2.7 CONCLUSION AND FUTURE DIRECTIONS

In this study, we proposed a novel framework for cross-domain sentiment classification using latent representation and opinionated word features. The experimental results suggest that our proposed methods statistically outperform the existing work in the literature.

In future, we plan to conduct a more thorough evaluation over a larger scale of data with more domains. In addition, the simplest way of utilizing hand-picked opinionated words is used in our hybrid method. There are a number of much more sophisticated methods available in the literature. We are keen to see if an advanced method would further increase the accuracy.

CHAPTER 3 STUDY II: LDA-BASED INDUSTRY CLASSIFICATION

3.1 INTRODUCTION

Industry analysis, which studies a specific branch of manufacturing, service, or trade, is widely used in financial analysis (Davis and Duhaime 1992, Kahle and Walkling 1996). Such analysis is used by various groups of people: (a) asset managers need industry analysis to investigate the target company’s competitive environment and growth opportunities, after which they could perform stock selection and valuation (Bhojraj and Lee 2002); (b) credit analysts need industry analysis to assess the target company’s financial status through the comparison of industry averages, after which they could rate the company; (c) investors need industry analysis to study the target industry’s competitiveness, profitability and growth, after which they could make investment decisions; (d) researchers need industry analysis to identify the industry that the target company belongs to, after which they could design appropriate control groups for their studies (Lee et al. 2012).

Before we can perform industry analysis, one crucial step to take is to define industry boundaries effectively and accurately. In other words, we need to assign firms into appropriate industries on the basis of commonalities before any further analysis could be conducted. Otherwise, further industry analysis could become impossible, or at least misleading. Appropriateness and accuracy of industry classification is the premise of an effective and valuable industry analysis.

There exist a number of *Industry Classification* schemes such as the Standard Industrial Classification (SIC)[8] and the North American Industry Classification System (NAICS)[9]. However, these schemes have two major limitations. Firstly, they are all static and assume that the industry structure is stable (Hoberg and Phillips 2013). However, firms often introduce new products, improve old products and discontinue outdated products, and thus enter and exit various industries. In addition, due to technology innovation, some industries change or even fade out, and new industries appear. Since firms and markets evolve with the passage of time, an effective industry classification approach should be able to capture the dynamic aspect of industries. Researchers have started to address this problem through annually updated documents such as financial statements (Chong and Zhu 2012).

Secondly, these schemes assume binary relationship – two firms either in the same industry or from different industries – and do not measure the degree of similarity. This is particularly important when identifying rivals for a target firm. Similarities between firms within the same industry vary a lot and we would like to select the most similar firms as rivals. We believe that an effective industry classification approach should not only be able to identify industries, but also can measure differences within industry, that is, to capture the within industry heterogeneity. In order to overcome this limitation, researchers have started a line of work referred as *Peer Firms Identification*, which aims to identify the most similar firms of the target firm. Data such as input-output (IO) tables

[8]http://www.census.gov/epcd/www/sic.html [Accessed May 1, 2013]

[9] http://www.census.gov/eos/www/naics/ [Accessed May 1, 2013]

(Fan and Lang 2000), 10-K forms (Hoberg and Phillips 2013) and EDGAR[10] search traffics (Lee et al. 2012) were used for this purpose. However, as will be discussed in details in the next section, these work suffer from weaknesses such as failure to consider firms' business scales and inaccurate classification.

In this study, we propose an industry classification methodology on the basis of business commonalities using the topic features learned by the Latent Dirichlet Allocation (LDA) (Blei et al. 2003) from firms' business descriptions. Unlike most of the existing work, which address either industry classification or peer firms identification, we address them concurrently since we believe they are essentially the same. In industry classification, firms are grouped with the industry center as the centroid and we refer to this as *industry-centric industry classification (ICIC)*; In peer firms identification, firms are grouped with the target firm as the centroid and this is referred as *firm-centric industry classification (FCIC)* in the current work. ICIC is applicable when there is no target firm and we just want to have an overview of the market and industries, while FCIC is useful when we have a target firm to study or compare. We represent each firm's business genre by the topic features learned from firms' business descriptions, over which industries are classified and peers are identified. ICIC is achieved through a clustering algorithm and FCIC is accomplished according to the business divergence between firms.

The rest of the chapter is organized as follows: we first review related work in literature. Then we provide the intuition of our method followed by an elaboration of our proposed

[10] http://www.sec.gov/edgar.shtml [Accessed May 1, 2013]

method. Whereafter, we present the results of our preliminary evaluation. Finally, we discuss possibilities for future work.

3.2 RELATED WORK

In this section, we first review related work in industry classification and then present existing research in peer identification.

3.2.1 Industry Classification

There are a number of industry classification schemes used by practitioners and researchers. Bhojraj et al.(2003) offered a comparison of several major industry classification schemes in a variety of applications in accounting, economics and finance, including the Standard Industrial Classification (SIC), the North American Industry Classification System (NAICS) and the Global Industry Classification Standard (GICS)[11]. We briefly introduce them below.

SIC was established in the United States in 1937 by the Central Statistical Board and classifies industries with a set of four-digit codes. Since the SIC is relatively obsolete, governmental agencies from the U.S., Canada and Mexico jointly developed the NAICS to replace SIC. Though the NAICS has largely replaced the SIC, certain government agencies, such as the U.S. Securities and Exchange Commission (SEC), are still using the SIC codes. Both SIC and NAICS are developed by governmental agencies, which may have little bearing on how investors actually perceive firm similarities (Bhojraj et al.

[11] http://www.msci.com/products/indices/sector/gics/ [Accessed May 1, 2013]

2003). The GICS, on the contrary, is a collaboration of Standard & Poor's and Morgan Stanley Capital International. It is based on the judgment of a team of financial analysts who read through regulatory filings to determine which firms are financially comparable and has been shown to outperform all other schemes in explaining stock return co-movements (Bhojraj et al. 2003). In the current research, we also rely on the regulatory filings. However, instead of reading them manually, we adopt text analytics techniques to automate the process.

Though quite a number of classification schemes are proposed, they all have the same limitation - they are static and assume that the industry structure is stable. Thus, they cannot capture the dynamic aspect of the industry. Researchers have started to address this problem by using annually updated regulatory filings. Chong and Zhu (2012) attempted industry classification in light of XBRL based financial information collected from the EDGAR. They modeled firms and the GAAP Taxonomy elements used by firms as a bipartite graph and applied a spectral co-clustering approach that simultaneously classified firms and financial statement elements over the network.

3.2.2 Peer Firm Identification

Industry classification as we discussed above has one major limitation: it can only present a binary relationship - two firms are either in the same industry or from different industries. In other words, industry classification does not distinguish firms in the same industry. In order to address this limitation, researchers have started their efforts to measure the degree of relatedness between firms.

Fan and Lang (2000) employed commodity flow data from input-output (IO) tables to measure the relatedness based on whether firms share the same inputs and outputs. Their results suggested that the new IO-based measures outperformed traditional measures based on SIC codes. One shortcoming of this method is the necessity for well-specified production processes, which are not available for industries such as software. Bhojraj and Lee (2002) developed “warranted multiples” - the future enterprise-value-to-sales and price-to-book ratio - for each firm with guidance from valuation theory and identified peers as those having the closest warranted multiples. Their experimental results showed the superiority of their proposed method over methods on the basis of other techniques such as industry and size matches.

Ramnathr (2002) defined the peer firms based on analysts’ choice of firm coverage. Firms that are followed by at least five analysts in common are categorized as peers. The intuition is that the brokerage house would assign similar firms for coverage to one analyst for the purpose of minimizing an analyst’s information acquisition cost. Franco, Hope, and Larocque (2013) improved this approach by using hand-collected data of peer choice by sell-side equity analysts in their research reports. They found that analysts are more likely to choose peer firms that are similar in size, leverage, etc, and select firms with high valuations.

Recently, there has been growing interest in using data from EDGAR of the U.S. Securities and Exchange Commission for industry classification and peer firm identification. Lee et al. (2012) used the Internet traffic patterns from the EDGAR website and an association rules based technique to identify peers. Their intuition is that firms appearing in chronologically adjacent searches by the same individual are fundamentally similar. The experimental results suggested that traffic-based approaches outperformed peer firms based on six-digit GICS groupings in explaining variations in base firms' stock returns. However, we found that some peers were clearly misidentified. For instance, Microsoft, a software corporation, was identified as a peer of Dow Chemical, a chemical corporation.

Hoberg and Phillips (2013) used nouns and proper nouns in 10-K forms' business description section for industry classification and peer firm identification. Specifically, they utilized those words to represent firms and adopted a text clustering algorithm to group firms into industries. In addition, they calculated the cosine similarity between the words of any two firms and selected peer firms using a simple minimum similarity threshold. They showed in the experiment that their text-based approach can explain firm characteristics better than SIC and NACIS. One major drawback of this work is that it failed to consider the business scale – peers should be of comparable business scale.

The current study also contributes to this strand of work. Though we also use 10-K forms downloaded from EDGAR, as will be discussed in the next section, our approach has several key characteristics that make it quite different from Hoberg and Phillips (2013).

3.3 SOLUTION OVERVIEW

We are interested in categorizing firms into industries based on their commonality of business. At a high level, our method consists of two steps: (1) deriving effective features from text data to represent firms' business; and (2) classifying firms into industries.

In order to represent firms' business, we utilize the "Item 1. Business" section of their 10-K form which is a required filling by the U.S. Securities and Exchange Commission (SEC) and is updated annually. It describes the business of the company, i.e., what the company does, what markets it operates in, etc. There are several advantages of using the Item 1 section for business representation. First, the section is updated annually, which enables our industry classification method to capture the evolvement of the firm's business. In addition, it is legally required that firms provide accurate information, which is the premise of high quality industry classification results. Thirdly, it is quite unlikely that a company can enter or exit one industry within one year, which assures the stability of our industry classification results. The use of 10-K forms restricts the current study to focus on public firms only; however, our proposed approach is generic enough to be applied in private firms, given that accurate business descriptions are provided.

We believe that each word in the "Business" section attributes to the corresponding firm's business activities. For instance, if a firm is involved in the oil business, words such as "fuel", "refinery", "crude" are very likely to appear in that firm's "Business" section. The Latent Dirichlet Allocation (LDA) (Blei et al. 2003) is adopted to learn those business activities, each of which is referred as a topic and defined as a multinomial distribution over words. Those topic features are formed as vectors to represent firms' business genre.

