A Multilabel Text Classification Algorithm for Labeling Risk Factors in SEC Form 10-K

KE-WEI HUANG and ZHUOLUN LI, National University of Singapore

This study develops, implements, and evaluates a multilabel text classification algorithm called the multilabel categorical K-nearest neighbor (ML-CKNN). The proposed algorithm is designed to automatically identify 25 types of risk factors with specific meanings reported in Section 1A of SEC form 10-K. The idea of ML-CKNN is to compute a categorical similarity score for each label by the K-nearest neighbors in that category. ML-CKNN is tailored to achieve the goal of extracting risk factors from 10Ks. The proposed algorithm can perfectly classify 74.94% of risk factors and 98.75% of labels. Moreover, ML-CKNN is empirically shown to outperform ML-KNN and other multilabel algorithms. The extracted risk factors could be valuable to empirical studies in accounting or finance.

Categories and Subject Descriptors: H.4.2 [**Information Systems Applications**]: Types of Systems—*Decision support*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Text analysis*; I.5.1 [**Pattern Recognition**]: Models—*Deterministic*; I.5.2 [**Pattern Recognition**]: Design Methodology—*Classifier design and evaluation*; I.5.4 [**Pattern Recognition**]: Applications—*Text processing*; J.1 [**Computer Applications**]: Administrative Data Processing—*Financial*; K.6.0 [**Management of Computing and Information Systems**]: General—*Economics*

General Terms: Algorithms, Performance, Management

Additional Key Words and Phrases: Text classification; text mining; multilabel classification; risk factors; annual reports

ACM Reference Format:
Huang, K.-W. and Li, Z. 2011. A multilabel text classification algorithm for labeling risk factors in SEC form 10-K. ACM Trans. Manag. Inform. Syst. 2, 3, Article 18 (October 2011), 19 pages.
DOI = 10.1145/2019618.2019624 http://doi.acm.org/10.1145/2019618.2019624

1. INTRODUCTION

It is widely recognized that corporate annual reports play a key role in financial markets. Corporate annual reports are regarded as one of the most important sources of information about a company. Disclosures in annual reports enable banks, institutional investors, central banks, and upstream suppliers or downstream distributors of the target firm to make more accurate decisions.

This study develops an algorithm for labeling risk factors reported in SEC form 10-K. Form 10-K is the report filed annually published by U.S. companies 90 days after the end of each fiscal year. These forms are much more detailed than the annual reports sent to shareholders. Since 2005, companies are required to report risk factors in a separate section (Section 1A) in form 10-K. The definition and examples of risk factors are provided in Section 4. In plain language, risk factors describe what could go wrong, likely external negative factors, possible future failures to meet any obligations, and any other

Authors' addresses: K.-W. Huang (corresponding author) and Z. Li, Department of Information Systems, School of Computing, National University of Singapore, Singapore; email: huangkw@comp.nus.edu.sg.

DOI 10.1145/2019618.2019624 http://doi.acm.org/10.1145/2019618.2019624

risks that should be disclosed to adequately warn investors. Because of the regulation change, risk factors have become a well-defined list of bullets; this provides an excellent research opportunity for applying text classification to label each bulleted factor.

This study develops a multilabel text classification algorithm that can identify 25 types of risk factors. For example, this algorithm can identify risk factors that indicate companies that face intense competition, have difficulty in managing mergers and acquisitions, or rely on few large customers. In Section 1A, risk factors are reported as a list of bullets. Each bullet is the unit of analysis for classification in this study. Since a company can write anything under each bullet, a bullet could convey information that belongs to more than one label. Therefore, we need to apply multilabel rather than single-label classification algorithms in this domain.

There are three major contributions of this study. First, classifying risk factors is an important research topic in accounting and finance (see Section 2.1). Second, this article demonstrates a novel application of text classification. Most existing text mining applications, especially those in finance or accounting, classify the target document into two or three labels with imprecise information. For example, most existing studies categorize an article or a section as positive or negative. The present study classifies risk factors into 25 types, each with its own unique meaning in accounting. Therefore, this approach can produce more concrete information than the existing literature in this domain. The same idea could be applied to extract valuable information from contracts, patents, analysts' reports, or other structured documents. Third, a new multilabel text classification algorithm named the multilabel categorical K-nearest neighbor (ML-CKNN) is proposed and implemented to label risk factors. ML-CKNN is similar to KNN [Cover and Hart 1967] and ML-KNN [Zhang and Zhou 2007]. The main idea of ML-CKNN is to identify the K-nearest neighbors in each category; this is the main departure point from the ML-KNN, which identifies the K-nearest neighbors in the full training set. These top-K records are used to compute a *categorical similarity score* between the sample and the target category. If this categorical similarity is larger than a threshold, then that sample is labeled as 1 for that type. In sharp contrast with ML-CKNN, the idea of ML-KNN is to compute a posterior probability that a label could be 1, using the label distribution of K-nearest neighbors. Detailed comparisons are provided in Sections 3.3 and 5.3. We show that ML-CKNN performs better than ML-KNN and other classic algorithms in this domain. The proposed algorithm can perfectly classify 74.94% of risk factors and 98.75% of labels. The intuition behind the excellent performance of ML-CKNN is that we only use few samples in each category to decide whether the target sample should be labeled as 1 for that category. This works well when the true positive samples in the same category may have very different word combinations, which is also the main feature of classifying risk factors.

The remainder of this article is organized as follows. Section 2 provides the literature review. Section 3 discusses ML-CKNN. Section 4 defines the risk factors and addresses data collection issues. Section 5 reports the empirical performance of ML-CKNN and several other algorithms. Section 6 concludes this work.

2. LITERATURE REVIEW

The present study is most closely related to two streams of literature. The current article demonstrates a new approach to apply computational methods to quantify textual information in finance or accounting. Moreover, our research could add to multilabel text classification studies.

2.1. Computational Text Analysis In Accounting And Finance

Recently, accounting researchers have started to apply various computational methods, especially content analysis, to quantify textual information in financial statements. In

an award-winning paper in accounting, Li [2008] studies the relationship between corporate earnings and the readability of 10-K filings. The author used two variables to measure readability: (1) the Fog Index, which is computed based on average sentence length and complex words with three or more syllabi, and (2) the length of the 10-K itself. The author found that companies with less readable (i.e., higher Fog Index and longer in length) 10-Ks have lower earnings (refer to Li [2011], for a complete survey). A short list of examples is provided in this section.

Following Li [2008], there is a surge in the number of studies that use tone, sentiment, or readability to investigate various issues in accounting. For example, Feldman et al. [2009] classify words into positive and negative categories in order to measure the tone change in the management discussion and analysis (MD&A) section of 10-Q and 10-K. The authors find that stock market reactions around the SEC filing are significantly associated with the tone change of the MD&A section, even after controlling for accruals and earnings surprises. Kothari et al. [2009] find that when content analysis indicates favorable disclosures in all media channels, the firm's risk declines significantly. Loughran and McDonald [2010] improve the readability used in Li [2008] for 10-K. Campbell et al. [2011] examine the information content of the risk factor section. By the percentage of key words associated with different types of risk in the risk factor section, they show that the type of risk that a firm faces (i.e., systematic, idiosyncratic, financial, legal, or tax) determines whether they devote a greater portion of their disclosures toward describing that risk type.

There is a similar trend of using content analysis techniques in the finance literature. Tetlock [2007] studies the relationship between the content of *Wall Street Journal* and the stock market. The independent variable is a sentiment index calculated by the number of positive words and negative words categorized by the General Inquirer (GI), a well-known dictionary used by psychologists. This study finds that media pessimism induces downward pressure on market prices. Following this study, Tetlock et al. [2008] quantify financial news stories by the same measure to predict firms' accounting earnings and stock returns. A recent paper by Loughran and McDonald [2009] shows that negative words categorized by the GI may not have negative meaning in 10-K. A new word classification is developed; the study shows that negative words by the new classification are more informative than those in GI.

