Decision Support Systems 50 (2010) 164–175

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

Making words work: Using financial text as a predictor of financial events

Mark Cecchini [a,*], Haldun Aytug [b,1], Gary J. Koehler [b,2], Praveen Pathak [b,3]

[a] School of Accounting, Darla Moore School of Business, University of South Carolina, United States
[b] Department of Information Systems and Operations Management, Warrington College of Business, University of Florida, United States

ARTICLE INFO

Article history:
Received 3 March 2009
Received in revised form 21 April 2010
Accepted 27 July 2010
Available online 6 August 2010

Keywords:
Automatic text analysis
Financial event prediction
Management fraud
Bankruptcy
SVM
WordNet

ABSTRACT

We develop a methodology for automatically analyzing text to aid in discriminating firms that encounter catastrophic financial events. The dictionaries we create from Management Discussion and Analysis Sections (MD&A) of 10-Ks discriminate fraudulent from non-fraudulent firms 75% of the time and bankrupt from nonbankrupt firms 80% of the time. Our results compare favorably with quantitative prediction methods. We further test for complementarities by merging quantitative data with text data. We achieve our best prediction results for both bankruptcy (83.87%) and fraud (81.97%) with the combined data, showing that that the text of the MD&A complements the quantitative financial information.

1. Introduction

There are many classification problems that can be addressed using quantitative financial variables. Accounting and financial research is heavily laden with archival studies which utilize publicly available financial data and large datasets in order to find salient relationships between variables and an event, such as bankruptcy [1,14], fraud [5,13] or an important financial benchmark, such as cost of capital [4]. The overwhelming majority of variables in archival studies are quantitative in nature. Recently, researchers have begun to look at text in financial statements to help better understand the dynamics in this previously untapped information source. Evidence that this area is gathering interest is seen in the work by Davis et al. [11], Li [19,22], Tetlock [30], Tetlock et al. [31], Antweiler and Frank [3], Das and Chen [10], and Aasheim and Koehler [1] which utilize text information from annual reports, the financial press, and internet message boards as indicators of financial phenomena. These papers rely on methodologies developed in the Information Retrieval (IR) literature.

Business-related text analysis research is not just relegated to financial analyses. Holton [15] looks at the previously untapped resource of e-mail correspondences as a proxy for determining disgruntled employees. This is important as disgruntled employees raise the risk of some types of fraud. Qu et al. [27] content analyze consumer comments from online rating systems to help determine what is important to the customer. Li and Wu [21] query online forums to detect hotspots (discussions of high interest and high volatility). Their research method employs sentiment analysis in order to detect the overall tone of the forum. They use support vector machines (as we do) to aid in the text mining effort of grouping the forums into clusters. Their method supports environmental scanning, like Wei and Lee [33]. Wei and Lee [33] develop a tool to detect events from new stories. They use a supervised learning method to extract information from known events to help classify future (unknown events). In our research, we attempt to detect financial events based on financial reports using a dictionary of tokens that discriminate between known event and nonevent firms. Gerdes Jr. [13] develops a tool to comb through EDGAR and pull out statements from SEC filings. Gerdes Jr.'s tool is meant to provide foundational support for researchers who want to exploit SEC filings and is effective when the researcher knows *a priori* the keywords she is trying to extract. Here, we do not assume *a priori* knowledge of the words or phrases important to financial events detection. We use computational linguistics tools to make our dictionary creation automatic.

Our research endeavors to create a new methodology that can be used by accounting and finance researchers who want to define a dictionary of key terms from financial text that are associated with an event or a state of nature. These dictionaries can be content analyzed by researchers for potentially valuable insights. A researcher may hypothesize that the number of times a keyword appears in a document is positively or negatively related to an outcome (such as Feng Li's [19] hypothesis about the word "risk" and future financial distress). Our methodology would help to automate the task of

* Corresponding author. Tel.: +1 803 777 6643.
E-mail addresses: cecchini@moore.sc.edu (M. Cecchini), haldun.aytug@cbal.ufl.edu (H. Aytug), koehler@ufl.edu (G.J. Koehler), Praveen.pathak@cba.ufl.edu (P. Pathak).
[1] Tel.: +1 352 392 2468.
[2] Tel.: +1 352 846 2090.
[3] Tel.: +1 352 392 9599.

0167-9236/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.dss.2010.07.012

finding a dictionary of many key words and rank these words in order of discriminatory power. The dictionaries can also be quantified into token counts which can be used in analyses for the benefit of augmenting prior research models which utilized only quantitative data. We test our methodology on two important financial events, fraud and bankruptcy. We create dictionaries of keywords that can help us predict fraud and bankruptcy, based on text from the Management Discussion and Analysis Section (MD&A) of the 10-K reports. This method could be used by regulators and government agencies to help determine the firms to investigate for fraud. It could also be used by investors to help determine the firms which are at higher risk for bankruptcy.

Prior literature utilizing text analysis (manual or automatic), for example [3,20,31], starts with a hypothesis about the structure of the text. We start with a corpus (a collection of documents) from firms exhibiting an event of interest (like fraud) and a second corpus from firms not exhibiting the event and use an automated procedure to content analyze the documents to produce a dictionary of keywords that discriminate two sets of documents. We also utilize the word counts of the keywords in order to create a function to discriminate the two classes. This function is tested on a holdout set of documents. The accuracy of the function as a discriminator helps to determine the value of the dictionary.

Our overarching contribution is the development of a methodology tuned specifically to financial events prediction. The result is several interrelated contributions. At the outset, we develop a text methodology and test its efficacy on two different financial events. Our tool is a blunt instrument developed using computational linguistics theories. There is no human intervention involved. Thus we show conservatively that there is textual information in the annual reports that can be exploited for the purposes of financial events detection. As a benchmark, we compare the results of our text method to traditional prediction methods (using quantitative financial variables). Our results show that textual information is competitive with quantitative information. We also show that text and quantitative information, when combined together, give the best results, showing that complementarities exist between text and quantitative data. We extend technical research in computational linguistics via the dictionary creation method. This method extends the Vector Space Model (VSM) to include WordNet and tunes the weights to discriminate between documents from two different domains (event and nonevent).

The rest of the paper is organized as follows. In Section 2 we give background literature and the technical roots of the project. In Section 3 we develop our methodology. In Section 4 we find support for the methodology and in Section 5 we conclude the paper and discuss future research avenues.