Topic features offer several benefits over word features used in Hoberg and Phillips (2013). (1) One major issue for text analysis is its high dimensionality which is essentially the number of unique words in a collection and causes the so-called "curse of dimensionality" (Archak et al. 2011; Korn et al. 2001). Use of topic features greatly reduces the data dimensionality and avoids the dimensionality curse. Though Hoberg and Phillips (2013) does not disclose the number of unique words, given the information in the paper and Heaps' Law (Heaps 1978), we can estimate that the number of distinct word features is about 44,000. However, we only use 50 topic features in our experiment. (2) Text data are typically quite sparse – while there are a huge number of potential words, number of words in a document is actually quite small. In Hoberg and Phillips (2013), the average number of words for each firm is only around 175. Use of topic features significantly reduces the data sparsity. According to our experiment, there is no zero-valued feature in firms' business representations. (3) Each topic is a multinomial distribution over words and we can use those probabilities to weigh words with respect to a certain topic. Top weighted words could be used to describe the business activities and furthermore, the industries, which offers natural interpretations of the resulting industries. (4) Topic features enable us to filter out irrelevant content very easily. For instance, in our experiment, we found that there is one topic corresponding to introduction of firms' management team. These kinds of topics should be excluded since they are not related to the firms' business. However, Hoberg and Phillips (2013) include nouns and proper nouns in those sections into firms' business representation, which inevitably brings noise and jeopardizes the accuracy of their approach.

After the business representations are constructed, we then classify firms into industries. Two types of industry classification are proposed: firm-centric industry classification (FCIC) and industry-centric industry classification (ICIC). FCIC is useful when there is a target firm to study or compare, and ICIC is applicable when there is no focused firm and we just want to have an overview of the industry and the market. FCIC is performed according to two criteria: business genre and scale. We believe that peer firms must have comparable business size. For instance, we have two firms, Microsoft and Tiger Logic, both of which design, develop and sell software products to customers. Although they are engaged in the same business, they are not peers and comparing them is meaningless since their business scale vary too much – Microsoft has a market capitalization of 276 billion USD while Tiger Logic only has 48 million USD. ICIC is accomplished through a clustering algorithm. These two methods might have some overlap. Top peers identified by FCIC are very likely to be in the same industry produced by ICIC. However, they are not the same and one is not a subset of the other. When a company is involved in multiple industries, its peers might be only similar in one industry.

Our approach has several key characteristics that make it quite different from Hoberg and Phillips (2013) which also use 10-K forms: (a) we use topic features to represent firms' business. As we have discussed, this offers a number of advantages over the word features used in Hoberg and Phillips (2013). (b) We consider the business scale in addition to the business activities, which is the only criterion considered in Hoberg and Phillips (2013). As we discussed, business scale is an indispensable criterion for peer identification. (c) We use completely different methods for industry classification and peer identification.

3.4 SOLUTION DETAILS

In this section, we describe the architecture of our system, and the details of each component in the architecture. We will use the piece of text from Google's Item 1 section of 10-K form “Our business is primarily focused around the following key areas: search, advertising, operating systems and platforms, enterprise and hardware products” as an example for illustrative purpose throughout the rest of this study.

3.4.1 Architecture

The system architecture of our approach is depicted in Figure 3.1. The representation construction aims to construct features to represent firms’ business genre effectively. It first performs routine text processing and then learns the topic features. After the representations are constructed, firms are classified either in a firm-centric or industry-centric way. We describe each component in Figure 3.1 in detail below.

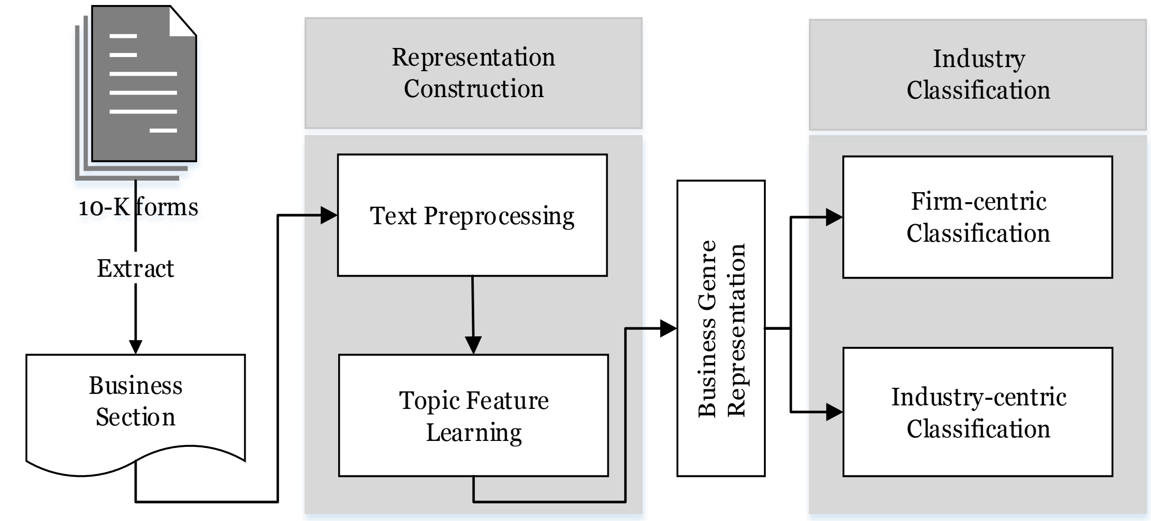
10-K forms
Extract
Business Section
Representation Construction
Text Preprocessing
Topic Feature Learning
Business Genre Representation
Industry Classification
Firm-centric Classification
Industry-centric Classification

Figure 3.1 System Architecture

3.4.2 Representation Construction

Text Preprocessing

Before feeding the text data into the LDA for topic feature learning, we first carry out lemmatization on each piece of “Business” section using the Stanford Core Natural Language Processing (NLP) toolkit (Stanford NLP Group 2013). Lemmatization, which transfers inflected forms to base form, or lemma, reduces the sparseness of the data and has been shown to be effective in text related tasks (Joachims 1998). For instance, “says”, “said” and “saying” will be all converted into “say”. Lemmatization is closely related to stemming. The difference is that stemming operates on a single word without knowledge of the context. For example, the word “meeting” can either be a base form of a noun or an inflected form of a verb. However, lemmatization will determine this based on the contextual Part-of-Speech (POS) information, and thus, we believe it is more appropriate for our current context.

We also remove words that appear very frequent. This includes those typical stop words such as “a”, “do”, “be”, which are not semantically informative. In addition, we also exclude common words that are used by more than 50% of all firms. We believe those common words carry little industry-specific information. After this step, we acquire a set of words that describe the business of a particular firm for topic feature construction.

Following the example in consideration, we would have “focus”, “search”, “advertise”, “operate”, “system”, “platform”, “enterprise” and “hardware” after this preprocessing step.

Topic Feature Learning

The Latent Dirichlet Allocation (LDA) (Blei et al. 2003) is selected for topic feature extraction. LDA is a three-level hierarchical Bayesian model, which models a document as a finite mixture over a set of underlying topics. A graphical representation of LDA adopted from Blei et al. (2003) is presented in figure 3.2. W represents a specific word in a document; Z is the topic that generates W; α, β are the parameters of the Dirichlet prior on the per-document topic distribution and per-topic word distribution; θ is the topic distribution for documents and φ is the word distribution for topics; K is the number of topics, N is the number of words in a document and M is the number of documents in a collection.

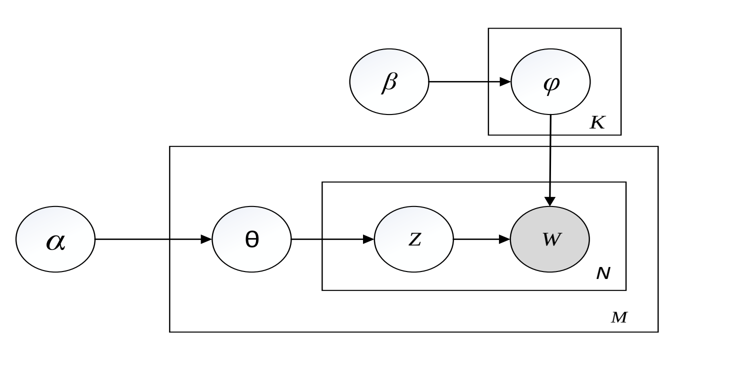

Figure 3.2 Plate Notation of a smoothed LDA

LDA posits that each word in a document is generated by a topic and each document is a mixture of a finite number of topics. Each topic is represented as a multinomial distribution over words. There are a number of outputs from the LDA. In the current research, we will use two of them: (1) p($topic_k$| $document_m$) – the probability of $topic_k$ occurring given $document_m$; (2) p($word_w$ | $topic_k$) – the probability of $topic_k$ generating $word_w$. The first set of probabilities is used as topic features to represent firms’ business genre and the second set of probabilities is used for industry description. In order to train the LDA, we need to specify the number of topics – k. In this study, we choose topic numbers using the perplexity scores as well as manually interpreting resulting topics. Typically, the perplexity scores decrease as topic number increases. We choose the number of topics that produces interpretable topics after the reduction of perplexity starts to decrease. The details of the LDA, including model estimation and inference, are beyond the scope of this study and interested readers can refer to Blei et al. (2003).

We believe that the creation of each word in the “Business” section of a firm’s 10-K form is attributable to the firm’s business genre. For instance, an oil company would tend to use words such as “fuel”, “refinery”, “crude”, etc; however, “broadband”, “wireless” and “subscriber” are more likely to appear in the business description of a firm in the telecommunication industry. Each business genre can be viewed as a topic, which generates words that constitute the business section according to a certain distribution. Given the words appeared, we can infer the underlying topics, or business genres, that generates the words. The probabilities p($topic_k$| $document_m$) are then used as topic feature to represent firms’ business. For instance, if we choose the number of topics to be 5, Google’s business genre representation learned from the previous piece of text would be the likes of (“0.0079”, “0.0031”, “0.0143”, “0.2523”, “0.0015”).

3.4.2 Industry Classification

Firm-centric Industry Classification

Firm-centric Industry Classification (FCIC) aims to find comparable firms for the target firm. In this type of classification, each firm has its own set of peer firms which constitute an industry. This is useful when we have a target firm to study or compare. We identify comparable firms considering two criteria. First, peer firms should be engaged in similar business activities. In addition, peer firms should have comparable business scales. As we have discussed previously, if two firms vary too much in business scale, even though in the same business genre, they are hardly peers.