Applying text classification to study finance or accounting issues is still in its nascent stage. Earlier studies focused on stock price prediction. Antweiler and Frank [2004] classify messages posted on Yahoo! Finance using naïve Bayes and Support Vector Machine (SVM). Individual messages were classified as bullish, bearish, or neutral. They found that messages can help to predict market volatility but not expected return. Similarly, Das and Chen [2007] classify Yahoo! Finance messages into the same three types using five existing algorithms. The overall evidence suggests that market activity is related to message board activity. Balakrishnan et al. [2010] classify each 10-K into three classes: outperforming, average, and underperforming. They show that this model captures information not contained in document-level readability. Recently, researchers started to become interested in studying the impacts of tones in the financial statement. Li [2010] utilizes naïve Bayes to classify the tone of forward-looking statements and finds that a change in the tone implies a future change in the performance of the company. Mangen and Durnev [2010] study the tone in restatement announcements and find that restatement tone affects restating firms' and their competitors' abnormal returns. Hanley and Hoberg [2010] study the information content of IPO prospectuses and IPO pricing. The authors find that MD&A most contributes to the informativeness of the prospectus.

In the information systems literature, Schumaker and Chen [2009] develop a stock trading system using quantitative portfolio selection strategies and news articles

quantified by SVM. The proposed system is shown to produce superior stock trading return in a short period of time. Gu et al. [2007] examine how users value virtual communities about investment. In this study, a key independent variable is the quality of messages, which are classified as signal, noise, or neutral. Bai et al. [2010] study the data quality risk in accounting information systems.

2.2. Comparisons

The present study is different from the aforementioned research in several aspects. First, and most importantly, the core methodology for quantifying textual information is different. We summarize four potential methods as follows: (1) content analysis by using packaged software, (2) keyword counting, (3) text classification, and (4) information extraction by Natural Language Processing (NLP). The easiest approach for conducting computational text analysis is to use packaged content analysis software. These software applications can generate readability variables, word frequencies, and other statistics for researchers to conduct further studies. Ease of use is the major benefit; this is the most popular approach adopted in social science studies. The second easiest approach is to categorize keywords and then count the number of keywords in each category. Next, researchers transform categorical word counts to numeric variables by a mathematical function that fits their research needs. This method is not difficult to implement in practice and may work well in some domains. The third approach is to use standard text classification techniques to transform textual information into several labels. Theoretically, this approach could be more powerful than the previous two methods. However, performance depends critically on the property of labels and the keyword distribution. Roughly speaking, text classification performs better when labels are correlated with the occurrence of a small set of keywords. Lastly, the most technologically advanced method is to use Natural Language Processing (NLP) tools to extract the semantics of the target textual information. Unfortunately, most existing studies in finance or accounting do not use this approach, because it requires demanding knowledge of NLP. Our study provides a novel text classification approach that can generate outputs with valuable information similar to those produced by NLP tools.

Moreover, the unit of analysis in this study is the individual risk factor, which has not been investigated in the literature. In the existing literature, the unit of analysis includes the whole news article, the whole SEC forms, one section (mostly the MD&A section) of 10-K, one sentence in the MD&A section, or one posting on the Internet forum. Using risk factors as the unit for text classification is an important improvement for accounting studies, because individual risk factors have different meanings. Also, each bullet of risk factors could convey more sophisticated information than one section or the whole annual report.

Furthermore, the output label of the current study is more informative. The output of the present study is 25 labels, each of which represents a specific meaning of risk factors. In sharp contrast, existing text classification applications typically have a very small number of labels. The most common labels are positive, negative, and neutral. These three labels could convey limited information. Our empirical study shows that text classification is a good fit for the risk factors classification: ML-CKNN can achieve very high accuracy even when labeling 25 classes.

Although existing studies in accounting or finance only use single-label classification, risk factors poses a multilabel classification problem. To the best of our knowledge, our work is the first multilabel application in this domain. To tackle this challenge, we propose a new algorithm that is tailored for risk factor classification and is empirically shown to outperform ML-KNN on this task.

Table I. Summary of the Text Analysis Methods Used in Accounting and Finance Literature

Author-Year	Unit of Analysis	Method	Main Output Variable
Current Paper	10-K: Individual Risk factors	A new multi-label text classifier	25 types of risk factors.
Campbell et al. (2011)	10-K: Risk Factor Section	Words categorization	5 types of risks
Li (2008)	10-K	Content Analysis	Readability. No label.
Li (2010)	10-K: Sentences in Item 7 MD&A	Single-label classifier	3 labels
Feldman et al. (2009)	10-K: Words in Item 7 MD&A	Words categorization	Word counts in 2 labels
Balakrishnan et al. (2010)	10-K: MD&A Section	Single-label classifier	3 labels
Hanley and Hoberg (2010)	4 Sections in IPO prospectuses	Single-label classifier	Cosine-similarity
Antweiler and Frank (2004)	Postings on Yahoo! Finance	Single-label classifier	3 labels
Das and Chen (2007)	Postings on Yahoo! Finance	Voting of single-label classifiers	3 labels
Tetlock (2007)	Finance news articles	Words categorization	A sentiment measure by positive and negative word counts
Tetlock et al. (2008)	Finance news articles	Words categorization	A sentiment measure by positive and negative word counts
Loughran and McDonald (2009)	10-K	Words categorization	A sentiment measure by positive and negative word counts

2.3. Multilabel Text Classification

Recently, much research has been done on multilabeled classification. A straightforward solution for multilabel classification algorithms is to transform the multilabel classification task into several binary single-label classification problems [Tsoumakas et al. 2010]. Specifically, a multilabel problem with Q labels can be decomposed into Q binary classification problems. In other words, we use a binary classifier to predict whether a record is 0 or 1 label-by-label. All standard text classification algorithms can be applied to solve a multilabel classification problem using this approach. We evaluate the performance of this approach in Section 5.3.

A second approach is to consider all possible instances as an individual type of label [Tsoumakas et al. 2010]. For example, if there are only three labels in a multilabel problem, we have 2^3 possible values of the output multilabel vector (i.e., (000), (001), (010). . .etc.). Next, we can apply any standard single-label algorithm with eight classes to solve this problem. This approach does not fit our task because we have 25 labels, leading to a single-label problem with 2^{25} classes.

The third approach is to extend a specific learning algorithm to handle multilabel data directly. Among all multilabel algorithms, our new algorithm is most similar to ML-KNN [Zhang and Zhou 2007], which is an adaptation of the KNN lazy learning algorithm for multilabel data. Similar to KNN, ML-KNN first finds the K-nearest neighbors of the target sample. Next, it calculates the posterior probability that a label is 0 or 1 by applying Bayes rules to the distribution of labels of the K-nearest neighbors. A record is labeled as 1 if the posterior probability that a label being 1 is greater than a threshold. Other multilabel classification algorithms include: boosting [Schapire and Singer 2000], support vector machine [Elisseeff and Weston 2002], decision tree [De Comité et al. 2003], neural network [Zhang and Zhou 2006], logistic regression [Cheng and Hüllermeier 2009], and hierarchy extraction [Brucker et al. 2010]. The proposed ML-CKNN could contribute to this line of studies by providing a simple new route.