2. Background literature

2.1. Text analysis in financial literature

Text analysis in financial literature is a recent phenomenon as vast amounts of electronic financial text have become readily available with the rise of the internet and SEC's requirement for electronic filings. Antweiler and Frank [3] and Das and Chen [11] research the effect of stock message boards on the stock market. Antweiler and Frank [3] use an off-the-shelf naïve Bayes method called the Rainbow package [23]. They endeavor to determine a measure of bullishness in the message boards for a stock. Using this bullishness measure they attempt to find out if the information on message boards is just noise. They reject this notion and find that the message boards carry information. For example, they find that disagreement in the message boards is correlated with heavier trading volume. Das and Chen [11] extend this study by using several statistical methods (including the Rainbow package) to determine bullishness. The result from each method is aggregated via a voting scheme. The predictive ability of these messages is weak for individual stocks but strong in the aggregate. These studies show that although noisy, even stock message boards carry valuable information. Ma et al. [24] look at inter-company citations in news stories and, using social network analysis, construct company revenue relations. Their work is especially useful when financial data is hard to get.

Li [20] explains that longer annual reports with more complex language are less readable to the ordinary investor. He hypothesizes that readability may be positively correlated with future earnings persistence and current earnings. His study uses the FOG index from computational linguistics to give a complexity score to each annual report. The findings support his hypothesis. Li's study shows that, at a coarse level, the text of annual reports do contain value relevant information (the FOG index gives an aggregate view of the financial text in an annual report). A more fine-grained tool, such as the one we propose, would designate key words and phrases within the annual report (or other text) that are significantly correlated with an event or happening.

There are a number of recent studies that hypothesize that the tone of a body of text affects an outcome. Davis et al. [12] study the tone of earnings press releases and find that an optimistic tone is associated with a higher future return on assets. The results also show that unexpected optimistic or pessimistic language is picked up by market participants, affecting share prices. This study utilizes an off-the-shelf software package called DICTION 5.0 to analyze the earnings press releases. DICTION is otherwise used to analyze political discourse by comparing the counts of optimism-increasing and optimism-decreasing words (based on linguistic theory).

Tetlock et al. [33] use a simplified VSM (described below in detail) in conjunction with a set of positive and negative words from the General Inquirer's (GI) Harvard IV-4 psychosocial dictionary to see if negative words in the popular press (specifically the Wall Street Journal and the Dow Jones News Service) affect share price. An interesting finding of the paper was that the qualitative information was complementary to the quantitative information. In other words, the text did not just indicate what was happening quantitatively for a firm, but was additive in value. This finding gives weight to the fact that there is value in the text of the popular press. Tetlock [30] uses the same methodology to test the interaction between the popular press (using the Wall Street Journal's "Abreast of the Market") and broad-based market participation. The study finds that a high amount of pessimism leads to lower market prices and unusually high or low values of pessimism lead to higher trading volume. The author creates a trading strategy based on the negative word counts that leads to positive abnormal returns.

The methodology used in Tetlock et al. [31], Tetlock [30] and Davis et al. [12] achieve interesting results using a small portion of the text. A methodology which could capture sentiment and other salient words or phrases by combing the entire text of information-rich content has the potential to open up new insights for this research community. This paper develops and tests such a methodology.

Li [19] hypothesized that risk and uncertainty for a firm is positively correlated with the number of times the words "risk" and "uncertainty" are found in a company's MD&As portion of an annual report. If the counts of the words "risk" and "uncertainty" increase year-over-year, the firm is likely to have more negative earnings the next year. This simple formula was used to create a trading rule which led to positive abnormal returns. The power of these keywords underscores the potential information content of the text in annual reports. A methodology that can fully utilize text data to develop an ontology[4] of key concepts and phrases might lead to a greater understanding of the linguistic character of financial text. This is the

[4] An ontology is a set of concepts based on a particular area of interest.

primary goal of this research. For the rest of this section we give some background literature related to the methodology we create in Section 3.

2.2. Vector space model

A primary goal of IR research is to relate relevant documents to user requests (for example, a search engine attempts to find the documents that best match a user-defined search). Using IR methods, one seeks to separate relevant textual documents from non-relevant ones. A powerful method in IR research is VSM [8,18,27]. After preprocessing,[5] the document is transformed into a vector of key word counts. Each index of the vector represents the count of a particular word in the document. The document vector is normalized (usually by the total number of words in the document). The resulting document vector is a complete transformation from qualitative text into quantitative representations of the documents.

Following Jurafsky [16], each document in the vector space model is represented by a vector of keywords as follows:

$$d_j = \left(w_{1,j}, w_{2,j}, \ldots w_{n,j}\right)'$$

where n is the number of keywords and w_{ij} is the weight for keyword i in document j. This characterization allows us to view the whole document collection as a matrix of weights and is called the term-by-document matrix. The columns of this matrix are the documents, d_j, and the terms are the rows. The cosine between two document vectors j and k provides a similarity measure.

When the vectors representing two documents are identical, they will have a similarity measure of 1; when there are no words in common between the two documents the vectors are orthogonal, and will have a similarity measure of 0. Finding weights that most accurately depict the importance of the keywords in the collection is very important to document classification. Word counts alone cannot always effectively discriminate between documents in a collection. Sparck Jones [29] made a seminal breakthrough on this problem using the tf–idf function for weights. The function stands for Term Frequency, Inverse Document Frequency. The basic tf–idf function is as follows:

$$w_{ij} = tf_{ij} * \log\frac{N}{n}.$$

tf_{ij} is the frequency of term t_i in document d_j, N is the number of documents in the collection and n is the number of documents where the term t_i occurs at least once. The logic is as follows: for tf_{ij} or term frequency, a word that occurs more often in a document is more likely to be important to the classification of that document. A measure of inverse document frequency, idf, is defined by $idf_{ij} = \log\frac{N}{n}$. A word that occurs in all documents is not helpful in the classification of the document (hence the inverse) and therefore gets a 0 value. A word that appears in only one document is likely to be helpful in classifying that document and gets a value of 1.

Many researchers have attempted to improve upon the basic vector space model. Improvements take the form of making the document vector more accurately depict the document itself. Part-of-speech tagging is one such improvement. A part-of-speech tagger [6] reads a document in natural language and tags every word with a part-of-speech identifier, such as noun, verb, adjective and adverb. The tags are created using sentence structure.

Another improvement is word sense disambiguation (WSD). WSD is an attempt to understand the actual meaning of a word, in the context of a sentence. Often words that are spelled identically have several meanings. In the basic vector space model, the document vector would take all instances of the word "crane" and add them up. What if one sentence read, "The crane is part of the animal kingdom" and another sentence read, "The crane was the only thing that could move the 2 ton truck to safety"? A word sense disambiguated vector would have two versions of the word crane if both showed up in the text. This avoids some confusion that might arise were we comparing the similarity between two documents, one which was about the bird, and the other about the piece of equipment. A tool that has been widely used to aid WSD is WordNet. In the next subsection we explain WordNet and its value for WSD.