We measure the similarity between two firms in terms of the Kullback–Leibler Divergence (KLD) (Kullback and Leibler 1951) of two firms' business genre representation we constructed in the previous section. KLD is widely used to calculate the divergence between two probability distributions and is well-suited for the current problem. The KLD of firm F_2 from firm F_1 can be calculated as follows:

$$D_{KL}(F_1 \,||\, F_2) = \sum_{T_i} F_1(T_i) \times \log \frac{F_1(T_i)}{F_2(T_i)} \qquad \text{Eq. 3.1}$$

where $F_1(T_i)$ and $F_2(T_i)$ are topic features we learned in the previous section. From the equation, we can see that KLD is asymmetrical and thus it not a distant metric. To overcome this, we use the following equation to compute the business genre divergence:

$$D_{genre}(F_1, F_2) = D_{KL}(F_1 \,||\, F_2) + D_{KL}(F_2 \,||\, F_1) \qquad \text{Eq. 3.2}$$

where $D_{KL}(F_1 \,||\, F_2)$ is the KLD of firm F_2 from firm F_1, and $D_{KL}(F_2 \,||\, F_1)$ is the KLD of firm F_1 from firm F_2. $D(F_1, F_2)$ measures the divergence between business genres of two firms, and therefore, the smaller the value, the more similar the two firms are. As we discussed previously, peer firms should have comparable business scale. We measure firms' business scales using their market capitalization, which is the total value of the issued shares of a publicly traded company. It can be calculated as follows:

$$marketcap = shareprice \times numberofoutstandingshares \qquad \text{Eq. 3.3}$$

The ratio of market cap of two firms is used to measure the business scale comparability. Specifically, we use the following equation:

$$D_{scale}(F_1, F_2) = \log_{10} \max(\frac{\text{market cap}_{F_1}}{\text{market cap}_{F_2}}, \frac{\text{market cap}_{F_2}}{\text{market cap}_{F_1}}) \qquad \text{Eq. 3.4}$$

$D_{scale}(F_1, F_2)$ would be close to zero if they have similar business scale. Finally, the business divergence of two firms is measured using the following equation:

$$D_{business}(F_1, F_2) = D_{genre}(F_1, F_2) + D_{scale}(F_1, F_2) \qquad \text{Eq. 3.5}$$

We can then rank firms with respect to the target firm according to the business divergence and select top firms with lowest divergence as peer firms that constitute the industry for the target firm.

Industry-centric Industry Classification

In some cases, we might not have a target firm and just want to have an overview of the industry and the market. In order to fulfill this type of need, we propose the Industry-centric Industry Classification (ICIC), which is analogous to SIC and NACIS. However, our method can capture the evolvement of the industries since our business genre representations are updated annually to represent the current business of firms. In addition, business divergence between any two firms could be easily calculated through equation 2. In other words, our method is able to overcome the two limitations of existing industry classification schemes aforementioned.

Any clustering algorithm, which can group firms into industries, can fulfill this task. In this study, we select k-means clustering. We believe that k-means is quite suitable for the current problem. It groups points according to their distances to the cluster centers. Another popular type is spectral clustering, which partitions points according to their connectivity. We believe that this criterion is not appropriate for this problem. That two firms are similar a third firm does not mean that those two firms are similar as well - it’s possible that the third firm has two different business segments.

One input parameter for almost all clustering algorithms is the number of clusters. We chose the appropriate cluster number by the sum of squared error (SSE), which is defined as the sum of the squared distance between each point of a cluster and its cluster center. Generally, the value of SSE should decrease as the cluster number increases. We select the number where the reduction of SSE slows dramatically as the cluster number since increasing the number of clusters does not have a substantial impact on the SSE.

3.5 EVALUATION

In this section, we first describe our dataset and evaluation metrics, and then discuss our experimental results.

3.5.1 Experimental Setting

We selected the constituents of the S&P 1500[12] as our firm sample. It combines the S&P 500, the S&P MidCap 400, and the S&P SmallCap 600 and covers approximately 90% of the U.S. market capitalization.

We downloaded 10-K forms filed by firms in S&P 1500 from 2009 to 2011 from the EDGAR[13] database.10-K forms are reported in nonstandardized HTML files and it is very hard to extract information from them (Huang and Li 2011). We found that most of the 10K forms provide links to specific sections in the Table of Contents. Making use of those links as well as the titles of each section in 10K forms, we are able to extract the business section quite effectively. Finally, we have 10-K forms in all three years from 2009 to 2011 for 1453 firms.

We collected the capitalization information from Yahoo! Finance and used GibbsLDA++ (Phan and Nguyen 2007) to learn the topic features. According to the methods discussed

[12] http://us.spindices.com/indices/equity/sp-composite-1500

[13] http://www.sec.gov/edgar.shtml

in the previous section, we chose the number of clusters to be 60 and number of topics to be 50.

3.5.2 Evaluation Metrics

Following Lee et al. (2012), Hoberg and Phillips (2013) and Bhojraj et al. (2003), we evaluate different peer identification and industry classification approaches by how they can help to explain movements in base firms' stock returns. Specifically, estimate the regression specification:

$$R_{i,t} = \alpha_t + \beta_t R_{p,t} + \epsilon_{i,t} \qquad \text{Eq. 3.6}$$

Where $R_{i,t}$ is the monthly return for firm i, $R_{p,t}$ is the average monthly portfolio return based on one peer identification or industry classification approach. In the experiment, we set the portfolio size to be 10. For peer identification, we select the top 10 peers of the base firm. For industry classification, we randomly select 10 firms from the industry the base firm belongs to. Following Lee et al. (2012), Hoberg and Phillips (2013) and Bhojraj et al. (2003), we also ignore the firms that have less than 10 peers.

We run cross-sectional regressions of equation 3.6 for every month from 2009 to 2011 and obtain an average adjusted R^2 based on the 36 regressions. The larger the average adjusted R^2, the more effective the peer identification/industry classification approach.

3.5.3 Evaluation Results

In this section, we first present the results for *firm-centric industry classification (FCIC)* and then discuss the results for *industry-centric industry classification (ICIC).*

Firm-centric Industry Classification

Hoberg and Phillips (2013), which also use 10-K forms for industry classification, is selected as our baseline here. Two different versions were explored: HP EA refers to the baseline method with firms in the portfolio equally averaged; HP WA refers to the baseline method with firms in the portfolio weighted averaged by the similarity with the base firm. For our proposed FCIC method, we also explored two versions – the equally averaged (FCIC EA) and the weight averaged (FCIC WA). In addition, in order to evaluate the effectiveness of considering business scale into industry classification, we also performed industry classification only considering business divergence (FCIC B EA for equally averaged and FCIC B WA for weight averaged).

Sample	HP–EA	HP–WA	FCIC–B– EA	FCIC–B– WA	FCIC–EA	FCIC–WA
S&P 500	0.0619	0.0644	0.1493	0.1655	0.1556	0.1723
S&P 1500	0.0456	0.0474	0.0990	0.1039	0.1125	0.1170

Table 3.1 Average Adjusted R^2 across Methods for FCIC

Notes: HP – baseline method Hoberg and Phillips (2013); EA – equally averaged; WA – weighted averaged.

We run regressions over two samples – S&P 500 and S&P 1500; and the results are presented in Table 3.1. From the table we can see that our proposed methods clearly outperform the baseline with a large improvement (R^2 increased by 100% to 200%), over both S&P 500 and S&P 1500. We could also see that all WA methods produce better results than their corresponding EA methods. This suggests that it is better to weigh the firms according to the similarity with the base firm for portfolio construction.

The table also indicates that including business scale into consideration for industry classification can indeed improve the results. For instance, the R^2 values for S&P500 and S&P 1500 are 0.1655 and 0.1039 respectively for the weighted average portfolio if we only consider business genre divergence; however, if we include business scale into consideration, the corresponding R^2 values increase to 0.1723 and 0.1170. This suggests the effectiveness of considering business scale for industry classification.

We also present top 10 peers for two firms in the following two figures. Figure 3.3 presents peers of Dow Chemical (NYSE:DOW) in 2009 and Figure 3.4 shows peers of Google Inc. (NASDAQ:GOOG) in 2011.