Given a training set S, an unlabelled input x_t, and the output vector Y_t.
There are two input parameters: K and T, where $K \in \mathbb{N}$ & $T \in \mathbb{R}$.
% Step 1: Calculate the similarity between x_t and each record in S
(1) for all records in S, $(\forall x_j \in X)$.
(2) calculate a similarity metric denoted by SIM(x_t, x_j)
(3) end of for
% Step 2: use the K-nearest-neighbor in each category to represent the similarity of that category. f(x_t,i) denotes the similarity of category i.
(4) for i = 1 to Q
(5) f(x_t,i) =average (SIM(x_t, x_j)),
(6) where x_j is among the K-nearest neighbors of which ith element of Y_j is 1.
(7) end of for
% Step 3: create the final label vector Y_t.
(8) for i = 1 to Q
(9) if f(x_t,i) >= T, set the ith element of Y_t is set to 1.
(10) end of for

Fig. 1. Pseudocode of ML-CKNN.

3. A MULTILABEL TEXT CLASSIFICATION ALGORITHM

3.1. The Algorithm: ML-CKNN

To be consistent with the literature, we follow the notations used by Zhang and Zhou [2007]. Let X denote the domain of a document (i.e., word vectors of risk factors) and Y be a finite set of labels. The number of labels is denoted by Q. In our main classification task, there are 25 types of risk factors. Therefore, $Q = 25$, and the size of Y is 2^Q. Each element of Y is denoted by Y_j and is a binary vector with 25 elements. The ith element of Y_j is 1 when the target training record is labeled positive for type i. Each labeled risk factor in the training set is represented by $S = \{(x_1, Y_1), (x_2, Y_2), \ldots, (x_m, Y_m)\}$, ($x_j \in X$, $Y_j \in Y$). For example, x_j is the value of the word vector that represents a specific document, while Y_j is a binary vector that represents the true labels. As a consequence, a classifier in this study is defined as $H{:}X \rightarrow Y$.

The pseudocode of ML-CKNN is reported in Figure 1. Explanations are provided as follows.

Step 0: TF-IDF [Han and Kamber 2006] is used to create a customized word vector of each risk factor.[1] The final word vector includes 1,430 composite words of one or two keywords.

[1]This study uses RapidMiner, an open-source data-mining application based on Weka, to create the customized word vector. RapidMiner's built-in filters, include "English Stop Word List Filter", "Token Length Filter (>= 4)", "Porter Stemmer", and "Terms-2-Gram Generator", are used to create a candidate word vector. The word vector produced by RapidMiner contains 16,353 composite words. Next, words with low occurrences (low TF) or low informative value are dropped. Specifically, we drop all words with less than three occurrences in the training set. For words less than ten occurrences in the training set, we only keep the words with more than 75% conditional frequency in one type/category. In other words, only keywords with high predictive power are kept: that is, when it shows up, with more than 75% the target risk factor belongs to a specific category in the training set.

Step 1: For each pair of a test record and a training record, we calculate the cosine similarity, the most widely adopted distance metric in text mining [Han and Kamber 2006]. Clearly, other similarity/distance metrics could be used in this step.
Step 2: The proposed algorithm classifies each sample as 0 or 1 type-by-type 25 times. Among training records labeled as 1 in type i, we identify the K-nearest neighbors. We use the average of the top-K similarity scores as the similarity of type i. We define this number as the categorical similarity.[2]
Step 3: For each type, if the categorical similarity is greater than a threshold T, the test record is labeled as 1 for that type; otherwise, it is labeled as 0.

There are two parameters in the aforementioned algorithm: (1) using top-K records to compute each category's similarity and (2) a threshold T that decides the smallest similarity score for setting the label to 1. Section 5.2 discusses how to optimize the performance by K and T. Different threshold values can be used for different types in general. To avoid overfitting issues and to be conservative about the performance of our algorithm, we use the same K and T for all labels. The other reason is that one important goal of this study is to compare the performance between ML-CKNN and ML-KNN. Using two parameters ensures a fair comparison between these two algorithms. We elaborate on how to set different K and T values for different types in Section 3.3.3.

3.2. The Complexity Analysis

The running time of this algorithm is given as follows.

$$N_{Test} \times (t_T \times N_{Training} + Q \times t_S),$$

where N_{Test} is the size of the test set, t_T denotes time for computing similarity, $N_{Training}$ represents the size of the training set, Q is the number of categories, and t_S means time for sorting similarity to find the K-nearest neighbors in each category. The running time of this algorithm is $O(n_i \log(n_i))$, where n_i is the number of records in category i.

First, the running time of our algorithm is linear in the size of test set (N_{Test}) and each test record can be processed in parallel and independently. Second, for each test record, there are two time-consuming computation steps: time for computing similarity and time for sorting to find the K-nearest neighbors. Again, both tasks are linear in $N_{Training}$ and can be processed in parallel and independently. Sorting time in each category is also manageable.

3.3. Strengths and Weakness of ML-CKNN

The main idea of ML-CKNN is to handle two challenging properties when classifying risk factors. First, correct samples in the same category may have very different word vectors. Also, some correct samples could be rare in the population of the training set. For example, the following two risk factors all represent Type 1 financial condition risks, but the wordings are completely different.

"We have experienced net losses, and we may not be profitable in the future."
"The company has significant leverage."

As a consequence, we should not use all training records to calculate the categorical similarity. Only records that are very similar to our test record should be used to compute the categorical similarity.

[2]"Mean of top-K similarity" is the most intuitive choice in this context. For future studies, it may be interesting to use other statistics, such as the median of top-K similarity or other percentiles of all training records in category *i*.

Second, some risk factors types are characterized by keywords that are very common in other types as well. For example, keywords such as demand, customers, or uncertain appear in several types. If we use ML-KNN, the K-nearest neighbors may have positive labels in several types, leading to high false positive results. By ML-CKNN, we can correctly label this kind of records with appropriate K and T values for computing the categorical similarity. Our empirical result in Section 5.2 shows that the accuracy of ML-CKNN is quite sensitive in T and also confirms that ML-CKNN can significantly outperform ML-KNN. We summarize other benefits of ML-CKNN in the next sections.

3.3.1. Simplicity. Like other instance-based algorithms such as KNN and ML-KNN, ML-CKNN is very easy to implement and to interpret. The idea of ML-CKNN is to compute a similarity for each category, which is very intuitive compared with those conceptually complicated algorithms such as SVM.

3.3.2. High Scalability in the Number of Labels and in the Size of Training Set. For multilabel classification, the idea behind ML-CKNN is especially appealing, as we may face a task with a large number of labels. This claim is particularly conspicuous when we compare ML-CKNN with ML-KNN. ML-KNN needs to identify the K-nearest neighbor in all records, whereas ML-CKNN needs to identify the K-nearest neighbor in each category. First, ML-CKNN clearly enjoys better scalability, because ML-CKNN only needs to sort 1/Q of the training set. Also, ML-CKNN can compute its top K records in each category in parallel, a fact leading to better scalability along the number of categories (labels). This feature enhances extensibility: researchers can add more labels later without affecting the existing training results of ML-CKNN.

3.3.3. Transparency of parameters and the associated adaptability. As explained in Section 3, the meaning of K and T are intuitive. In practice, ML-CKNN can set a different K and T for each risk factor. This could be an important feature for improving performance. This feature allows researchers to set a higher K in categories with more homogeneous correct samples and a smaller K in categories with heterogeneous correct samples. The value of T depends on the nature for each label. For instance, some risk factor types may have a one-to-one strong correlation with certain keywords, so we can set a larger cutoff for T. Other risk factors types could be more difficult to classify, because word counts have a weak correlation with that label; in this case, we should set a smaller T.

3.3.4. Shortcomings. The main weakness is that ML-CKNN ignores the interdependencies between different labels, because categorical similarity is computed independently. If there are any correlations between labels that are not reflected in the correlation of word occurrences, then ML-CKNN may perform worse in terms of accuracy.