2.3. WordNet

"WordNet is an online lexical reference system with a design that is inspired by current psycholinguistic theories of human lexical memory. English nouns, verbs, adjectives and adverbs are organized into synonym sets, each representing one underlying lexical concept [26]". A lexical reference system is one which allows a user to type in a word and get in return that word's relationships with other words. During this research time frame, WordNet included 117,798 nouns, 11,529 verbs, 21,479 adjectives and 4481 adverbs [26]. Hand-crafted by linguists, the basic relation in WordNet is called synonymy. Sets of synonyms (called synsets or concepts) form its basic building blocks. For example, the word "history" is in the same block as the words past, past times, yesteryear, and yore. WordNet defines a synset as "a set of words that are interchangeable in some context without changing the truth value of the preposition in which they are embedded [26]".

There are two main types of relations in WordNet, lexical relations and semantic relations. Lexical relations are between words and semantic relations are between concepts. Nouns are organized into a separate lexical hierarchy as are verbs, adjectives and adverbs. Nouns are separated from other parts of speech because their relationships are considered different than the relationships between verbs and adjectives. A relationship between concepts can be hierarchical, as is the case of hyponyms and hypernyms. A hyponym/hypernym is an "is a" relation on nouns. WordNet uses a rooted tree structure. Starting at the root node, the concept is very general (as in the case of "entity" or "psychological feature"). As one goes up the tree, one encounters more fine-grained concepts. For example, a robin is a subordinate of the noun bird and bird is a superordinate of robin. The subordinates are called hyponyms (is a kind of bird) and the superordinates are called hypernyms (robin is a kind of bird). Modifiers, which are adverbs and adjectives are connected similarly as are verbs. Synonymy and hyponymy are two of many relations in WordNet. See Sarkar [28] for a list of other WordNet relations with examples.

Traditional vector space model retrieval techniques focus on the number of times a word stem appears in a document without considering the context of the word. Consider the following two sentences, "What are you eating?" and "What's eating you?" The words "what," "are" and "you" would most likely be stop words. So what's left in both sentences is the word "eating". Therefore, the two sentences above would have identical meaning in the vector space model even though they are completely different. Having the correct WordNet concept labels for "eating" in both sentences would mean that each one would get its own spot in a vector space model vector since they have different meanings. Further, using concepts and contexts it is possible to create a lexical reference system that interprets data specific to a particular area of interest, such as financial events prediction.

[5] Preprocessing consists of stop-word filtering and truncating all words in the document into word stems (called stemming or lemmatizing). Stop words are commonly occurring words such as 'the', 'and' etc. Word stems are the base form of words, without suffixes. Stemming is important because the meaning of words like "stymies" and "stymied" are basically the same. If we stem the two words, they both become "stymie."

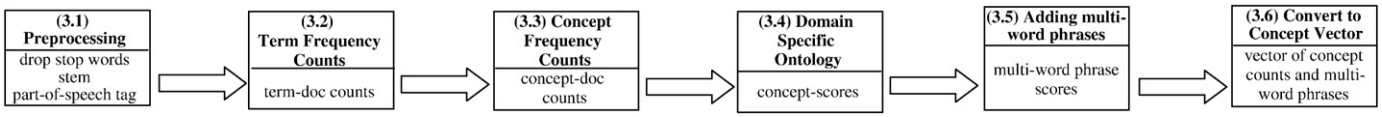

Fig. 1. Steps to determining discriminative text.

2.4. Ontology creation

An ontology is a set of concepts based on a particular area of interest. For example, a glossary at the back of a textbook is an ontology for the book's subject matter. Navigli and Velardi [25] explain that there are top ontologies, upper domain ontologies, and specific domain ontologies. Top ontologies are populated with general, abstract concepts (for example, a dictionary of the English language). WordNet is a top ontology. Upper domain ontologies are more specialized, but still very general. Specific domain ontologies are populated with concepts that focus on a narrow subject. Navigli and Velardi [25] explain "A domain ontology seeks to reduce or eliminate conceptual and terminological confusion among the members of a user community who need to share various kinds of electronic documents and information." In this paper we create ontologies specific to the accounting domain with the purpose of finding words that discriminate between two classes (the financial event class and the nonfinancial event class).

Khan and Luo [17] construct ontologies using domain corpora and clustering algorithms. Navigli and Velardi [25] extract candidate terminology from a domain corpus and filter it against contrastive corpora (a set of documents of general subject matter, such as the contents of a newspaper) in order to develop their ontology. Contrastive corpora are used to help tease out the domain specific concepts. This is done by contrasting the general words/phrases one would find in a non-specialized publication (such as a newspaper) against the words/phrases found in literature specific to the domain.

Buitelaar and Sacaleanu [7] create a method of ranking synsets by domain relevance. The relevance of a synset is determined by its importance to the domain corpus. The importance is determined by the amount of times the concept appears in the corpus. The result is an ordered list of domain terms. Our methodology automatically develops an ordered list of discriminatory synsets using WordNet. In the next section we explain this methodology in detail.

3. Methodology

We develop a methodology for automating ontology creation. It includes preprocessing, obtaining term frequency counts, using WordNet to collapse term frequency counts into concept frequency counts, development of the domain specific ontology based on the concepts with the highest discriminating power, scoring multi-word phrases, and finally, converting the top scoring concepts/multi-word phrases to a vector of values for use in any number of analyses, including prediction and regression. Fig. 1 above shows the steps to be completed. The steps are explained in detail below.

3.1. Preprocessing

Each document in both corpora[6] goes through a preprocessing routine which includes: 1. removing stop words, 2. stemming, and 3. part-of-speech tagging.

[6] For this paper, a corpus is a set of documents of a single class. In the case of bankruptcy, the bankrupt set of documents are one corpus and the nonbankrupt set of documents are another corpus. A domain in our work would include all documents within the same test set (bankrupt and fraud). For example, the fraud domain would include all fraud and nonfraud cases.

3.2. Term frequency counts

Term frequency counts (tf_{ij}) are computed for each word i in document j. Fig. 2 below shows a stylized illustration using two bankrupt and two nonbankrupt firm documents. Fig. 2 is referred to in this, the *Concepts*, and the *Domain Specific Ontology* subsections to illustrate the method via example. Going from left to right, the steps to dictionary creation are applied to the original term frequency counts. Column I shows the term frequency counts. Column II shows the term frequency counts after removing Stop Words and Stemming. Column III shows the Part-of-Speech tagged words.