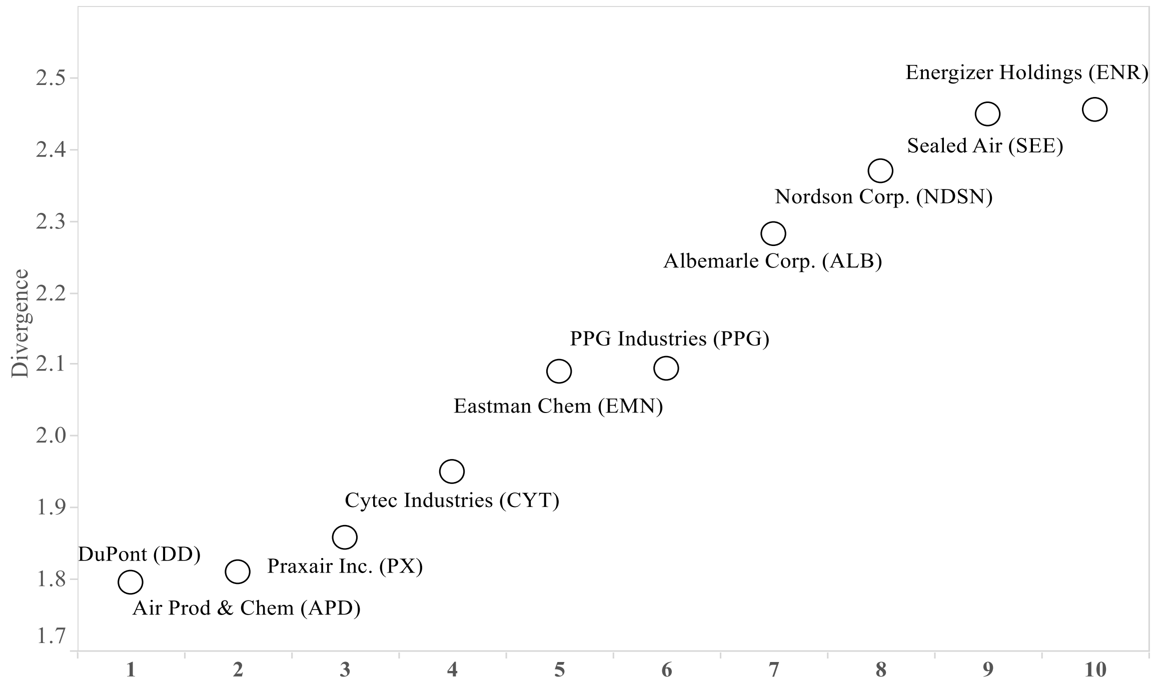

Figure 3.3 Top 10 Peers of Dow Chemical in 2009

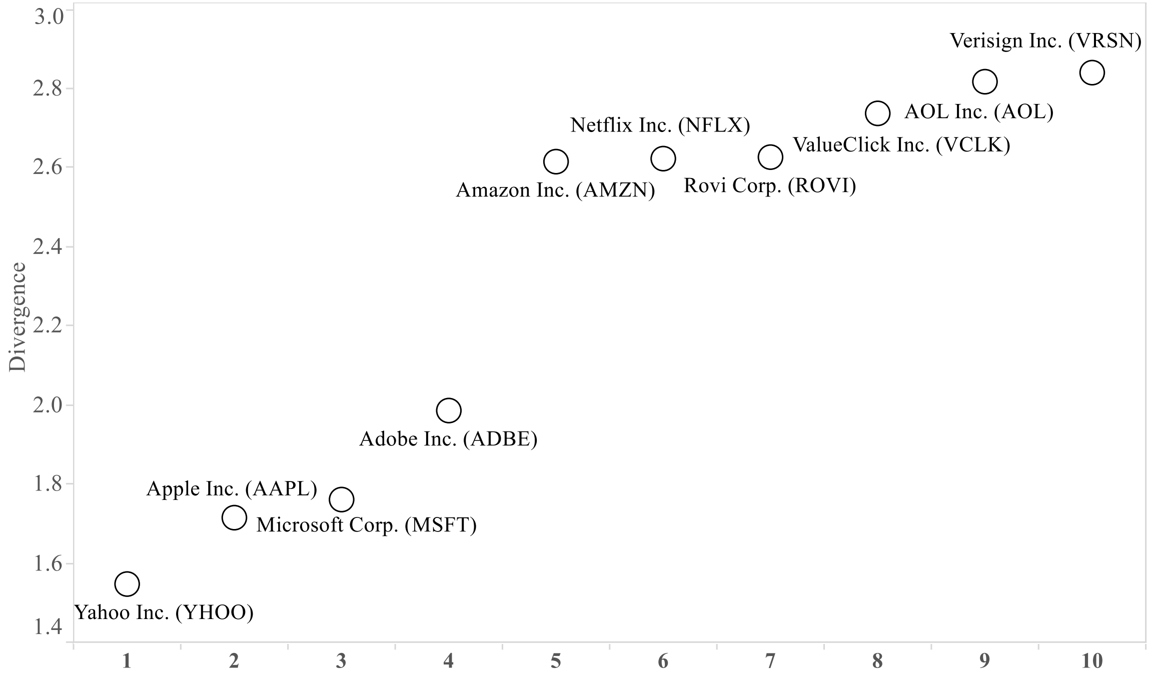

Figure 3.4 Top 10 Peers of Google Inc. in 2011

Industry-centric Industry Classification

For ICIC methods, in addition to Hoberg and Phillips (2013), we also compared our method with GICS. SIC, NACIS and GICS are the three most widely used industry classification schemes. We only compared GICS here since it provides the best performance among the three industry classification schemes (Bhojraj et al. 2003).

Sample	GICS	HP	ICIC
S&P 500	0.0940	0.0538	0.0979
S&P 1500	0.0560	0.0396	0.0602

Table 3.2 Average Adjusted R^2 across Methods for ICIC

The results are presented in Table 3.4. Unsurprisingly, the average adjusted R^2 values of ICIC are lower than those of FCIC. From the table, we can see that ICIC outperforms both the GICS and HP methods for both samples. Specifically, the HP method gives far lower R^2 values than GICS and ICIC while ICIC is slightly better than GICS. These results conclusively demonstrate the effectiveness of our topic feature representation.

We also present the top 5 firms in terms of market capitalization for two industries in the following two tables. Table 3.3 presents firms in the payment industry in 2011 and table 3.4 shows firms in the mass media industry in 2010.

1	Visa Inc. (V)
2	Mastercard Inc. (MA)
3	American Express Co. (AXP)
4	Western Union Co. (WU)
5	Discover Financial Services (DFS)

Table 3.3 Top 5 Firms in Payment Industry in 2011

1	The Walt Disney Co. (DIS)
2	Comcast Corp. (CMCSA)
3	DIRECTV (DTV)
4	Time Warner Inc. (TWX)
5	CBS Corp. (CBS)

Table 3.4 Top 5 Firms in Mass Media Industry in 2010

3.6 CONTRIBUTIONS AND LIMITATIONS

Our research contributes to the industry classification literature by introducing a novel industry classification approach. We introduced the use of topics as features for firm business genre representation, which overcomes the so-called "curse of dimensionality" and sparse data issue. In addition, we considered the business scale as an important factor for firm-centric classification, which avoids identifying two firms with distinct business sizes as peers. Thirdly, our approach makes use of the annually updated business description in 10K forms and conduct industry classification every year, which allows us to adjust the industries as the firms' business change. Fourthly, our approach is capable of measuring the similarity between any two firms, which captures the within industry heterogeneity. Finally, our evaluation results suggest that our proposed method outperform GICS and existing methods in literature.

Besides contributing to literature, this study also benefits the practitioners. Asset managers could use our approach to investigate the target company’s competitive environment and growth opportunities for stock selection and valuation. In addition, credit analysts could use our approach to assess the target company’s financial status through the comparison of industry average for company rating. Thirdly, investors could use our approach to study the target industry for investment decision-making. Finally, researchers could use our approach to design appropriate control groups for their studies.

There are several limitations in this study. Firstly, there is no ground truth available for industry classification and peers identification. Thus, it is impossible to attain precise and accurate results using our methods. However, our methods outperform existing ones according to the experimental results. Secondly, the effectiveness of the proposed methods depends on the accuracy of the business section in the 10K forms. If the business activity descriptions provided by the companies are not accurate, the effectiveness of our methods would also be affected.

3.7 CONCLUSION AND FUTURE RESEARCH

In this study, we proposed a novel approach for industry classification based on the topic features learned by the LDA model. Two types of classification – firm-centric classification and industry-centric classification were explored. The evaluation results showed the effectiveness of our method.

There are several future directions for this study. Firstly, we would like to extend the current work by applying the approach to other markets other than US market. Secondly, we would like to apply the proposed industry classification method to various application settings. For instance, selecting peers for the purpose of evaluating CEO compensation.

CHAPTER 4 STUDY III: MOBILE APPLICATIONS DOWNLOAD ESTIMATION

4.1 INTRODUCTION

Mobile apps represent the fastest growing consumer product segment of all times (Kim 2012), dwarfing other consumer goods that have exhibited hyper-consumption, such as digital music. Congruent with the consumption numbers, the production scale of apps is eye-popping as well – approximately 15000 new apps are launched every week, compared to 100 movies and 250 new books that are released weekly (Datta et al. 2012). The total number of apps created to run on smart phones exceeds 1.3M currently, with Apple[14] and Android[15] markets representing the bulk of these apps [16]. Not surprisingly, the dollars spent on the app space is substantial: about $15B spent in app stores in 2012 (Manninen 2012). Without doubt, apps are big business.

To be successful, an app needs to be popular. The most commonly used measure of app popularity is the number of times it has been downloaded into consumers' smart-devices. Effectively, the download count of an app (which we will simply refer to as "downloads") is the make-or-break metric that determines whether the publisher will reap the rewards of having invested in creating the app – (a) if it is a paid app, the downloads will determine the revenue the app generates, (b) if it is an ad-driven app, the downloads will

[14] http://itunes.apple.com/gb/genre/ios/id36?mt=8 [Accessed July 29, 2012]
[15] https://play.google.com/store?hl=en [Accessed July 29, 2012]
[16] http://en.wikipedia.org/wiki/App_Store_(iOS) [Accessed July 29, 2012]

determine the price of advertising on this app and (c) in some cases, e.g., Instagram[17], the downloads will even cause the publishing company to be valued at hundreds of millions (indeed, in some cases billions) of dollars. Every app creator wants to create apps that will garner massive downloads. A multi-billion dollar industry has emerged just to help apps achieve popularity, referred to as the mobile marketing industry, which participants such as Fiksu[18] and Tapjoy[19].

In addition to its huge business value, app download numbers are also quite valuable from a research perspective. The rapid growth of app market offers an excellent place for studies such as innovation (Boudreau 2011), competitive strategies in hypercompetitive market (Kajanan et al. 2012) and so on. Studies in the app market necessities download numbers to measure the success of an app.

However, it turns out that number of download is one of the most closely guarded secrets in the mobile industry – only the native store knows the download number of an app. Even developers can only have access to the limited information on downloads of their own apps. For instance, Apple does not provide downloads of an app from iPhone and iPad separately to its developer – only the aggregated download is provided. Yet, the knowledge of downloads could benefit many – competitors could benchmark themselves against their rivals, advertisers could judge the attractiveness of specific app media, consumers could get a “real” sense of popularity, and researchers could use it to measure

[17] http://instagram.com/ [Accessed July 29, 2012]
[18] http://www.fiksu.com [Accessed July 29, 2012]
[19] https://www.tapjoy.com/ [Accessed July 29, 2012]

the success of an app. Even though app stores release lists of top apps for each category as well as overall, which is ranked according to the download numbers and typically include 400 apps in each list, it is still necessary to know the number of download since those ranks cannot fully cater the needs of the aforementioned groups of people. For instance, it is really hard for competitors to get an actual view on how far it behind their rivals – one rank lower might means hundreds of thousands downloads less or may be actually quite close. In addition, ranks in Apple app store are country-specific and Android market only releases global ranks, which disable the comparison of downloads across countries. However, developers might need this information to allocate their limited resources and furthermore, maximize the profits. Therefore, app download numbers are extremely advantageous for both researchers and practitioners.

As a result, in recent times, there has been interests in estimating app downloads (Garg and Telang 2012). However, much of this work suffers from inaccuracies deriving from inadequate data, e.g., they only tested their model over 12 data points (from one app), as well as flawed assumptions regarding the app market, which will be discussed in detail later. In addition, the method described in that work is only applicable for paid apps. Yet, we believe that download information of free apps is extremely important, probably even more important than paid apps. Though the free apps could be downloaded costless, their in-app purchase features can produce enormous revenue, that is, more than the revenue generated by paid apps. Our data indicates that in the top grossing app list where apps are ranked by the revenue generated, about 58% of the apps are free apps. More importantly, the portion of free apps increases to 74% and 92% for the top 100 and top 10 grossing apps respectively. This clearly demonstrates the importance of free apps and the download information of those apps, which is yet to be investigated in the literature.

In this study, we introduce a model of *daily app downloads estimation* that, to our knowledge, is the most accurate estimation model available. We focus on free apps in the current work, which complements Garg and Telang (2012). We base our work on a large real-life dataset obtained from one of the app analytics firms and consider app ranks, time effect as well as the category effect for download estimation.

The rest of the chapter is organized as follows: section 2 introduces related work in literature and then we elaborate our proposed model in section 3. Section 4 presents the approach to estimation the proposed model. Experimental results are represented in section 5 and section 6 concludes this chapter.