The other weakness is that the training sample size and T could not be too small. The problem of ML-CKNN is that if few positive samples resemble the target test record, then it will be labeled as 1. It is highly possible that few samples may have high similarity because of luck or labeling error. If the training set or T is small, then the performance of the algorithm will deteriorate.

The total running time is not a strength of ML-CKNN. The most time-consuming step is the identification of the K-nearest neighbors in each category. From our experiment, performance in terms of speed is worse than that of the decision tree or naïve Bayes but faster than that of SVM or neural network. In short, the speed of ML-CKNN could be considered as an average case among multilabel classification algorithms.

4. EXPERIMENT: CLASSIFYING RISK FACTORS IN 10-K

4.1. SEC Form 10-K and Risk Factors

The source of risk factors is collected from SEC form 10-K filings of all publicly listed companies in the U.S. These filings are publicly available from The Electronic Data Gathering, Analysis and Retrieval (EDGAR) database on the Internet. In this study, 6,208 firms' 21,077 10-K files from January 1st in 2006 to May 31st in 2010 are collected in HTML format from EDGAR. The starting year is 2006, because companies have been required to report risk factors in a separate section (Item 1A – Risk Factors) in their annual reports since 2006. According to Item 503(c) of Regulation S-K (229.503(c) of this chapter), "risk factors" are defined as follows.

> "Where appropriate, provide under the caption "Risk Factors" a discussion of the most significant factors that make the offering speculative or risky. This discussion must be concise and organized logically. Do not present risks that could apply to any issuer or any offering. Explain how the risk affects the issuer or the securities being offered. Set forth each risk factor under a subcaption that adequately describes the risk. The risk factor discussion must immediately follow the summary section."

For example, the headings of the first five risk factors in the 2010 annual report of Oracle are as follows.

(1) "Economic, political, and market conditions, including the recent recession and global economic crisis, can adversely affect our business, results of operations and financial condition, including our revenue growth and profitability, which in turn could adversely affect our stock price."
(2) "We may fail to achieve our financial forecasts due to inaccurate sales forecasts or other factors."
(3) "We may not achieve our financial forecasts with respect to our acquisition of Sun or our entrance into a new hardware systems business, or the achievement of such forecasts may take longer than expected. Our profitability could decline if we do not manage the risks associated with our acquisition and integration of Sun."
(4) "Our success depends upon our ability to develop new products and services, integrate acquired products and services and enhance our existing products and services."
(5) "Our strategy of transitioning from Sun's indirect sales model to our mixed direct and indirect sales model may not succeed and could result in lower hardware systems revenues or profits. Disruptions to our software indirect sales channel could affect our future operating results."

These risk factors are reported as a list of bullets with detailed explanations appearing immediately after each bullet (which do not appear here for brevity). In this study, text classification is applied to the headings, but not to the descriptions, text classification on both headings and descriptions led to worse performance in pilot studies.[3]

Most companies report 15 to 60 risk factors in each annual report. For the same company, risk factors are quite similar across years. Typically, companies prepare 10-Ks by adding one to five risk factors to the previous year's 10-K after reordering some risk factors.

One limitation of this study is the use of computer programs to parse and separate risk factor bullets in HTML files. Form 10-Ks are reported in nonstandardized HTML files. Without the regulation change in 2005, it is almost impossible to locate and extract all risk factors using software programs. Even if risk factors were summarized

[3]The reason could be that descriptions include many more words than headings. Therefore, the same keyword may show up in several types' descriptions, reducing the power of that keyword to identify a specific type of risk factors.

in Section Item 1A after 2006, it remains a challenging task to extract all risk factors, because companies use very different HTML languages to represent the beginning of the target section. Furthermore, extracting each bullet of risk factors and separating headings of risk factors from other parts is even more challenging. Companies may use all kinds of HTML languages to represent a bullet point of risk factor and can insert subcaptions to represent a group of similar risk factors. For example, firms may insert "risks related to our products" before three to five risk factors, adding another layer of complexity to the parsing program. Some companies have violated the accounting rules to report risk factors in other sections or in completely different formats. In the end, roughly 75% of all 10-K forms and 500,051 risk factors were collected.

4.2. Training Set and the Risk Factors Labeling

As the first step for implementing the supervised learning algorithm, the researchers read hundreds of annual reports to subjectively identify 25 types of risk factors. These risk factors types are explained in detail in Table III along with examples of risk factors. This list of risk factors is extensive but is not exhaustive, because in practice, companies may report any type of risk factor. Some risk factors are very firm-specific, and similar factors appear in very few 10-Ks. In general, there may exist hundreds or even thousands types of risk factors, but this study focuses on a small set of common risk factors. For the convenience of coding risk factors, the definitions of risk factors are chosen to be mutually exclusive and relatively well-defined. We also refer to the existing literature about manually coding risk factors for other issues [Abdou and Dicle 2007; Nelson and Pritchard 2007]. Clearly, this step involves the researchers' subjective judgment: some important risk factor types may be left out in this study. However, missing a few important categories does not affect the absolute performance of ML-CKNN (as explained in Section 3.3.2) and also does not affect the performance comparisons in Section 5.

Four student researchers in information systems were recruited to label 10,000 risk factors in fiscal year 2007. These four students are native English speakers who have taken financial accounting. Since risk factor types have concrete definitions and each risk factor is reported in plain English with detailed explanations in its body, labeling risk factors does not involve professional accounting knowledge. These 10,000 risk factors were chosen from the 10-K forms of the 800 largest companies in 2007 in terms of total asset value. Students were briefed on the definitions of risk factor types and given some labeled examples to learn the exact definition of risk factors. The year 2007 was chosen, because it is the middle year of the sample period. The training set was built by one year's data, because firms may report identical risk factors in different years. In the extreme case, firms may report an identical risk factor five years in a row.

Each student labeled 5,000 risk factors. In other words, each risk factor was labeled by two students. A user-friendly Web interface was developed for them to label risk factors. They were given four weeks to label 5,000 risk factors and finished the task between two to four weeks. The final training set was created using the records that two students completely agreed on all labels. Because of this sample bias, this training set could be an easier set of risk factors for human coders and also for text classification algorithms. In other words, the accuracy of our experimental results in Section 5 could be overestimated. However, this bias does not affect the relative performance of our algorithm. The final training set included 3,153 risk factors from 4,267 companies' 92,993 risk factors in 2007.

As another example of risk factors labeling, the first five risk factors of Oracle mentioned in the preceding section are labeled as Type 8, Type 22, Type 4, Type 24, and Type 25, respectively. In the Oracle examples, all five risk factors are labeled as a single type. In general, one risk factor could be labeled as multiple types.

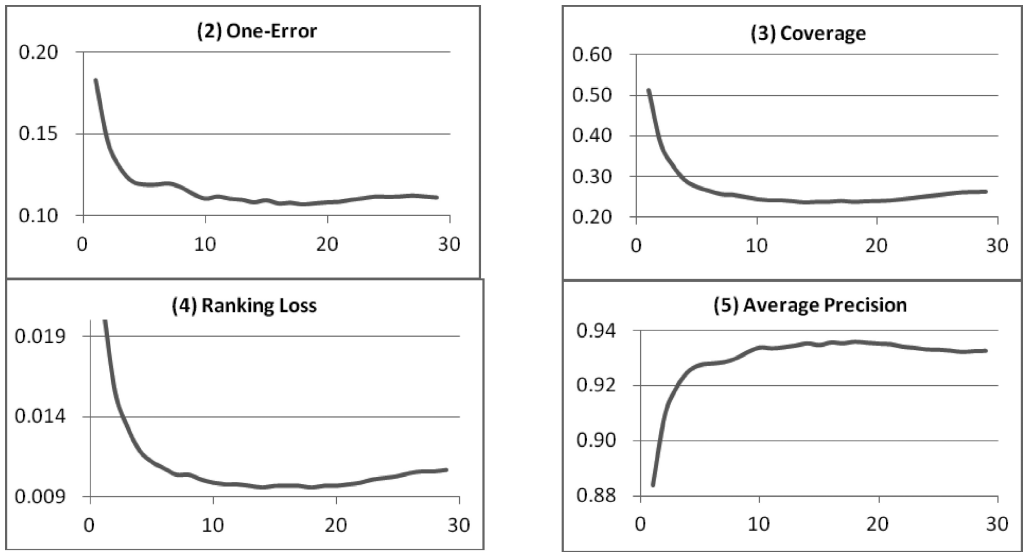

Fig. 2. Performance of ML-CKNN for different values of K.