3.3. Concepts

In this step, all terms become WordNet Concepts. This process creates two further levels of word sense disambiguation. In the first, the terms are consolidated into WordNet Concepts. As explained above, a WordNet Concept may encompass many words that mean basically the same thing (via synsets). Tagging each word with a WordNet concept will inevitably lead to consolidation. This is because there will be situations where two words will be part of the same synset (such as debt, liability and financial obligation). The second level of word sense disambiguation uses a heuristic to help choose the correct sense of the word for the domain. Each word and part-of-speech (for example debt/noun) may have more than one meaning. In order to determine the most appropriate meaning we add the word counts for all words in each synset for that particular word. The synset with the highest score for the whole domain is the one we deem to be the concept most appropriate for the domain. Below is an illustrative example showing two senses of the word liability:

Sense 1: indebtedness, liability, financial obligation
Sense 2: liability, susceptibility, susceptibleness

Going through all of the documents in both corpora, we add the word counts for indebtedness, liability, financial obligation, susceptibility and susceptibleness. Then we add the total word counts for each sense (i.e., we add counts for indebtedness, liability, and financial obligation to arrive at total word count for sense 1 and add counts for liability, susceptibility, and susceptibleness to arrive at total word count for sense 2). The sense with the highest number becomes the domain sense. Notice that liability will add to the scores of both Sense 1 and Sense 2. The idea is that we want to find the concept that is most appropriate for the domain. One would suspect that looking at MD&As we would find Sense 1 to be the most appropriate. The two levels of WSD from this example are:

1. The terms indebtedness, liability and financial obligation will now be consolidated into a single concept. This has the advantage of reducing the number of attributes which usually helps control overfitting and reduces computational overhead.
2. The heuristic determines the synset that is most frequently discovered in the domain and makes that the domain synset so that any time the individual words in Sense 1 appear, they will be

A - Bankrupt Firms

I	II	III	IV	V
tf	Stop List/Stemmed	Part-of_Speech Tagged	WordNet Concept Shrunk	Concept document score (cds)
Bankrupt Firm 1				
automobile 10	automobile 10	automobile - Noun 10	automobile - Noun 17	automobile - Noun 17 * log (3/2) = 2.99
car 7	car 7	car – Noun 7		
risk 5	risk 5	risk – Noun 5	risk – Noun 5	risk – Noun 5 * log (3/1) = 2.39
debt 7	debt 7	debt – Noun 7	debt – Noun 7	debt – Noun 7 * log (3/1) = 3.34
net loss 3	net loss 3	net loss – Noun 3	net loss – Noun 3	net loss – Noun 3 * log (3/1) = 1.43
a 8				
Bankrupt Firm 2				
oil 8	oil 8	oil – Noun 8	oil – Noun 12	oil – Noun 12 * log (3/1) =5.73
petroleum 4	petroleum 4	petroleum – Noun 4		
loans 4	loan 4	loan – Noun 4	loan – Noun 4	Loan – Noun 4 * log (3/1) = 1.91
net losses 5	net loss 5	net loss – Noun 5	net loss – Noun 5	net loss – Noun 5 * log (3/1) = 2.39
And 5				

B - NonBankrupt Firms

I	II	III	IV	V
tf	Stop List/Stemmed	Part-of_Speech Tagged	WordNet Concept Shrunk	Concept document score (cds)
NonBankrupt Firm 1				
automobiles 9	automobile 9	automobile – Noun 9	automobile – Noun 9	automobile – Noun 9 * log (3/2) = 1.584821
net income 6	net income 6	net income – Noun 6	net income – Noun 11	net income – Noun 11 * log (3/1) = 5.248334
earnings 5	earning 5	earnings – Noun 5		
acquisition 4	acquisition 4	acquisition - Noun 4	acquisition - Noun 7	acquisition - Noun 7 * log (3/1) = 3.339849
purchase 3	purchase 3	purchase – Noun 3		
the 4				
NonBankrupt Firm 2				
Manufacturing 5	Manufacturing 5	Manufacturing – Noun 5	Manufacturing – Noun 5	Manufacturing – Noun 5 * log (3/1) = 2.385606
Growth 8	Growth 8	Growth - Noun 8	Growth - Noun 8	Growth - Noun 8 * log (3/1) = 3.81697
Net Income 6	Net Income 6	Net Income – Noun 6	Net Income – Noun 9	Net Income – Noun 9 * log (3/1) = 1,584821
Earnings 3	Earning 3	Earning – Noun 3		
But 6				

Fig. 2. Illustrative example of Methodology.

immediately tagged as Sense 1 of liability. The result of this step is a set with WordNet concept counts cf_{ij}, where

$$\sum_j tf_{ij} = \sum_i \sum_j cf_{ij} \quad (1)$$

Fig. 2, column IV shows the concept counts (note the reduced dimensions).

3.4. Domain specific ontology

Below we develop two discriminatory functions, one gives a concept document score and the other a concept score. A high concept document score indicates that a concept has discriminatory power between a particular document in a corpus and all documents of another corpus (for example, a bankrupt firm MD&A against all nonbankrupt firm

MD&As). The concept score captures discriminatory concepts which are common among the documents within a corpus (for example, common concepts among bankrupt firms). Here we explain our method of creating a concept score for each concept based on its ability to help discriminate between the two corpora. The process is a two step process. In step 1, a concept document score (*cds*) is created using a contrastive corpus. In step two a concept score (*cs*), is created.

Step 1. In order to create the *cds* we extend the *tf–idf* function. We change *tf* to *cf* as we are looking at concepts, not terms. We also change the inverse document frequency portion of the function. This change is made to find the concepts that best separate an individual document in one corpus from all documents in the other corpus. The normal *tf–idf* function looks for keywords that discriminate between any number of documents without the additional information of discriminating between a class of documents. Our function is shown below:

$$cds_{ijk} = cf_{ijk} * \log\left(\frac{N_l + 1}{n_l + 1}\right), k, l \in \{1,2\}, k \neq l \quad (2)$$

where i is the concept index, j is the document index and k,l are the corpus indices. For example, cds_{ij1} is the concept document score of document i of concept j in corpus 1 with respect to corpus 2 (since we consider only two corpora). We add a "+1" to the *idf* portion of the function to indicate the document from corpus k (so the total documents under consideration are the entire corpus l plus the individual document from k). A high *cds* score indicates that concept i has discriminatory power between document j in corpus k from all documents in corpus l. *cds* are normalized by dividing cf_{ijk} by the total number of terms in document j. The result will be a list of concepts together with a score marking the discriminatory power for each by document. The *cds* function can be seen graphically in Fig. 3, column V of Fig. 2 shows the *cds* calculations for the bankruptcy example.