4.2 RELATED WORK

In spite of the value of the mobile app download information, there exists scant work on demands or downloads of mobile apps. Garg and Telang (2012) presented an approach to infer downloads for paid apps on Apple’s App Store using publicly available data. They utilized the Top Paid Apps list and the Top Grossing list, which are released by Apple. Two drawbacks of this work are: (1) they assume paid apps do not have in-app purchase. Though app sales revenue still dominates the revenue for paid apps now, we believe it is not desirable to ignore it, especially considering that the in-app purchase is set to dominate the app business (Kent 2012). (2) In their work, they simply treated the estimated aggregate download of top 300 paid apps from Distimo[20] as the exact download, which would inevitably bring error and make the accuracy of their estimates dependent on the accuracy of Distimo's estimates. (3) They only evaluated their model over 12 data points from one app and we believe it is not enough to demonstrate the effectiveness of their model convincingly.

Our current work distinguishes from Garg and Telang (2012) in several aspects. First, we focuses on free apps while Garg and Telang (2012) focused on paid apps. As we discussed earlier, free apps constitute 74% and 92% of the top 100 and top 10 grossing apps, respectively, which suggests that free apps are much more important. In Garg and Telang (2012)'s short discussion of free app download estimation, they replaced the price with in-app purchase price and followed similar procedure for the free app download estimation. Their method implies that: (1) equal revenues from the in-app purchase are generated on each app and (2) in-app purchases are made only once when the app is downloaded. We believe both implications are unrealistic: (1) apps typically provide more than one in-app purchase package which would definitely result in unequal revenue and (2) users typically make in-app purchases after a certain time of their purchase and could make more than one in-app purchase on a single app, implying that an increase of revenue is not necessarily resulted from an increase of app download. Second, we also consider the time effect, which is effective according to our experimental results. Finally, we estimate a distinct model for each category while Garg and Telang (2012) only

[20] http://www.distimo.com/ [Accessed December 1, 2012]

estimate one overall model. As we will discuss later, estimating a model for each category would expand the scope of apps that could be estimated.

Ghose and Han (2012) estimated how the demand of a mobile app may change with the price, size and age of the app as well as the length of the app description. They find that app demand increases with the size of apps and the length of description, but decreases with the age of apps.

In spite of the scantiness of work on the demands or downloads of mobile apps, there exists a number of work that have investigated product sales in Amazon (Brynjolfsson et al. 2003, Chevalier and Mayzlin 2006, Ghose and Ipeirotis 2011), which is similar to the app download estimation, and has approximated sales using sales rank (Chevalier and Goolsbee 2003). In these work, factors such as number of reviews, average rating and price have been demonstrated as important factors influencing sales. The impact of readability and spelling errors in reviews on product sales has also been examined (Ghose and Ipeirotis 2011).

4.3 MODEL

In this section, we first give a overview of our download estimation model and then elaborate each compont of the model.

4.3.1 Overview

We aim to estimate the download of a particular mobile app in a specific day, i.e., the daily download. Because of the limitation of the data available, in this research, we focus on the daily download estimation of free apps on Apple iOS platform. However, our method could be easily adopted for other apps given the appropriate data.

Overall, our model consists of two sets of variables. The first set is the rank of an app. Though app stores do not disclose the exact download, they provide top app lists where download is the most important ranking criteria (Venturedata 2012). Sales ranks have been used in a number of research on demand estimation in the literature since the inception of the Amazon, though none of them focus on mobile apps. In this research, we will also use the rank to estimate the app download. Besides the rank, a set of dummy variables capturing the individual time effect is also included. Existing research suggested that time plays an important role in the download of an app (Henze and Boll 2011).

In the model, we do not consider number of reviews, average rating that have been demostrated to be important factors influencing product sales (Chevalier and Goolsbee 2003, Ghose and Han 2012). We believe that in case of mobile apps, the effects of the factors are reflected in the ranks – the more the reviews and the higher the rating, the higher the rank of an app is. Thus, rank incorporates all these factors that influence the download of a specific app and we only need to consider other factors that affect the download numbers of all apps simultaneously, for instance, the time variable.

4.3.2 Rank

Though the Apple app store does not disclose the exact download, it publishes top app lists and download is the most important ranking criteria (Venturedata 2012). Apple releases three types of list, namely, top free app list, top paid app list and top grossing app list. These lists are updated several times in a day. In the first two types of lists, apps are ordered according to Apple's own formula for ranking an app. However, it has been demonstrated that the number of download is the most important factor. In the third type of lists, ranking criterion is the revenue generated by the app.

Apps are grouped into various categories in the store. Currently, Apple app store has 23 categories. For each category, the store publishes top free app list, top paid app list and top grossing app list. Furthermore, Apple has two major platforms, that is, iPhone and iPad, so six lists are published every day for each category. Besides these category-specific ranks, Apple app store also releases the overall top app lists, which rank the apps in all categories, for both iPhone and iPad platform. So in total, we have 144 top app lists for an Apple app store. Note that Apple maintains 155 apple stores for 155 different countries.

In this research, we assume the download and the rank follow a power law. This assumption is used in many existing work on demand estimation (Chevalier and Goolsbee 2003). According to the power law, the download and the rank follow a log-linear distribution:

$$\ln(download) = \beta_0 + \beta_1 \ln(rank) \qquad \text{Eq. 4.1}$$

where β_0 and β_1 are constants.

One limitation of this method is that an app has to be ranked so that its download can be estimated and only a small portion of the apps appears in the ranking list. We could only acquire top 1000 free apps for iPhone platform and top 200 free apps for iPad platform.

Thus, for 23 categories, we can consider maximum 27600 apps. However, we do not think this is a problem. First, though the number of apps in the ranking list is small, they contribute the major part of the total download. Downloads of unranked apps are quite small and less likely to be useful. Second, the estimated download range is actually quite large. According to our experimental results, the download of the lowest ranked app in the list is typically around several hundreds. However, the download of the top ranked app is around hundreds of thousands. In other words, we can estimate the majority of the possible download value range. Lastly, very few real-world distributions follow a power law over the entire range. Newman (2005) points out that the distribution must deviate from the power law form below some minimum frequency. In our case, the distribution would deviate when the rank is too low. For real application, it is usually necessary to choose a minimum frequency, or a maximum rank value, to make sure that the data follows a power law. In this research, we only estimate the download for ranked apps, which indicates that our maximum rank value is 1000 for iPhone platform and 200 for iPad platform.

As we mentioned before, Apple offers two platforms and users could download apps on either iPhone/iTouch or iPad and most apps are available on both platforms. So in this research, we will estimate two sets of models, one for iPhone platform and the other for iPad platform.

Mobile apps are structured in native store in various categories. We have observed the apps in “Games” category, in general, have higher downloads than the apps in “Weather” or “Catalog” category. A large number of apps in a category indicates a larger demand of apps from that category, and so, it is more likely that apps in that category will have more downloads. We could infer that apps with the same rank in different categories would have varied actual download. In order to solve this issue, we estimate a distinct model for each category. We believe addition of dummy variable would not help much here because shape of the distribution might be quite different across categories.

Another reason we estimate a model for each category is to expand the scope of the apps that could be estimated. As discussed earlier, we could only acquire top 1000 apps for the iPhone platform and top 200 apps for the iPad platform in the overall ranking list. Thus, if we use the overall rank instead of the category rank, only 1000 apps for iPhone and 200 apps for iPad could be estimated. However, if the category rank is used, we are able to estimate 1000 apps for iPhone and 200 apps for iPad for each category, that is, maximally 27600.

4.3.3 Time Effect

In the previous section, we linked the rank to the actual download. However, rank is a relative measurement. So we also need to consider some factors that affect the download of all apps simultaneously.

Henze and Boll (2011) suggested that the time plays an important role in the download of an app and Sunday evening is the best time to release an app. This type of effect obviously could not be captured by the rank alone. In order to overcome this issue, we include the time effect in our estimation model by addition of a set of time dummy variables and the final daily download estimation model becomes as follows:

$$\ln(download) = \beta_0 + \beta_1 \times \ln(rank_c) + \sum_{i=1}^{6} \beta_2^i \times day_i \qquad \text{Eq. 4.2}$$

or

$$download = \exp(\beta_0 + \beta_1 \times \ln(rank_c) + \sum_{i=1}^{6} \beta_2^i \times day_i) \qquad \text{Eq. 4.3}$$

where $rank_c$ is the category-specific rank and day_i refers the i^{th} day of the week. Please note that we only have 6 dummies, representing Monday to Saturday, to avoid the dummy variable trap.

4.4 MODEL ESTIMATION

In this section, we elaborate our approach to estimate the parameters of our model introduced in the previous section. As discussed in the previous section, we estimate a distinct model for each category. We first describe the method for categories in which we have actual download data of at least one app. We call this direct estimation. Next we present the approach for categories in which actual download data of any app is not available, i.e., indirect estimation.

4.4.1 Direct Estimation

When actual download data is available, model estimation is quite straightforward. A simple linear regression would derive the model. Suppose we have the actual download numbers of a ranked app A in category C, we can run a regression over the download data and ranks of A and estimate coefficients in equation 2. The estimated download of any ranked app in category C, say app B, could be easily derived using these estimated coefficients and ranks of app B via equation 3. In the experiment, we estimate our model using the ordinary least squares (OLS) estimator in R[21].

4.4.2 Indirect Estimation

One of the challenges of this research is the model estimation due to the lack of data. So the more common situation is that the actual download of any app is not available for a category whose model we intend to estimate.

Let A be an app that we have actual download data and A belongs to category K. Let $rank_A$ be ranks of A, $download_A$ be the download data of A and $MODEL_K$ be the download estimation model for category K derived through direct estimation. Let APP^K be the intersection of the top app list of category K and the overall top app list, $RANK_C^K$ and $RANK_O^K$ be the category ranks and overall ranks of apps in APP^K. Let U be the category in which the actual download of any app is unavailable and we want to estimate the download for an app, say app B, in category U. Let APP^U be the intersection of the category U top app list and the overall top app list, $RANK_C^U$ and $RANK_O^U$ be the category ranks and overall ranks of apps in APP^U, $MODEL_U$ be the download estimation model for category U. We would like to estimate the download model for category U, $MODEL_U$, using download data of app A.

[21] http://www.r-project.org/ [Accessed July 29, 2012]

Estimation of $MODEL_U$ seems unreachable, since we do not have the actual download data of any app in category U. Nevertheless, if we have estimated downloads of some apps in category U, we should be able to estimate $MODEL_U$. We observe that a number of apps in category U are ranked on both the top app list of category U and the overall top app list, i.e., APP^U. So those apps have two types of ranks, namely, the category rank and the overall rank. Thus, having the overall download estimation model $MODEL_O$, we can use the overall rank to estimate the download of any app in APP^U, which can in turn used to derive $MODEL_U$.