5. EMPIRICAL EVALUATION AND PERFORMANCE COMPARISONS

5.1. Evaluation Criteria

Following the literature in multilabel classification [Zhang and Zhou 2007], we evaluate ML-CKNN for accuracy and five metrics for multilabel classification. Results are reported in Figure 2. Please refer to Zhang and Zhou [2007] for the formal definitions of these measures. Intuitions of these metrics are provided as follows.

(1) Hamming loss: this measure evaluates how many times an instance label pair is misclassified. The smaller the value of Hamming loss, the better the performance. In other words, this metric measures the accuracy at the label level.

To explain the remaining four metrics, we need to introduce the concept about the predicted ranking of labels. In the context of multilabel classification, ranking of labels is calculated based on the probability that label is 1. Most algorithms will assign a score to each label and the ranking is calculated by comparing that score. In ML-CKNN, categorical similarity serves this scoring function.

(2) One-error: this measure evaluates the percentage of labels with the best predicted rank is not correctly classified. The smaller the value, the better the performance.

(3) Coverage: this measure evaluates on average how much value of the predicted rank covers all true positive labels. The smaller the value, the better the performance.

(4) Ranking loss: this measure evaluates the average fraction of label pairs that are incorrectly ordered. The smaller the value, the better the performance.

(5) Average precision: this measure evaluates the average fraction of labels ranked above a particular label y as in Y which actually are in Y. The larger the value, the better the performance. Note that this metric is not reflective of accuracy in the common sense.

5.2. Tuning by Parameters K and T

Recall that K and T are the input parameters of ML-CKNN. In ML-CKNN, the categorical similarity, (x_t,i) in Figure 1 is the scoring function for ranking. It is straightforward to verify that $f(x_t,i)$ depends only on K but not T. As a consequence, K could be deter-

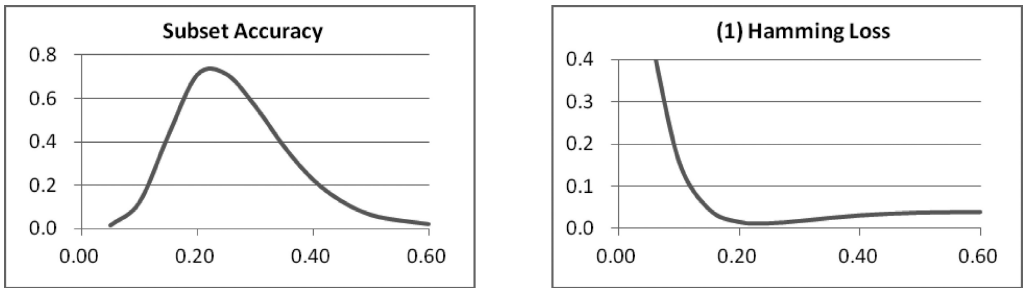

Fig. 3. Performance of ML-CKNN for different values of T.

mined by optimizing metrics (2) to (5). In general, different values of K could be optimal by different metrics. If there is a disagreement, we can adopt a weighted metric for determining the optimal K.

The performance of ML-CKNN is reported in Figure 2 and Table IV in the Appendix. Coincidently, K = 18 optimizes all four metrics. Our results show that as long as K is larger than 10, the performance does not differ much and starts to deteriorate when K is greater than 20. The reason could be that we have only 30 to 40 samples in a few categories. When we set the same value of K across all labels and K is approaching 30, performance is certain to deteriorate when almost all training records in each category are included.

Next, given K = 18, we can further improve the performance by choosing T, the cutoff value for categorical similarity. We evaluate performance by Hamming loss and subset accuracy, both of which are common measures used in the multilabel classification literature. Intuitively, Hamming loss measures the label-level accuracy, whereas subset accuracy measures the sample-level accuracy. Subset accuracy is the strictest measure of prediction success, as it requires the predicted set of class labels to be an exact match of the true set of labels.

Figure 3 and Table V suggest that the optimal T value from minimizing Hamming loss is 0.23 with Hamming loss at 0.0121, which implies that the label-level accuracy is 98.89%. The optimal T value from maximizing subset accuracy is 0.22 with accuracy at 74.94%. We choose T = 0.22 for the following performance comparisons, because its overall performance is slightly better. In this classification exercise, performance by Hamming loss is close to perfect, because one risk factor is typically labeled as 1 in very few of the 25 categories. Most of the labels' true value is 0. Without any intelligent algorithm, setting all labels to zero leads to roughly 1/25 = 4% Hamming loss. As a result, it is easy to achieve very low Hamming loss in this case.

The optimized subset accuracy is around 74.94%, which is much better than the prior probability, which can be conservatively estimated as 1/25 = 4% (i.e., assuming single label with 25 classes). Our analysis about subset accuracy shows that the performance is sensitive to the choice of T. Figure 3 also suggests that using a very low value of T may completely destroy the performance of ML-CKNN. This is intuitive, because ML-CKNN labels a record as 1 when its categorical similarity is greater than T; thus, a low T may lead to an extremely high false positive rate.

5.3. Performance Comparisons

In this section, ML-CKNN is compared with the pioneering ML-KNN algorithm [Zhang and Zhou 2007] and other classic algorithms, including naïve Bayes, decision tree