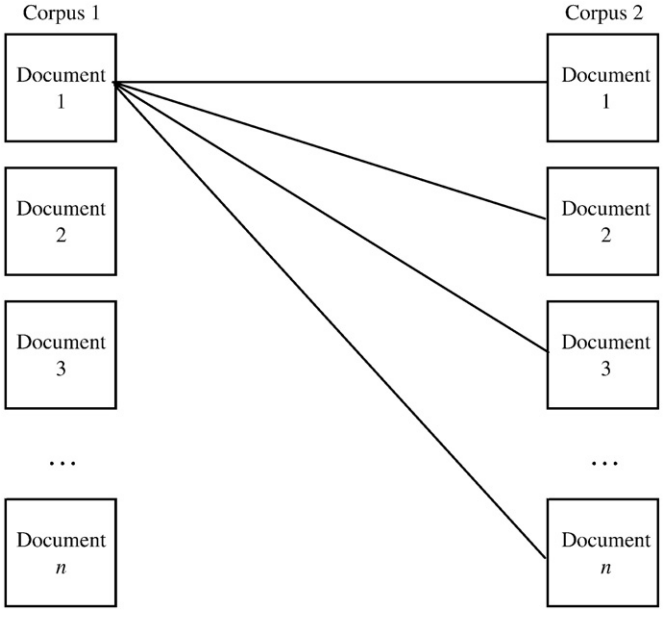

Fig. 3. Concept Document Score (*cds*).The *cds* for document 1 in Corpus 1 above is determined by comparing document 1 to all documents of Corpus 2. The best discriminators of Document 1 will have the highest *cds*. The contrastive corpus for Corpus 1 is Corpus 2 and the contrastive corpus for Corpus 2 is Corpus 1. This process is completed for all documents in both corpora.

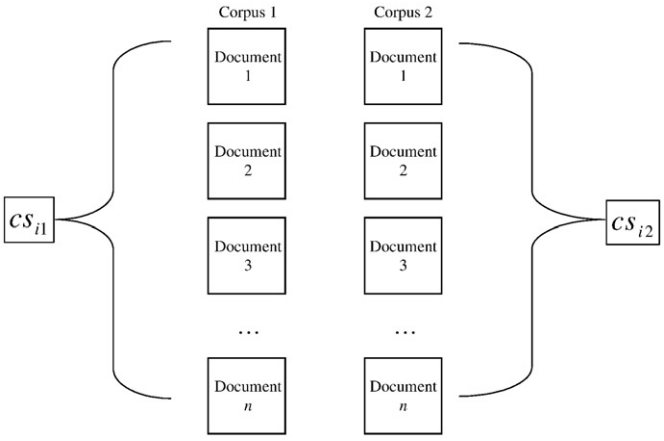

Fig. 4. Concept Score (*cs*). The Concept Score is determined by a function of the frequency of the concept in an individual document within the corpus and the number of documents the concept appears in corpus-wide. This measure lowers the value of concepts specific to a particular document (company) and raises the value of concepts that appear frequently in the entire corpus.

Step 2. The concept score pulls out the discriminatory concepts which are common among the documents within a corpus. This step is important because it gets rid of the terms which are special to individual documents in a corpus (such as company specific information) but do not help to discriminate between the two corpora. The function employs the *cds* and a *df* function. For this function, we want to increase the value of concepts that appear in more documents within the corpus. So those will be weighted higher. The function is shown below:

$$cs_{ik} = \sum_j cds_{ijk} * \log(n_k), k \in \{1,2\} \quad (3)$$

The highest cs_{ik} scores are compiled for each corpus. The scores from each corpus are merged and the top scores are kept to create a dictionary of discriminatory concepts. The cs_{ik} function can be seen graphically above in Fig. 4:
The final dictionary and *cs* for the bankrupt and nonbankrupt domains in our example from Fig. 2 are shown in Fig. 5 below. Note how tokens that are specific to a particular firm (e.g. automobile) end up with 0 weight, whereas tokens that are found across documents within the domain (e.g. net income for the nonbankrupt domain) are given the highest weights. In a real situation there will be much more nuance to the results. That is, there will be tokens that appear in both domains but are more prevalent in one than the other. There will also be tokens that appear in few documents within a domain, thus lowering (but not completely zeroing out) that token's *cs*.

3.5. Adding multi-word phrases

Multi-word phrases are important because phrases have more complete context information than individual concepts. For this reason we add multi-word phrases to our methodology as follows. Once the dictionary of individual concepts is created, we can find the multi-word phrases that have discriminatory power. We collect a term count of all two-word and three-word phrases and then follow the steps we performed above.[7] The process includes the cds_{ijk} and the cs_{ik} functions (this time we are looking at multi-word terms instead of

[7] The only difference between the single word process and the multi-word process is that we do not use WordNet for multi-word phrase WSD. When looking at multiple words, a context is built in so it is not necessary.

Concept Score (cs)	
Bankrupt	Nonbankrupt
Automobile - 2.993551 * log (1) = 0	Automobile – 1.584821 * log (1) = 0
Risk – 2.385606 * log (1) =0	Net Income – (5.248334 + 1.584821) * log(2) = 2.056985
Debt – 3.339849 * log (1) =0	Acquisition - 3.339849 * log (1) = 0
Net Loss – (1.431364 + 2.385606) * log(2) = 1.149022	Manufacturing – 2.385606 * log (1) = 0
Oil – 5.725455 * log (1) =0	Growth - 3.81697 * log (1) = 0
Loans –1.908485 * log (1) =0	

Fig. 5. Example of *cs*.

individual concepts). At the end of the process we have a score for each multi-word phrase. We merge the scores for the two and three-word phrases with the individual concept scores. The highest *cs* term may be a concept, a two-word phrase or a three-word phrase depending on its discriminatory power. The final result is a dictionary of concepts, two-word phrases and three-word phrases, ordered by each one's cs_{ik}. From here on, we will use the word "token" to refer to both concepts and multi-word phrases.

3.6. Token vector

The final step is to get a count of all tokens in the ontology for each document. The result is a vector of concepts of *n* dimensions, where *n* is a parameter we define. The ontology can be as small or as large as we like. The larger the ontology, the more information we give the optimizer, however, the greater the risk of overfitting. The smaller the ontology, the less information, however, the risk of overfitting is minimal. In the validation tests below we create vectors with counts[8] of 10, 20, 30, 40, 50, 100, 200, 300, 400, 500 and 1200. In the next section we test the usefulness of our methodology by testing its predictive ability (information content) on two important financial events, bankruptcy and management fraud.

4. Data, testing methodology and results

In order to test our methodology we gathered datasets for two important financial events, bankruptcy and fraud. Validation on two different financial events helps us showcase the methodology's potential. We chose bankruptcy and fraud because both have a catastrophic effect on shareholder value.

4.1. Data

Our bankruptcy dataset consists of MD&As for 78 companies that went bankrupt between 1994[9] to 1999 as well as a match set of 78 control firms. We gathered our bankrupt firms from Compustat/CRSP using the "reason for delist" code. We limited our set to industries within the 1 digit SIC codes 2 and 3. Our matched sample is a random sample of firms matched on year, 4 digit SIC code and Total Assets (control firm within 10% of total assets of match). The MD&As are gathered for the year prior to bankruptcy.

The fraud dataset includes 61 fraud companies and 61 nonfraud companies between 1993 and 2002. We gathered the fraud set by parsing Accounting and Auditing Enforcement Releases (AAERs) from the SEC website (www.sec.gov). For the fraud tests we gathered the MD&As for the year of the fraud event. We also gathered a set of 61 nonfraud companies using the same matching criterion that was utilized for the bankruptcy dataset. A listing of summary statistics of firms for both datasets is shown in Table 1 below.