In order to estimate $MODEL_O$, we need to have download data of at least one app that is ranked in the overall list. In this research, we consider a worse, more common situation where we do not have the actual download values of an app ranked in the overall list. We utilize APP^K, the intersection of the top app list of category K and the overall top app list. If we have estimated downloads of apps in APP^K, we could derive $MODEL_O$. First, we use the download and rank values of app A to derive $MODEL_K$. Next, using $MODEL_K$, we derive the download estimates of apps in APP^K. From this estimated downloads, we derive $MODEL_O$.

So when the actual download of any app is not available for a category whose model we intend to estimate, we utilize the overall top app list. Specifically, we use the overlap between the category top app list and the overall top app list as a bridge to estimate the model for the category in which actual download data of any app is unavailable based on download data of an app in any other category.

Thus the summary of the estimation procedure is as follows: we first estimate $MODEL_K$ based on the actual download data of app A. Next, with the help of $MODEL_K$ together with $RANK_C^K$, the estimated download of each app in APP^K is calculated. Subsequently, we treat these estimated downloads as the actual ones, and estimate the overall model, $MODEL_O$, taking advantage of the overall ranks $RANK_O^K$. After we acquire the overall model, $MODEL_O$, we then calculate the estimated download for each app in APP^U, using the set of ranks $RANK_O^U$. Finally, we use those estimated downloads and the category ranks $RANK_C^U$ to estimate the model for category U, $MODEL_U$.

Through the above procedure, we could estimate the download model for any category as long as we have actual download data of apps for at least one category. This significantly increases the applicability of our approach across broad category and app ranges.

4.5 EVALUATION

This section first introduces the data set used in our experiment and then discusses the experimental results.

4.5.1 Data Set

Training Data

Exact number of mobile apps download is extremely hard to acquire. However, through Mobilewalla (Datta et al. 2012), we acquired the exact daily download of three free apps - one from the Medical category (from May 4, 2011 to August 23, 2012) and the other two from the Lifestyle category (one from October 17, 2011 to October 10, 2012 and the other one from May 20, 2011 to October 10, 2012). We believe that this dataset itself is quite valuable, which makes our current work possible.

The two Lifestyle apps are only available on iPad, thus, this set of data will be used to estimate the iPad apps download model. In addition, the Medical app is available on both iPhone and iPad, and therefore, the download is aggregate download for both platforms. In order to estimate our iPhone apps download model, we need to split the download of the Medical app and get the download on iPhone only. Using the aforementioned Lifestyle apps download data and the approach introduced in section 4, we could estimate the iPad apps download model for the Medical category. Then we calculated the estimated download from iPad for the Medical app, subtract it from the total download, and get the download from iPhone.

Now we successfully acquired the download data we need to estimate our model. Next we collect the corresponding ranks from Mobilewalla (Datta et al. 2012). Since Apple updates ranks several times in a day, those ranks we collected are average ranks. We refer this data set as Training Data I. The descriptive statistics of this data set are shown in Table 4.1.

	$Rank_{iPhone}$	$Download_{iPho}$	$Rank_{iPad}$	$Download_{iPad}$
# of Instance	477	477	862	862
Average	18.32	761.30	62.78	977.75
Standard Deviation	12.69	280.46	38.90	1109.99

Table 4.1 Descriptive Statistics of the Training Data I

Category	iPhone		iPad	
	# of Apps	# of Instance	# of Apps	# of Instance
Medical	85	2541	15	69
Books	205	9371	169	2977
Games	5384	156884	2138	38206
Lifestyle	821	41565	177	7827
Photo & Video	783	24278	139	2075
Utilities	854	31340	142	4562

Table 4.2 Descriptive Statistics of the Training Data II

We also have actual weekly download data of a few apps from Medical, Books, Games, Lifestyle, Photo & Video and Utilities category and these data will be used for testing purpose. As we discussed in section 4, in order to estimate the download model for categories in which actual download data of any app is not available, we need ranks (overall rank and category rank) of apps that are ranked both in the overall top app list and the category top app list. So we collect these ranks from Mobilewalla for the aforementioned 6 categories. We refer this data set as Training Data II. Because of the different degrees of popularity of different category, the number of instances we could collect varies. The descriptive statistics are shown in Table 4.2. Please note that an instance means a combination of app, date, overall rank and category rank.

Testing Data

We also acquired the weekly downloads of a small set of free apps in following categories: Books, Games, Lifestyle, Photo and Video, and Utilities. Again, we collected corresponding ranks from Mobilewalla. The descriptive statistics are shown in Table 4.3. We use the date of a Monday to represent the week it belongs to. Please note that an instance means a combination of app, download, week and category rank. The download data might be missing for certain weeks.

Category	# of Apps	# of Instance	Date Range	Average	S. D.
Books	1	21	Apr. 2, 2012 ~ Aug. 26, 2012	2621.91	260.43
Games	4	42	May 28, 2012 ~ Aug. 26, 2012	8345.78	3025.01
Lifestyle	1	5	May 28, 2012 ~ Jul. 1, 2012	15125.25	1739.39
Photo & Video	1	4	Jul. 30, 2012 ~ Aug. 26, 2012	2346.25	765.28
Utilities	2	21	Apr. 2, 2012 ~ Jul 1, 2012	12441.24	3425.82

Table 4.3 Descriptive Statistics of the Testing Data

All the three data sets are used in our experiment to build the model and test it. We estimate our model using the ordinary least squares (OLS) estimator in R[22].

4.5.2 Estimation Results

Our estimation results for iPhone apps are listed in Table 4.4. The results suggest that the most downloaded app has about 700K downloads in a single day. The exponent of the β_0 roughly represents the number of download of the app ranked first in list. So the value of β_0 indicates the popularity of the category it represents. From the table we can see that, Games category is the most popular category and the Medical category has the least downloads. According to our estimation, for the iPhone platform, the top ranked Games app have about 700K downloads on a single day. On the contrary, a Medical app only need around 5.5K downloads to be ranked first. This is also reflected by the number of

[22] http://www.r-project.org/ [Accessed July 29, 2012]

apps in the store as we find that there are much more Games apps than Medical apps in the Apple app store.

Category Coefficient	Overall	Medical	Books	Games	Lifestyle	Photo & Video	Utilities
β_0	13.5310	8.6036	9.7878	13.5166	10.5953	11.0310	11.1289
β_1	-0.8683	-0.7232	-0.7801	-1.0242	-0.6885	-0.9084	-0.8685
β_2^1	-0.0233	0.0029	0.0037	-0.0647	-0.0044	0.0029	-0.0057
β_2^2	-0.1202	-0.0262	-0.0769	-0.1906	-0.0950	-0.0876	-0.0894
β_2^3	-0.1770	-0.0808	-0.1252	-0.2646	-0.1436	-0.1486	-0.1367
β_2^4	-0.2187	-0.1262	-0.1515	-0.3009	-0.1891	-0.2019	-0.1776
β_2^5	-0.2940	-0.2295	-0.2528	-0.3601	-0.2770	-0.2858	-0.2589
β_2^6	-0.1527	-0.1659	-0.1581	-0.1626	-0.1574	-0.1687	-0.1473

Table 4.4 Model Estimation Results for iPhone Apps

Another interesting phenomenon is that nearly all coefficients of time dummies are negative, indicating that Sundays are the days that have most number of apps downloaded. This is consistent with the result of Henze and Boll (2011).

The estimation results of iPad apps download are listed in Table 4.5. Similar with iPhone results, the exponent of the β_0 roughly represents the number of download of the app ranked first in list. Again, Games apps is the most popular category and produce the most downloads. Based on the results, we can calculate that the number of download on iPad is much less than the number of download on iPhone. For instance, on Mondays, the top ranked app on iPad would have approximately 343K downloads while about 735K copies should be downloaded for an app to be rank first in the iPhone ranking list.

Again, almost all coefficients of time dummies are negative suggesting that Sundays are the days that have most number of apps downloaded on the iPad platform.

Category Coefficient	Overall	Medical	Books	Games	Lifestyle	Photo & Video	Utilities
β_0	12.9990	6.7143	10.5696	12.7791	10.0192	9.2635	10.1435
β_1	-1.0003	-0.5667	-1.5497	-1.1679	-0.8346	-0.8896	-1.1230
β_2^1	-0.2508	-0.1742	-0.2219	-0.2913	-0.2536	-0.2042	-0.1171
β_2^2	-0.2833	-0.5264	-0.1786	-0.3474	-0.3141	-0.1630	-0.0404
β_2^3	-0.2804	-0.2606	-0.1680	-0.3715	-0.3269	-0.1040	0.0146
β_2^4	-0.2725	-0.4134	-0.1365	-0.3471	-0.3388	-0.1338	-0.0009
β_2^5	-0.2399	-0.3729	-0.1063	-0.2667	-0.3543	-0.1442	-0.0220
β_2^6	-0.0506	-0.1695	0.0007	-0.0412	-0.1175	-0.0339	0.0085

Table 4.5 Model Estimation Results for iPad Apps

4.5.3 Estimation Accuracy

In this section, we first evaluate our model over a set of actual weekly download data (Table 3). Next, we compare our estimated daily aggregate of top 200 iPhone apps downloads with the App Store Competitive Index (Fiksu 2012).

Comparison with Actual Download

First, we compute the estimated download of the apps in the testing data set using the approach described in section 4 and compare them with the actual download. Because we do not have actual download data of additional app in Medical category for testing purpose, for the Medical category, we randomly select 80% of the data in Training data set I as training data and the remaining 20% for testing. As we mentioned previously, for testing purpose, we only have weekly download data as shown in Table 4.3. So we estimate the daily download from Monday to Sunday and sum them up to get the estimated weekly download.

We use the percentage error to measure the estimation accuracy. The percentage error of a particular instance is calculated as follows:

$$error = \frac{|estimated\ download - actual\ download|}{actual\ download} \times 100\%$$

The average errors for each category are shown in Table 4.6.

Model Category	Without Time Variables	With Time Variables
Medical	24.2	22.7
Books	47.9	39.2
Games	29.6	12.8
Lifestyle	12.9	16.3
Photo & Video	31.7	31.0
Utilities	28.1	33.3
Average	29.7	25.9

Table 4.6 Estimation Error

All values are in percentage

On average, our method achieves 29.7% error with only rank data and further reduces the error to 25.9% when time variables are included. We believe this error falls in a range that is acceptable for real life practice.