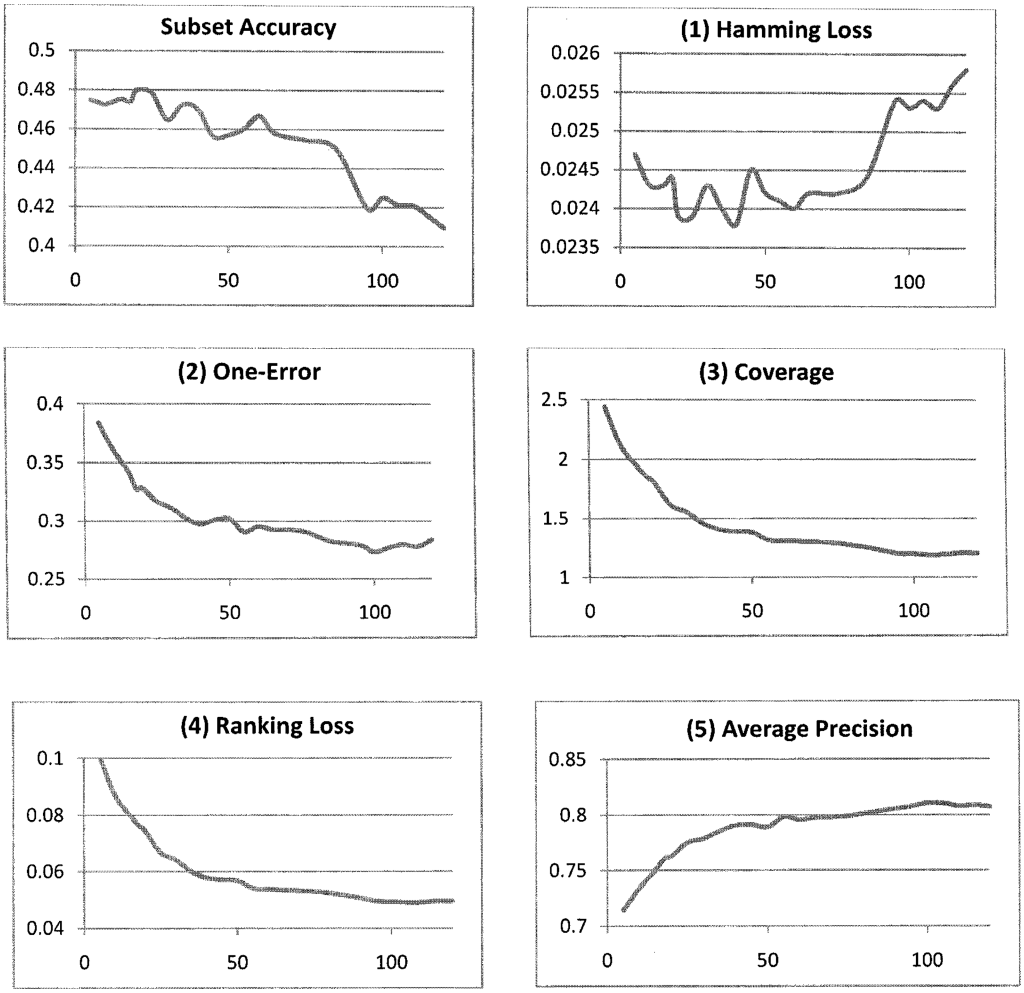

Fig. 4. Performance of ML-KNN for different values of K.

(J48), SVM (from Weka), and RAKEL [Tsoumakas and Vlahavas 2007]. Naïve Bayes, decision tree, and SVM are evaluated as one-versus-all binary classifiers.[4] Ten-fold cross-validation is performed on the dataset in Section 4.

The main benchmarking case is ML-KNN. The experimental results are reported in Figure 4 and Table VI, where the number of nearest neighbors considered by ML-KNN varies from 5 to 120. Consistent with the findings by Zhang and Zhou [2007], Figure 4 and Table VI show that the number of nearest neighbors used by ML-KNN does not significantly affect the performance of the algorithm. Zhang and Zhou [2007] and later studies mainly set $K = 10$ by rule of thumb. In our application, $K = 20$ seems to perform better by most metrics. At $K = 20$, Hamming loss is 0.024, while subset accuracy is 47.99%, both of which are much worse than the proposed ML-CKNN's performance of 0.0125 and 74.94%, respectively.

[4]One-versus-all classifier means we classify each label by a 0-1 binary classifier label-by-label 25 times.

Table II. Experimental Results of Each Multilabel Learning Algorithm (mean ± std)

Evaluation Metric	Algorithm					
	ML-CKNN	ML-KNN	RAKEL	Naïve Bayes	Decision Tree - J48	SVM
Subset Accuracy	0.7494 ± 0.0202	0.4799 ± 0.0284	0.7320 ± 0.0274	0.2941 ± 0.0220	0.7212 ± 0.0284	0.7843 ± 0.0164
Hamming Loss	0.0125 ± 0.0011	0.0239 ± 0.0013	0.0125 ± 0.0015	0.2202 ± 0.0169	0.0129 ± 0.0013	0.0098 ± 0.0008
One-Error	0.1071 ± 0.0120	0.3289 ± 0.0173	0.1794 ± 0.0223	0.5854 ± 0.0217	0.1781 ± 0.0246	0.1976 ± 0.0165
Coverage	0.2369 ± 0.0384	1.8103 ± 0.1307	2.0529 ± 0.2439	3.7825 ± 0.3709	1.5225 ± 0.1471	2.3502 ± 0.1726
Ranking Loss	0.0096 ± 0.0016	0.0744 ± 0.0057	0.0848 ± 0.0100	0.1570 ± 0.0154	0.0628 ± 0.0061	0.0972 ± 0.0073
Average Precision	0.9359 ± 0.0062	0.7626 ± 0.0124	0.8515 ± 0.0177	0.5658 ± 0.0191	0.8640 ± 0.0173	0.8307 ± 0.0126

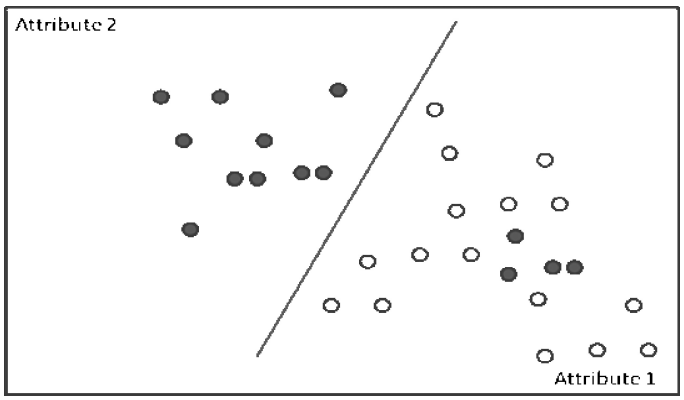

Fig. 5. An example: ML-CKNN vs. SVM.

Comparing ML-KNN to ML-CKNN, the sensitivity of performance in K is a two-edged sword. Since K does not affect the performance of ML-KNN, we can set K at any reasonable value by rule of thumb. However, this property also implies that we cannot improve the performance of ML-KNN by adjusting K. Recall that we can further improve the performance of ML-CKNN by setting different K and T values for different labels. This provides additional support for the superior performance of ML-CKNN in terms of classification accuracy.

Next, we summarize the performance metrics of ML-CKNN (K = 18 and T = 0.22), ML-KNN (K = 20), and four other algorithms in Table II. The value reported in each cell is the mean of metrics from 10 folds where the value following "± " gives the standard deviation. The best result on the mean of each metric is shown in bold.

Table II shows that ML-CKNN performs fairly well in terms of all evaluation criteria. Only SVM outperforms ML-CKNN with respect to Hamming loss and subset accuracy. ML-CKNN performs the best on all other four metrics. Particularly, we can observe that ML-CKNN substantially improves the performance of ML-KNN.

SVM is generally regarded as the best-performing algorithm. Our simple algorithm's main feature is to use few closest samples for classification. Figure 5 illustrates an example in which our algorithm may outperform SVM and most other algorithms.

Interestingly, our results also suggest that ML-CKNN is beneficial for the ranking performance but not in terms of accuracy when compared with the SVM-based algorithm. Investigating the influence on specific performance measures in more detail is an interesting topic for future work.

6. CONCLUSION

This article proposes a multilabel text classification algorithm, ML-CKNN, which is a variant of the state-of-the-art multilabel text classification algorithm ML-KNN. We em-

pirically show that ML-CKNN can outperform ML-KNN and most existing algorithms on classifying risk factors in annual reports. Our application also demonstrates a novel example to apply text classification techniques to produce high-quality information: our labels could provide meanings that are more precise than positive or negative labels. Researchers could exploit similar research opportunities in other structured documents, such as contracts, patents, or other forms for regulatory compliance purposes.

Our classification application may contribute to burgeoning computational studies in accounting and finance along several future research directions. First, risk factors can be used to evaluate the corporate default risks. Potentially, similar risk factors classification algorithms can be incorporated into the existing credit rating information systems used by S&P, Moody's, or Fitch. The second direction is to investigate the relationship of risk factors and the return or volatility of stock prices. Our algorithm could be incorporated into trading analytics systems used by hedge funds. Third, risk factors could be used to predict earnings for stock analysts or mutual fund managers. Compared to abundant marketing data-mining applications, investment analytics is an underexplored gold mine.