Panel A of Table 1 shows the summary statistics for bankrupt and nonbankrupt firms. We report the mean, the standard deviation and the p-value of Total Assets, Sales, Market Value, Current Ratio and Return on Assets. The p-value determines if the summary statistics between bankrupt and nonbankrupt firms in our sample have statistically significant differences. At the 0.05 level we find only Return on Assets and Market Value as significantly different. Market Value tends to be lower for firms in poor financial health (such as those that are within a year of bankruptcy). The Return on Assets for bankrupt firms tends to be lower because the company is using assets inefficiently or operating at a loss.

Panel B of Table 1 shows the summary statistics for fraud and nonfraud firms. At the 0.05 level, the p-value for Total Assets, Sales and Market Value indicate that the samples are not significantly different. Return on Assets was significantly lower for fraud firms than for nonfraud firms. The Current Ratio was borderline at a 0.069 p-value. These differences are expected as fraud tends to occur when firms are under increased financial pressure.

We create a dictionary using all of the firms in the bankruptcy dataset. The top 20 tokens of the ontology, as measured by their concept scores, are shown below in Panel A of Table 2. We go through the same process for the fraud dataset. The top 20 tokens for fraud can be found below in Panel B of Table 2.

Panel A shows that "net income", "gross margin" and "research and development" are at the top of the list for discriminative power for bankruptcy. These are textual references to quantitative financial data appearing on financial statements. The top 7 *cs* are financial statement variables. This shows that the methodology is picking up relevant data. The word pair "risk uncertainty" scored as the 129th most discriminatory token. This was the first time risk came up in the ontology and is of lower relevance than would be expected for bankruptcy. This illustrates a benefit of utilizing this methodology as it gives discriminators (subject to the results of the validation tests below) free from any prior research biases.

Panel B shows that the highest weighted discriminator for fraud is "year end December". This does not mean that firms with a year end of December are more or less fraudulent. It means that the amount of times that the phrase "year end December" is written in an MD&A is a discriminator between fraud and nonfraud firms. The number 2 discriminator was "year end". This shows that the methodology is a heuristic and as such will make mistakes (such as scoring both "year end" and "year end December" highly). This could be potentially be corrected by deleting the lower scoring phrase (in this case "year end"). However, there is a risk to this when considering potentially relevant two and three-word phrases. Some documents may only have the two-word phrases and those phrases may help with prediction for those documents. The main goal of this dictionary in its present form is to aggregate the entire set of concepts for prediction. Having some errors is acceptable as long as prediction is still effective. The highest

[8] The count represents the size of the ontology. The top 10 concepts/multi-word phrases are in the vector of 10, the top 20 concepts/multi-word phrases are in the vector of 20, and so on.

[9] The cutoff of 1994 was chosen because electronic 10-Ks became available on EDGAR 1993 and our MD&A dataset is for the year prior to bankruptcy.

Table 1
Summary statistics.

Dataset	# of firms	Total assets		Sales		Market value		Current ratio		Return on assets	
		Mean	St. dev.	Mean	St. dev.	Mean	St. dev.	Mean	St. dev.	Mean	St. dev.
Panel A: bankruptcy											
Bankrupt	78	84.65	145.86	87.63	143.15	41.92	77.85	2.66	5.05	−0.88	2.40
Nonbankrupt	78	90.96	166.49	160.06	465.69	128.10	238.52	3.31	2.39	−0.15	0.94
	p-value	0.853		0.214		0.003		0.363		0.016	
Panel B: Fraud											
Fraud	61	17,001.19	86,450.58	4803.25	13,216.14	9386.95	27,603.36	1.68	0.91	−0.36	1.02
Nonfraud	61	15,188.55	72,743.74	3778.76	10,640.68	5717.65	15,710.87	3.46	6.97	0.00	0.34
	p-value	0.856		0.449		0.149		0.069		0.014	

scoring relevant multi-word phrase is number four, "research development expense". Prior fraud research has hypothesized that research and development expense is an account of concern. Next on the list were "interest income", "interest expense" and "gross profit". The perpetrators of fraud are trying to hide their fraud so it is not contrary to expectations that we do not find any obvious terminology. It is noteworthy that in both ontologies the top 20 tokens were multi-word phrases. This makes sense as the information content in phrases should be higher than that of singular concepts.

The individual token scores, however, tell just part of the story. Below we use these terms as attributes to build linear discriminant functions. The power of this method is the cumulative story that the aggregation of the tokens can tell about a company. The testing performed below finds linear discriminant functions that use up to 1200 tokens as attributes.

Table 2
Top 20 concepts/multi-word terms from bankruptcy (panel A) and fraud (panel B) ontologies (ordered by *cs*).

Panel A: bankruptcy			Panel B: fraud		
Number	Concept/multi-word phrase	Concept score (*cs*)	Number	Concept/multi-word phrase	Concept score(*cs*)
1	Net income	0.755	1	Year end December	0.319
2	Gross margin	0.709	2	Year end	0.295
3	Research development	0.572	3	Company have	0.291
4	Gross profit	0.566	4	Research development expense	0.270
5	General administrative expense	0.542	5	Interest income	0.256
6	Income tax	0.529	6	Interest expense	0.243
7	General administrative	0.511	7	Gross profit	0.243
8	Company have	0.488	8	Research development	0.237
9	Research development expense	0.461	9	Company expect	0.227
10	Percentage net sale	0.425	10	Net income	0.220
11	Year end	0.391	11	Net sale	0.198
12	Net sale	0.387	12	Liquidity capital resource	0.195
13	Interest expense	0.386	13	Capital expenditure	0.192
14	Company believe	0.380	14	Operate expense	0.184
15	See note	0.379	15	Cost sale	0.181
16	Year end December	0.379	16	Company believe	0.174
17	Interest income	0.371	17	Foreign currency	0.174
18	Company expect	0.370	18	Company plan	0.173
19	Net cash	0.367	19	Income tax	0.170
20	Fiscal year	0.348	20	Foreign currency exchange	0.169

4.2. Testing methodology

We validate the effectiveness of the ontology via a two step process: (1) we gather the number of times each token appears in the ontology and create a vector of token counts for all companies, and (2) we develop a discriminant function (using SVM discussed below) on the token count vectors that attempts to learn the token pattern that best separates the classes (fraud from nonfraud or bankrupt from nonbankrupt). We then test on a holdout set to see if the tokens discriminate between event and nonevent firms using the MD&As (see Fig. 6 below). If positive, the validation has two effects. First, it shows that the methodology we created can be used as a tool and is of potential interest to future researchers. Second, it confirms our expectation that there is information content in the MD&As that can help predict financial events.