The 3.8% reduction of average error when time variables are added indicates the effectiveness of those variables. However, errors do not descrease in all categories with the addition of time dummies. Sepcificly, in Medical, Books, Photo & Video and Games category, the inclusion of time dummies effectly reduces the estimation error and the addition of dummies results into a increase of estimation error in Lifestyle and Utilities category. This might suggest that the demands of the Lifestyle and Utilities apps are relatively stable and thus, it is not necessary to consider the time effect in this context. So for these categories, a model without time variable would be more appropriate.

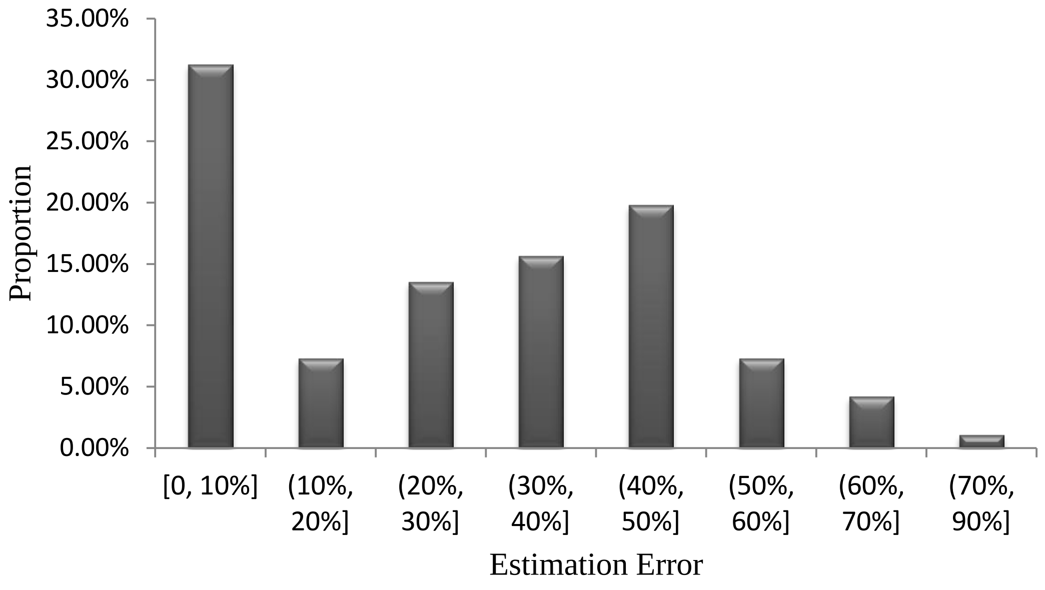

Figure 4.1 Estimation Error Distribution

Estimation error ranges from 12.8% to 39.2% across categories with the time variables. Games category has the lowest error which is little surprising. We expect the Medical or Lifestyle category to have the lowest error since we have actual data for those two categories. We speculate the reason why estimation for Games category is more accurate than the rest is that Games category has far more instances than other categories in Training Data Set II as shown in Table 4.2. There are far more Games apps that appear in the overall ranking list and thus, we have much more instances for training.

Besides the average estimation error, it is also interesting to see the error distribution, which further offers us information on how our models work. For sake of space, we combine testing instances from all categories and the results are shown in Figure 4.1. The horizontal axis represents the estimation error. The vertical axis represents the portion of the instances in percentage. For example, the second column from the left represents that about 31% of the instances has estimation error between 0% and 10%.

From the figure we could see that about 40% of the instances have errors below than 20%. Specially, there are about 31% of instances that has estimation error below than 10%, which is quite accurate. In addition, aproximately 10% of the instances have estimation error greater than 50%, suggesting that the probability of our model giving large error is relatively low. Furthermore, there is no instance that has error more than 90% error.

We could not compare our model with other counterparties since there is no comparable work in the literature. Though Garg and Telang (2012) had a short discussion on free app download estimation, they have not reported the accuracy of their approach.

Comparison with App Store Competitive Index

App Store Competitive Index (Fiksu 2012) is maintained by Fiksu and tracks the monthly average aggregate downloads per day achieved by the top 200 ranked free iPhone apps in the United States. They estimated the aggregate daily downloads of top 200 apps ranges from 4.05 million to 6.79 million with an average of 5.01 million for the time period of October 2011 to September 2012

Using the model estimated in the previous section, we calculated our estimation of the aggregate daily downloads of top 200 free iPhone apps in US market. Our estimation ranges from 4.62 million (on Fridays) to 6.2 million (on Sundays) with an average of 5.41

million, which fits quite well with the App Store Competitive Index. This result convincingly demonstrates the effectiveness of our download estimation model.

4.6 LIMITATIONS AND FUTURE DICECTIONS

There are some limitations of this study. Firstly, given some of our models are estimated indirectly, the results may not quite accurate. Secondly, our testing set is relatively small. The estimation errors may be understated or overstated. Thirdly, since rank is an independent variable in the model, we could not estimate the downloads of the unranked apps, though we can have an upper bound.

Our work can be extended in servals ways. Firstly, though the estimation error is relatively low, there is still room for improvement. One obstacle is the limited amount of data for training. We could acquire more data and then improve the accuracy of our model. Additionally, we can also extend our download estimation model to paid apps as well as apps in markets other than the US. One possible approach to do this is to assume that the number of downloads is proportional to the number of ratings since it is obviously impossible to acquire actual download data for all the markets.

4.7 CONCLUSION

In this study, we proposed an approach for mobile app download estimation with the help of app ranks released by official app stores. Time and category effects are also considered in our model. Our estimation corresponds quite well with the App Store Competitive Index. In addition, we tested our model on a real-life dataset and the experimental results suggested that our approach could achieve 25.9% estimation errors on average. In addition, the error distribution indicated that about 40% of the instances have errors below than 20%, and only approximately 10% of the instances have estimation errors greater than 50%.

CHAPTER 5 CONCLUSION

This thesis has focused on three predictive data analytics problems that are important to firms' business and their management teams' decision-making. Specifically, study I focused on cross-domain sentimental classification when labeled data in the target domain was not available. Study II proposed a novel approach for industry classification and peer firm identification based on 10-K forms. Study III explores the estimation of mobile app downloads using app ranks.

Study I focused on sentiment classification and proposed a novel framework for cross-domain sentiment classification using latent features and opinionated word features. This study has contributions in two aspects: firstly, to our best knowledge, this study provides the first attempt to combine the sentiment information from source domain labeled data and hand-picked opinionated words together for the cross-domain sentiment classification task; secondly, the proposed methods, both Intelligent Single Source Domain (ISSD) and Multiple Source Domain (MSD), statistically outperform the existing work addressing the same problem according to the experiment.

Study II focused on competitor identification and proposed a novel approach for industry classification and peer identification based on the topic features learned by the LDA model. This study has contributions in several aspects: firstly, it introduced the use of topics as features for firm business genre representation, which overcomes the so-called "curse of dimensionality" and sparse data issue; secondly, this study included business scale into consideration for peer identification and the experimental results demonstrate its effectiveness; thirdly, this study proposed an approach that is capable of measuring the similarity between any two firms, which captures the within industry heterogeneity; fourthly, the experimental results suggests that the proposed approach outperforms GICS and Hoberg and Phillips (2013).

Study III focused on the estimation of mobile app downloads and introduced a download estimation model for free apps which complements Garg and Telang (2012). Specifically, study III utilized app ranks released by official app stores, as well as time and category for app downloads estimation. According to an experiment on a real-life dataset, the proposed approach can achieve 25.9% estimation error on average.

REFERENCE

Abbasi, A., and Chen, H. 2008. CyberGate: A Design Framework and System for Text Analysis of Computer-Mediated Communication, *MIS Quarterly*, 32(4):811–837.

Bhojraj, S., and Lee, C. M. C. 2002. Who Is My Peer? A Valuation-Based Approach to the Selection of Comparable Firms, *Journal of Accounting Research*, 40(2):407–439.

Bhojraj, S., Lee, C., and Oler, D. 2003. What's My Line? A Comparison of Industry Classification Schemes for Capital Market Research, *Journal of Accounting Research*, 41(5):745–774.

Blei, D., Ng, A., and Jordan, M. 2003. Latent Dirichlet Allocation, *Journal of Machine Learning Research*, 3(1):993–1022.

Blitzer, J., Dredze, M., and Pereira, F. 2007. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification, *Proceedings of 45th Annual Meeting of the Association of Computational Linguistics (ACL'07)*, Prague, Czech, 187–205.

Bollegala, D., Weir, D., and Carroll, J. 2011. Using Multiple Sources to Construct a Sentiment Sensitive Thesaurus for Cross-Domain Sentiment Classification, *Proceedings of 49th Annual Meeting of Association for Computational Linguistics (ACL'11)*, Portland, USA, 132–141.

Boudreau, K. 2011. Let a Thousand Flowers Bloom? An Early Look at Large Numbers of Software App Developers and Patterns of Innovation, *Organization Science*, Forthcomin:

Brynjolfsson, E., Hu, Y. (Jeffrey), and Smith, M. D. 2003. Consumer Surplus in the Digital Economy: Estimating the Value of Increased Product Variety at Online Booksellers, *Management Science*, 49(11):1580–1596.

Buhmann, M. D. 2003. *Radial Basis Functions: Theory and Implementations*, Cambridge University Press.

Carreira-Perpinan, M. A., and Hinton, G. 2005. On Contrastive Divergence Learning, *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTATS' 05),* Savannah Hotel, Barbados, 33–40.

Chevalier, J., and Goolsbee, A. 2003. Measuring Prices and Price Competition Online: Amazon Vs. Barnes and Noble, *Quantitative Marketing and Economics*, 1(2):203–222.

Chevalier, J., and Mayzlin, D. 2006. The Effect of Word of Mouth on Sales: Online Book Reviews, *Journal of Marketing Research*, 43(3):345–354.

Chong, D., and Zhu, H. 2012. Firm Clustering based on Financial Statements, *Proceedings of 22nd Annual Workshop on Information Technologies and Systems (WITS)*, Orlando, Florida, USA, 43–48.

Conrad, D., and DeSouza, G. N. 2010. Homography-based Ground Plane Detection for Mobile Robot Navigation Using a Modified EM Algorithm, *Proceedings of 2010 IEEE International Conference on Robotics and Automation*, Alaska, USA, 910–915.

Datta, A., Dutta, K., Kajanan, S., and Pervin, N. 2012. Mobilewalla: A Mobile Application Search Engine, *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, 95(5):172–187.

Davis, R., and Duhaime, I. 1992. Diversification, Vertical Integration, and Industry Analysis: New Perspectives and Measurement, *Strategic Management Journal*, 13(7):511–524.