The present research has several limitations. First, the definition of risk factor types could be improved, and more risk factors types could be classified. Within each type, there are three to five detailed subcategories that could provide even more information. Performance of classifiers could be further improved by hierarchical classification or by natural language processing techniques. Second, in this article, the distance metric for ML-CKNN is cosine-similarity. It would be interesting to see whether other kinds of distance metrics could further improve the performance of ML-CKNN. At the same time, exploring other definitions of categorical similarity could be another fruitful future work. Lastly, from our experience in conducting empirical performance evaluation, we feel that there is a limit in improving accuracy by classification algorithms. In practice, the best solution that can approach perfect accuracy could be an ensemble approach, that is, using classification algorithms together with classification rules created by domain experts.

APPENDIX

Table III. Risk Factor Categorizations and Definitions

	Name	Definitions and examples
1	Financial condition risks	Factors related to history of loss, resulting in poor financial conditions. e.g., "*We have experienced net losses, and we may not be profitable in the future.*"
2	Restructuring risks	The target company has filed bankruptcy protection, or the company mentioned it is undergoing restructuring. e.g., "*We may need to incur impairment and other restructuring charges, which could materially affect our results of operations and financial conditions.*"
3	Funding risks	Inability to raise capital to expand, for normal operations, or match competition. e.g., "*Banctrust may need to raise capital in the future when capital may not be available on favorable terms or at all.*"
4	Merger & Acquisition risks	Any factor that is related to M&A: e.g., Acquisitions may not meet expectation, or the M&A cost is high. e.g., "*Implementing our acquisition strategy involves risks, and our failure to successfully implement this strategy could have a material adverse effect on our business.*"

(*continued*)

Table III. (*Continued*)

5	Regulation changes	Any risk that is about government regulation changes, including environmental, accounting, or privacy laws. e.g., *"The company is subject to environmental regulations and liabilities that could weaken operating results."*
6	Catastrophes	Natural disasters or terrorists attack. e.g., *"Future terrorist attacks may have a material adverse impact on our business."*
7	Shareholder's interest risks	This includes: (1) The holder's interest is different from the shareholders (2) the shareholder has very strong control power (few large shareholders) (3) no large shareholder. e.g., *"We may encounter conflicts of interest with our controlling stockholder"*
8	Macroeconomic risks	This includes the following risks: economic downturn, financial crisis, high energy price, inflation, or recession. e.g., *"Demand for our products will be affected by general economic conditions."*
9	International risks	This includes factors that are related to global operations, including currency/exchange rate risks. e.g., *"Our international operations are subject to many uncertainties, and a significant reduction in international sales of our products could adversely affect us."*
10	Intellectual property risks	The target company may infringe or be infringed by other company's patents. e.g., "*we may not be successful in adequately protecting our intellectual property."*
11	Potential defects in products	Product liabilities or any risks related to product defects. e.g., *"We may incur substantial costs as a result of warranty and product liability claims which could negatively affect our profitability."*
12	Potential/Ongoing Lawsuits	Current/ongoing significant litigation or lawsuits. e.g., *"We are currently subject to securities class action litigation, the unfavorable outcome of which might have a material adverse effect on our financial condition, results of operations and cash flows."*
13	Infrastructure risks	Risks related to changes, upgrades, maintain the target company's infrastructure, which includes distribution network, IT, or organizational infrastructure. e.g., *"The infrastructure of our transmission and distribution system may not operate as expected, and could require additional unplanned expense which would adversely affect our earnings."*
14	Disruption of operations	Risks about operations may be disrupted due to complex manufacturing process or software systems. e.g., *"Material disruption to our manufacturing plants in Wisconsin could adversely affect our ability to generate revenue."*
15	Human resource risks	Risks about attracting, recruiting, maintaining key personnel or employees, such as CEO, executives, R&D staff, or sales people. e.g., *"We depend upon our key personnel and they would be difficult to replace."*
16	Licensing related risks	Dependent on other company's technology licensing or government license to operate business. e.g., *"If we are unable to renew our licenses or otherwise lose our licensed rights, we may have to stop selling products or we may lose competitive advantage."*
17	Suppliers risks	Any risks related to upstream suppliers, including OEM manufacturers. e.g., *"A change in sales strategy by the company's suppliers could adversely affect the company's sales or earnings."*

(*continued*)

Table III. (*Continued*)

18	Input prices risks	Any risks about the input prices (raw material prices) may go up. e.g., *"Our inability to pass through increases in costs and expenses for raw materials and energy, on a timely basis or at all, could have a material adverse effect on the margins of our products."*
19	Rely on few large customers	High concentration on few large customers. e.g., *"Our sales could be negatively impacted if one or more of our key customers substantially reduce orders for our products"*
20	Competition risks	Industry is competitive, faces strong or increasing competition. e.g., *"We compete in distribution industries that are highly competitive and we may not be able to compete successfully."*
21	Industry is cyclical	Industry is cyclical. e.g., *"We operate in an industry that is cyclical and that has periodically experienced significant year-to-year fluctuations in demand for vehicles*
22	Volatile demand and results	Demand and/or Financial results are volatile and unpredictable. e.g., *"Our future revenue, gross margins, operating results and net income are difficult to predict and may materially fluctuate."*
23	Volatile stock price risks	The target company's stock price is volatile. e.g., *"The price of our common stock has fluctuated widely in the past and may fluctuate widely in the future."*
24	New product introduction risks	Potential delays or fails in new product introduction, or the new production introduction is critical to the target company's success. e.g., *"Our success depends on our ability to successfully develop and commercialize additional pharmaceutical products"*
25	Downstream risks	Risks associated with distributors or retailers. e.g., "*We face a number of risks related to our product sales through intermediaries.*"

Table IV. Performance of ML-CKNN with Different Values of K

N	One-Error	Coverage	Ranking Loss	Average Precision
1	0.183	0.513	0.0211	0.884
2	0.145	0.381	0.0156	0.909
3	0.130	0.327	0.0134	0.919
4	0.121	0.291	0.0119	0.925
5	0.119	0.274	0.0112	0.928
6	0.119	0.264	0.0108	0.928
7	0.120	0.256	0.0104	0.929
8	0.118	0.254	0.0104	0.930
9	0.114	0.249	0.0101	0.932
10	0.111	0.244	0.0099	0.934
11	0.112	0.241	0.0098	0.934
12	0.111	0.241	0.0098	0.934
13	0.110	0.239	0.0097	0.935
14	0.108	0.236	0.0096	0.935
15	0.110	0.237	0.0097	0.935
16	0.108	0.237	0.0097	0.936
17	0.108	0.240	0.0097	0.935
18	**0.107**	**0.237**	**0.0096**	**0.936**
19	0.108	0.239	0.0097	0.936

(*continued*)

Table IV. (*Continued*)

20	0.108	0.239	0.0097	0.935
21	0.109	0.240	0.0098	0.935
22	0.110	0.244	0.0099	0.934
23	0.111	0.247	0.0101	0.934
24	0.112	0.251	0.0102	0.933
25	0.112	0.254	0.0103	0.933
26	0.112	0.257	0.0105	0.933
27	0.113	0.261	0.0106	0.932
28	0.112	0.261	0.0106	0.933
29	0.111	0.262	0.0107	0.933