We create two classification problems (one for the Bankruptcy data set and one for Fraud) in which the vector of token counts for all firms are labeled with a +1 for the nonbankruptcy (nonfraud) cases, and a −1 for the bankrupt (fraud) cases. We then use a classification method called the Support Vector Machine (SVM) to create a classifier (discriminant function) for each data set. The SVM function serves to further refine the concept score (*cs*).

SVM, a machine learning method based on statistical learning theory [32], produces a linear discriminant function. SVM explicitly balances empirical risk (the risk of incorrectly classifying something in the training set) with structural risk (the risk of overfitting using overly complex hypothesis spaces) — a theoretically important property that is lacking in many machine learning methods. Also, SVM has been shown to be successful in working with large feature spaces (number of concepts/multi-word phrases in our case) and small datasets. However, other methods that produce discriminant functions can be used in conjunction with our dictionary, such as Fisher's linear discriminant functions, logit and probit. More information about linear SVM can be found in [9]. In Table 3 we reproduce a ranking of the top negative (event) tokens and the top positive (nonevent) tokens after applying SVM to our dataset. The rankings of attributes are based on the product of the SVM coefficient and the attribute average.

The top "Bankruptcy" tokens/multi-word phrases include Gross Margin Decline, Inflation Impact Company, and Note Due. The top "Fraud" tokens/multi-word phrases include Additional Cost, Net Increase, and Cash Flow. These tokens are salient accounting phrases. For the bankruptcy domain, the phrases are clearly indicative of financial pressures. For the fraud domain, the token "Additional Cost" indicates a condition that produces greater financial pressure (the fraud triangle includes financial pressure as one of the conditions that supports management fraud).

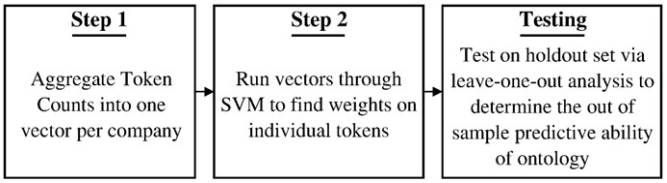

Fig. 6. Testing methodology.

We chose to use a leave-one-out analysis to test the effectiveness of our ontologies. Leave-one-out analysis[10] is an accepted method of testing when dealing with small datasets [14]. It is particularly well-suited for our task as we are trying to validate the dictionaries' effectiveness in discriminating between the two corpora for each domain (bankruptcy and fraud). Leave-one-out analysis works by creating different training and holdout combinations (called folds). In every fold the training set includes all but one company (that company is the holdout set). There are as many combinations as there are companies in the dataset. Every company is in the holdout set once. The reported results are the aggregation of the results of all tests. The rankings in Table 3 represent the average ranking across these leave-one-out runs.

4.3. Results

Table 4 reports the results of tests on all of our dictionary sizes. We report the dictionary size along with the overall accuracy and the Type I and Type II errors. A Type I error is recorded when an event is classified as nonevent (a fraud is classified as nonfraud or a bankruptcy is classified as a nonbankruptcy). A Type I error is of greater concern when considering catastrophic events such as bankruptcy or fraud. A Type II error is recorded for each nonevent classified as an event. Although we are less concerned with Type II errors, we would not find a model that classified all companies as event companies to be useful, even though it would have a 0% Type I error. If both error types are low then the function is discriminating between event and nonevent companies and the dictionary carries value relevant information.

The best overall out of sample results (using leave-one-out) for the bankruptcy dataset show 80% accuracy (see Panel A of Table 4). The Type I error and Type II error are balanced at approximately 20% each.[11] The result shows that using only the MD&As the methodology is able to correctly distinguish between firms that will go bankrupt in one year against firms that will not 80% of the time. This result validates the usefulness of the dictionary of concepts/multi-word terms as well. The best result was achieved using 1200 concepts/multi-word terms.

The best overall out of sample results for the fraud dataset was an accuracy of 75.41% (see Panel B of Table 4).[12] This result was achieved with 200 concepts/multi-word terms and with 500 concepts/multi-word terms. The solution with the lower Type I error would be the one we would choose as we are more concerned with mislabeling a bankrupt firm as a nonbankrupt firm than vice versa. The 500 feature results have a lower Type I error (21.31%).

In both datasets the holdout results show that the Type I error starts off very low at 10 features and then rises steadily with the

Table 3
Top negative and positive concepts/multi-word terms based on SVM model refinement: bankruptcy (panel A) and fraud (panel B) ontologies.

Panel A: bankruptcy		Panel B: fraud	
Negative (event) rank	Concept/multi-word phrase	Negative (event) rank	Concept/multi-word phrase
1	Gross margin decline	1	Additional cost
2	Company record charge	2	Cash generate
3	Impact inflation company	3	Net increase
4	Result gross profit	4	Remain decrease
5	Result gross	5	Cash flow
6	Company have recently	6	Revenue have
7	Interest expense year	7	Foreign exchange
8	Company continue effort	8	Goodwill amortization
9	Benefit	9	Company undertake
10	Note due	10	Forward-looking statement
Positive (nonevent) rank	Concept/multi-word phrase	Positive (nonevent) rank	Concept/multi-word phrase
1	Reference	1	Company plan
2	Inflation inflation	2	Discuss note
3	Inflation inflation have	3	Result include
4	Company complete secondary	4	Reference
5	Amendment	5	Increase relate
6	Management	6	Noninterest income
7	Decline net sale	7	Financing activity
8	Member	8	Increase sale
9	Decrease cash	9	Certain circumstance
10	Result completion	10	Product revenue

Table 4
Prediction results for bankruptcy and fraud datasets.

Ontology size	Overall accuracy	Type I error (# of event classified as nonevent)	Type II error (# of nonevent classified as event)
Panel A: bankruptcy			
10	55.48%	17.95% (14)	70.51% (55)
20	56.77%	24.36% (19)	61.54% (48)
30	59.35%	21.79% (17)	58.97% (46)
40	54.84%	29.49% (23)	60.26% (47)
50	56.13%	29.49% (23)	57.69% (45)
100	52.90%	39.74% (31)	53.85% (42)
200	63.23%	34.62% (27)	38.46% (30)
300	71.61%	26.92% (21)	29.49% (23)
400	78.71%	17.95% (14)	24.36% (19)
500	73.55%	26.92% (21)	25.64% (20)
1200	80.00%	19.23% (15)	20.51% (16)
Panel B: fraud			
10	55.74%	13.11% (8)	75.41% (46)
20	62.38%	32.79% (20)	29.51% (18)
30	54.92%	36.07% (22)	54.10% (33)
40	45.08%	49.18% (30)	60.66% (37)
50	53.28%	45.90% (28)	47.54% (29)
100	60.66%	37.70% (23)	40.98% (25)
200	75.41%	26.23% (16)	22.95% (14)
300	72.13%	24.59% (15)	31.15% (19)
400	68.85%	31.15% (19)	31.15% (19)
500	75.41%	21.31% (13)	27.87% (17)
1200	72.95%	26.23% (16)	27.87% (17)

[10] In a leave-one-out analysis the SVM is trained using the token vectors of every firm except the one that is left out for testing. If the firm could only be categorized by tokens that were unique to the firm (and thus not generalizable) then our leave-one-out accuracy would be low because the terms that discriminate the firm being tested would garner no weight in the trained SVM model (because those tokens would not be represented in the training data). In our results section we show that this is not the case. We are able to get accuracies of 75% and 80% on fraud and bankruptcy, respectively.