Ding, X., Liu, B., and Yu, P. 2008. A Holistic Lexicon-based Approach to Opinion Mining, *Proceedings of the 1st Conference on Web Search and Web Data Mining (WSDM' 08)*, Palo Alto, California, USA, 231–240.

Fan, J. P. H., and Lang, L. H. P. 2000. The Measurement of Relatedness: An Application to Corporate Diversification, *The Journal of Business*, 73(4):629–660.

Fiksu. 2012. App Store Competitive Index,

Gantz, J., and Reinsel, D. 2012. The Digital Universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East, 1–16.

Garg, R., and Telang, R. 2012. Inferring App Demand from Publicly Available Data, *MIS Quarterly*, Forthcoming.

Gehrke, A., Sun, S., Kurgan, L., Ahn, N., Resing, K., Kafadar, K., and Cios, K. 2008. Improved Machine Learning Method for Analysis of Gas Phase Chemistry of Peptides, *BMC Bioinformatics*, 9(515):1–15.

Ghahramani, Z. 2004. Unsupervised Learning, *Advanced Lectures on Machine Learning*, 72–112.

Ghose, A., and Han, S. P. 2012. Estimating Demand for Mobile Application, *Proceedings of 2012 AppWeb Workshop*, Lyon, France, Ghose, A., and Ipeirotis, P. G. 2011. Estimating the Helpfulness and Economic Impact of Product Reviews: Mining Text and Reviewer Characteristics, *IEEE Transactions on Knowledge and Data Engineering*, 23(10):1498–1512.

Glorot, X., Bordes, A., and Bengio, Y. 2011. Domain Adaptation for Large-scale Sentiment Classification: A Deep Learning Approach, *Proceedings of 28th International Conference on Machine Learning (ICML' 11)*, Bellevue, Washington, USA, 513–520.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. 2009. The WEKA Data Mining Software: An Update, *SIGKDD Explorations*, 11(1):10–18.

Hatzivassiloglou, V., and Wiebe, J. 2000. Effects of Adjective Orientation and Gradability on Sentence Subjectivity, *Proceedings of 18th International Conference on Computational Linguistics (COLING' 00),* Saarbrücken, Germany, 174–181.

He, Y., Lin, C., and Alani, H. 2011. Automatically Extracting Polarity-Bearing Topics for Cross-Domain Sentiment Classification, *Proceedings of 49th Annual Meeting of Association for Computational Linguistics (ACL'11)*, Portland, USA, 123–131.

Henze, N., and Boll, S. 2011. Release Your App on Sunday Eve: Finding the Best Time to Deploy Apps, *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, New York, NY, USA, 581–586.

Hevner, A., and Chatterjee, S. 2010. *Design Research in Information Systems:Theory and Practice, Integrated Series in Information Systems*, (Vol. 22) :Springer.28–31.

Hevner, A., March, S., Park, J., and Ram, S. 2004. Design Science in Information Systems Research, *MIS Quarterly*, 28(1):75–105.

Hinton, G. 2002. Training Products of Experts by Minimizing Contrastive Divergence, *Neural Computation*, 14(1):1771–1800.

Hoberg, G., and Phillips, G. M. 2013. Text-Based Network Industries and Endogenous Product Differentiation,

Hu, M., and Liu, B. 2004. Mining and Summarizing Customer Reviews, *Proceedings of the 10th ACM Conference on Knowledge Discovery and Data Mining (KDD' 04)*, Seattle, Washington, USA, 168–177.

Huang, K.-W., and Li, Z. 2011. A Multi-Label Text Classification Algorithm for Labeling Risk Factors in SEC Form 10-K, *ACM Transactions on Management Information Systems*, 2(3):18:1–18:19.

Jindal, N., and Liu, B. 2006. Identifying Comparative Sentences in Text Documents, *Proceedings of the 29th International ACM Conference on Research and Development in Information Retrieval (SIGIR' 06)*, Seattle, Washington, USA, 244 – 251.

Jindal, N., and Liu, B. 2008. Opinion Spam and Analysis, *Proceedings of the 1st Conference on Web Search and Web Data Mining (WSDM' 08)*, Palo Alto, California, USA, 219–230.

Joachims, T. 1998. Text Categorization with Support Vector Machines: Learning with Many Relevant Features, *Proceedings of the 10th European Conference on Machine Learning (ECML' 98)*, Chemnitz, Germany, 137–142.

Kahle, K., and Walkling, R. 1996. The Impact of Industry Classifications on Financial Research, *Journal of Financial and Quantitative Analysis*, 31(3):309–335.

Kajanan, S., Pervin, N., Narayan, R., Datta, A., and Dutta, K. 2012. Takeoff and Sustained Success of Apps in Hypercompetitive Mobile Platform Ecosystems: An Empirical Analysis, *In Proceedings of 2012 International Conference on Information Systems (ICIS)*, Orlando, Florida, USA,

Kent, J. 2012. Free for All: In-app Purchases to Dominate Smartphone App Business,

Kim, R. 2012. Appsfire Scores $3.6M As App Discovery Demands Grow,

Kullback, S., and Leibler, R. 1951. On Information and Sufficiency, *Annals of Mathematical Statistics*, 22(1):79–86.

Larochelle, H., and Bengio, Y. 2008. Classification using Discriminative Restricted Boltzmann Machines, *Proceedings of 25th International Conference on Machine Learning (ICML' 08)*, Helsinki, Finland, 536–543.

Lee, C., Ma, P., and Wang, C. 2012. Identifying Peer Firms: Evidence from EDGAR Search Traffic,

Li, S., Lin, C.-Y., Song, Y.-I., and Li, Z. 2010. Comparable Entity Mining from Comparative Questions, *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL' 10)*, Uppsala, Sweden, 650 – 658.

Lin, J. 1991. Divergence Measures Based on the Shannon Entropy, *IEEE Transactions on Information Theory*, 37(1):145–151.

Liu, B. 2010. Sentiment Analysis and Subjectivity, *Handbook of Natural Language Processing, Second Edition*, Chapman and Hall.1–38.

Liu, B., Hu, M., and Cheng, J. 2005. Opinion Observer: Analyzing and Comparing Opinions on the Web, *Proceedings of 14th World Wide Web Conference (WWW' 05)*, Chiba, Japan, 342–351.

Liu, K., and Zhao, J. 2009. Cross-Domain Sentiment Classification Using a Two-Stage Method, *Proceedings of 18th ACM Conference on Information and Knowledge Management (CIKM'09)*, Hong Kong, China, 1717–1720.

Manninen, J. 2012. Mobile app revenue will hit $15B in 2011,

Mejova, Y., and Srinivasan, P. 2012. Crossing Media Streams with Sentiment: Domain Adaptation in Blogs, Reviews and Twitter, *Proceedings of the 6th International AAAI Conference on Weblogs and Social Media*, Dublin, Ireland, 234 – 241.

Michiels, S., Koscielny, S., and Hill, C. 2005. Prediction of Cancer Outcome with Microarrays: A Multiple Random Validation Strategy, *Lancet*, 365(9458):488–492.

Mitchell, T. 1997. *Machine Learning*, McGraw-Hill.

Newman, M. E. J. 2005. Power Laws, Pareto Distributions and Zipf's Law, *Contemporary Physics*, 46(5):323–351.

Pan, S. J., Ni, X., Sun, J.-T., Yang, Q., and Chen, Z. 2010. Cross-Domain Sentiment Classification via Spectral Feature Alignment, *Proceedings of 19th International World Wide Web Conference(WWW'10)*, Raleigh, USA, 26–30.

Pang, B., and Lee, L. 2008. Opinion Mining and Sentiment Analysis, *Foundations and Trends in Information Retrieval*, 2(1):1–135.

Pang, B., Lee, L., and Vaithyanathan, S. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques, *Proceedings of 2002 Conference on Empirical Methods on Natural Language Processing (EMNLP' 02)*, Philadelphia, USA, 79–86.

Peddinti, V., and Chintalapoodi, P. 2011. Domain Adaptation in Sentiment Analysis of Twitter, *Proceedings of 2011 AAAI Workshop on Analyzing Microtext*, San Francisco, USA, 44 – 49.

Phan, X.-H., and Nguyen, C.-T. 2007. GibbsLDA++: A C/C++ Implementation of Latent Dirichlet Allocation,

Podolyan, Y., Walters, M. A., and Karypis, G. 2010. Assessing Synthetic Accessibility of Chemical Compounds Using Machine Learning Methods, *Journal of Chemical Information and Modeling*, 50(6):979–991.

Press, W., Teukolsky, S., Vetterling, W., and Flannery, B. 2007. Support Vector Machines, *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press.

Riloff, E., and Wiebe, J. 2003. Learning Extraction Patterns for Subjective Expressions, *Proceedings of 2003 Conference on Empirical Methods on Natural Language Processing (EMNLP' 03),* Sapporo, Japan, 25–32.

Rui, H., and Whinston, A. 2011. Designing a Social-Broadcasting-Based Business Intelligence System, *ACM Transactions on Management Information Systems*, 2(4):Article 22.

Shmueli, G., and Koppius, O. 2011. Predictive Analytics in Information Systems Research, *MIS Quarterly*, 35(3):553–572.

Smolensky, P. 1986. Information Processing in Dynamical Systems: Foundations of Harmony Theory, *Parallel Distributed Processing: Explorations in the Microstructures of Cognition*, MIT Press.194–281.

Tarca, A. L., Carey, V. J., Chen, X., Romero, R., and Drăghici, S. 2007. Machine Learning and Its Applications to Biology, *PLOS Computational Biology*, 3(6):e116.

Titov, I. 2011. Domain Adaptation by Constraining Inter-Domain Variability of Latent Feature Representation, *Proceedings of 49th Annual Meeting of Association for Computational Linguistics (ACL'11)*, Philadelphia, Pennsylvania, USA, 417–424.

Turney, P. 2002. Thumbs Up or Thumbs Down? Semantic OrientationApplied to Unsupervised Classification of Reviews, *Proceedings of 40th Annual Meeting of Association for Computational Linguistics (ACL'02)*, Portland, USA, 62–71.

Venturedata. 2012. Apple Updates App Store Application Ranking Algorithm: Downloads a Greater Impact,

Wiebet, J., and Bruce, R. 1999. Development and Use of a Gold Standard Data Set for Subjectivity Classifications, *Proceedings of 27th Annual Meeting of the Association for Computational Linguistics (ACL' 99)*, College Park, Maryland, USA, 264–253.

Xu, K., Liao, S. S., Li, J., and Song, Y. 2011. Mining Comparative Opinions from Customer Reviews for Competitive Intelligence, *Decision Support Systems*, 51(4):743–754.