Table V. Performance of ML-CKNN with Different Values of T

T	Hamming Loss	Subset Accuracy		T	Hamming Loss	Subset Accuracy
0.05	0.4815	0.0128		0.20	0.0159	0.7112
0.10	0.1612	0.1198		0.21	0.0138	0.7360
0.15	0.0465	0.4300		0.22	0.0125	**0.7494**
0.20	0.0159	0.7112		**0.23**	**0.0121**	0.7468
0.25	0.0129	0.7146		0.24	0.0122	0.7341
0.30	0.0182	0.5667		0.25	0.0129	0.7146
0.35	0.0254	0.3787		0.26	0.0135	0.6955
0.40	0.0314	0.2250		0.27	0.0144	0.6685
0.45	0.0353	0.1240		0.28	0.0156	0.6391
0.50	0.0378	0.0615		0.29	0.0168	0.6053
0.55	0.0388	0.0357		0.30	0.0182	0.5667
0.60	0.0394	0.0191				

Table VI. Performance of ML-KNN with Different Values of K

K	Hamming Loss	Subset Accuracy	One-Error	Average Precision	Coverage	Ranking Loss
5	0.0247	0.4745	0.3846	0.7141	2.4425	0.101
10	**0.0243**	**0.4726**	**0.3614**	**0.734**	**2.1127**	**0.0873**
15	0.0243	0.4752	0.3438	0.7502	1.9365	0.08
18	0.0244	0.4736	0.3279	0.7611	1.8521	0.0763
20	**0.0239**	**0.4799**	**0.3289**	**0.7626**	**1.8103**	**0.0744**
25	0.0239	0.4784	0.3174	0.7748	1.6215	0.0666
30	0.0243	0.4646	0.3117	0.7786	1.5601	0.0641
35	0.024	0.4723	0.303	0.7852	1.4584	0.06
40	0.0238	0.4701	0.297	0.7906	1.4055	0.0578
45	0.0245	0.4564	0.3011	0.7911	1.3873	0.0571
50	0.0242	0.457	0.3024	0.789	1.3822	0.0568
55	0.0241	0.4602	0.2906	0.7982	1.3169	0.0541
60	0.024	0.4666	0.2951	0.7954	1.3089	0.0538
75	0.0242	0.4545	0.2912	0.799	1.2927	0.053
100	0.0253	0.4251	0.2731	0.8106	1.2003	0.0493
120	0.0258	0.4098	0.2833	0.8073	1.2035	0.0495

REFERENCES

Abdou, K. and Dicle, M .F. 2007. Do risk factors matter in the Ipo valuation? *J. Finan. Regul. Compl. 15*, 1, 63–89.

Antweiler, W. and Frank, M. Z. 2004. Is all that talk just noise? The information content of internet stock message boards. *J. Finance, 59*, 3, 1259–1294.

Bai, X., Nunez, M., and Kalagnanam, J. 2010. Managing data quality risk in accounting information systems. *Inf. Sys. Res.*, To appear.

Balakrishnan, R., Qiu, X. Y., and Srinivasan, P. 2010. On the predictive ability of narrative disclosures in annual reports. *Euro. J. Oper. Res., 202*, 3, 789–801.

Brucker, F., Benites, F., and Sapozhnikova, E. 2010. Multi-Label classification and extracting predicted class hierarchies. *Patt. Recogn. 44*, 724–738.

Campbell, J. L., Chen, H., Dhaliwal, D.S., Lu, H., and Steele, L. 2011. The information content of mandatory risk factor disclosures in corporate filings, .Tech. repo. ,University of Arizona.

CHENG, W. AND HÜLLERMEIER, E. 2009. Combining instance-based learning and logistic regression for multilabel classification. *Mach. Learn. 76*, 2, 211–225.

COVER, T. AND HART, P. 1967. Nearest neighbor pattern classification. *Inf. Theory 13*, 1, 21–27.

DAS, S. AND CHEN, M. 2007. Yahoo! For amazon: sentiment extraction from small talk on the web. *Manag. Sci., 53*, 9, 1375–1388.

DE COMITÉ, F., GILLERON, R., AND TOMMASI, M. 2003. Learning multi-label alternating decision trees from texts and data. In *Proceedings of the International Conference on Machine Learning and Data Mining in Pattern Recognetion.* 251–274.

ELISSEEFF, A. AND WESTON, J. 2002. Kernel methods for multi-labelled classification and categorical regression problems. *Advan. Neural Inf. Process. Syst. 14*, 681–687.

FELDMAN, R., GOVINDARAJ, S., LIVNAT, J., AND SEGAL, B. 2009. Management's tone change, post earnings announcement drift and accruals. *Rev. Accoun. Studies*, to appear.

GU, B., KONANA, P., RAJAGOPALAN, B., AND CHEN, H. 2007. Competition among virtual communities and user valuation: The case of investing-related communities. *Inf. Syst. Res. 18*, 1, 68–85.

HAN, J. AND KAMBER, M. 2006. *Data Mining: Concepts and Techniques*. Morgan Kaufmann.

HANLEY, K. AND HOBERG, G. 2010. The information content of Ipo prospectuses. *Rev. Finan. Studies, 23*, 7, 2821–2864.

KOTHARI, S. P., LI, X., AND SHORT, J. E. 2009. The effect of disclosures by management, analysts, and business press on cost of capital, return volatility, and analyst forecasts: A study using content analysis. *Account. Rev. 84*, 5, 1639–1670.

LI, F. 2008. Annual report readability, current earnings, and earnings persistence. *J. Account. Econo. 45*, 2-3, 221–247.

LI, F. 2010. The information content of forward-looking statements in corporate filings—A naive bayesian machine learning approach. *J. Account. Res.* To appear.

LI, F. 2011. Textual analysis of corporate disclosures: A survey of the literature. *J. Account. Liter.* To appear.

LI, H., GUO, Y., WU, M., LI, P., AND XIANG, Y. 2010. Combine multi-valued attribute decomposition with multi-label learning. *Expert Syst. Appl. 37*, 12.

LOUGHRAN, T. AND MCDONALD, B. 2009. When is a liability not a liability. *J. Finan.* To appear.

LOUGHRAN, T. AND MCDONALD, B. 2010. Measuring readability in financial text. *SSRN eLibrary*.

MANGEN, C. AND DURNEV, A. 2010. The real effects of disclosure tone: Evidence from restatements. *SSRN eLibrary*.

NELSON, K. K. AND PRITCHARD, A. C. 2007. Litigation risk and voluntary disclosure: The use of meaningful cautionary language. *SSRN eLibrary*.

SCHAPIRE, R. E. AND SINGER, Y. 2000. Boostexter: A boosting-based system for text categorization. *Mach. Learn. 39*, 2, 135–168.

SCHUMAKER, R. P. AND CHEN, H. C. 2009. A quantitative stock prediction system based on financial news. *Inf. Process. Manag. 45*, 5, 571–583.

TETLOCK, P., SAAR-TSECHANSKY, M., AND MACSKASSY, S. 2008. More than words: quantifying language to measure firms' fundamentals. *J. Finance, 63*, 3, 1437–1467.

TETLOCK, P. C. 2007. Giving content to investor sentiment: The role of media in the stock market. *J. Finance,62*, 3, 1139–1168.

TSOUMAKAS, G., KATAKIS, I., AND VLAHAVAS, I. 2010. Mining multi-label data. In *Data Mining and Knowledge Discovery Handbook*, 667–685.

TSOUMAKAS, G. AND VLAHAVAS, I. 2007. Random k-labelsets: An ensemble method for multilabel classification. In *ProceedingS of the Conference on Machine Learning: ECML.* 406–417.

ZHANG, M. L. AND ZHOU, Z. H. 2006. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Trans. Knowl. Data Engin.* 1338–1351.

ZHANG, M. L. AND ZHOU, Z. H. 2007. Ml-Knn: A lazy learning approach to multi-label learning. *Patt. Recogn., 40*, 7, 2038–2048.

Received May 2011; revised August 2011; accepted August 2011