[11] We performed cost sensitive evaluation in SVM, weighting the bankrupt costs higher than nonbankrupt but found that the results are unchanged at all cost levels from 1:1 (bankrupt:nonbankrupt) to 1000:1. We have the same finding for the fraud dataset as well.

[12] Leave-one-out results are reported as the average accuracy of all the runs.

number of features to a peak (40 for fraud and 100 for bankruptcy) and then decreases afterwards. For low numbers of features, the results are volatile and often the SVM classifies nearly everything in one label. The low Type I error may be a result of the so called "swamping" effect for both datasets. As the number of features increase, true discrimination occurs as the Type I and Type II errors move together. In neither case was an obvious overfitting apparent.[13]

4.4. Substitute–complement tests

We further test the methodology by comparing it to quantitative methods used for both bankruptcy and fraud detection. This tests whether the information content of text can be a substitute for quantitative variables. We also add the quantitative variables to the text variables to see if they complement each other. The notion of complementarities between text and quantitative variables is a compelling one and has been observed by Tetlock et al. [33] as we noted earlier. If complementarities exist, then the text and the quantitative values are not focusing on the same aspects. Therefore, information is gained by aggregating the two.

We used models from seminal papers in fraud and bankruptcy research for our substitute–complement tests. For fraud, we used Beneish [5] and for bankruptcy we used Altman [2]. Beneish's [5] fraud discrimination ratios are as follows:

$$DSRI = \frac{\frac{Receivables_t}{Sales_t}}{\frac{Receivables_{t-1}}{Sales_{t-1}}} \quad GMI = \frac{\frac{Sales_{t-1} - CostofGoodsSold_{t-1}}{Sales_{t-1}}}{\frac{Sales_t - CostofGoodsSold_t}{Sales_t}}$$

$$AQI = \frac{\frac{1 - CurrentAssets_t + PP\&E_t}{TotalAssets_t}}{\frac{1 - CurrentAssets_{t-1} + PP\&E_{t-1}}{TotalAssets_{t-1}}} \quad SGI = \frac{Sales_t}{Sales_{t-1}}$$

$$DEPI = \frac{\frac{Depreciation_{t-1}}{Depreciation_{t-1} + PP\&E_{t-1}}}{\frac{Depreciation_t}{Depreciation_t + PP\&E_t}} \quad SGAI = \frac{\frac{Selling, General, AdmExp_t}{Sales_t}}{\frac{Selling, General, AdmExp_{t-1}}{Sales_{t-1}}}$$

$$LVGI = \frac{\frac{LongTermDebt_t + CurrentLiabilities_t}{TotalAssets_t}}{\frac{LongTermDebt_{t-1} + CurrentLiabilities_{t-1}}{TotalAssets_{t-1}}}$$

$$TATA = \frac{\Delta CurrentAssets_t - \Delta Cash_t - \Delta CurrentLiabilities_t - \Delta CurrentMaturitiesofLTD_t - \Delta IncomeTaxesPayable_t - DepreciationandAmortization}{TotalAssets_t}$$

Altman's bankruptcy discrimination function is as follows:

$$AltmanZScore = \beta_1 \frac{WorkingCapital}{TotalAssets} + \beta_2 \frac{retainedEarnings}{TotalAssets} + \beta_3 \frac{EBIT}{TotalAssets} + \beta_4 \frac{MarketValueofEquity}{TotalLiabilities} + \beta_5 \frac{Sales}{TotalAssets}$$

[13] Overfitting can occur when the number of features greatly outnumbers the number of companies in the dataset (as is the case with some of our experiments). Overfitting is discovered by analyzing the results of a holdout set. If the classifier overfits, the holdout set accuracy will be low while the training accuracy is high. Our results are not low on our holdout set (via leave-one-out analysis).

We constructed the Altman ratios and set them up as the vector components and tested them in SVM using leave-one-out analysis (as we did with our other tests). In order to keep the entire sample intact, we replaced missing values with their mean value for the entire sample [22]. Our tests were performed using SVM to create the discriminant function in order to maintain a uniformity of testing. We found that Altman's ratios were able to classify bankruptcy data correctly 66.67% of the time (see Table 5, Panel A). The highest accuracy achieved using text alone was 80.00% (from Table 4, Panel A). This is further evidence of the value of the information content of financial text. We combined the vectors formed from text with the Altman ratios and ran experiments with all of the dictionary sizes (10, 20, 30…1200). The results are reported in Table 5, Panel B. The highest overall accuracy was achieved with the dictionary of size 1200 combined with the Altman ratios (83.87%). This was a 4.84% improvement over the text alone and a 25.8% improvement over Altman alone. We also report the complementarities in Table 5, Panel B (where complementarities are achieved when the combined results are greater than the highest individual result). Complementarities were achieved on all dictionaries of size 300 or greater.

We constructed the Beneish ratios and tested the fraud dataset in the same manner. Beneish was able to correctly classify the dataset 40.16% of the time. Given that this is a 1:1 matched dataset, that is worse than random prediction and considerably lower than the highest accuracy achieved with text alone (75.41%). This suggests that there is more information in the text that is helpful in classifying the fraud firms in our sample than there is in the quantitative financial variables put forth by Beneish. We performed the complementarities tests combining Beneish ratios with all dictionary sizes.

Table 5
Substitute/complement tests using Altman (1968) ratios for bankruptcy and Beneish (1999) ratios for fraud.

Ontology size	Overall accuracy	Type I error (# of event classified as nonevent)	Type II error (# of nonevent classified as event)	Complementary results (+), same (0), worse (−)
Panel A: Altman alone				
	66.67%	42.3% (33)	24.4% (19)	
Panel B: text + Altman				
10	63.23%	31.2% (24)	42.3% (33)	−
20	58.06%	39.0% (35)	44.9% (30)	−
30	59.35%	37.7% (29)	43.6% (34)	−
40	53.55%	44.2% (34)	48.7% (38)	−
50	59.35%	39.0% (30)	42.3% (33)	−
100	60.00%	36.4% (28)	43.6% (34)	−
200	61.94%	35.1% (27)	41.0% (32)	−
300	75.48%	20.8% (16)	28.2% (22)	+
400	80.65%	19.5% (15)	19.2% (15)	+
500	77.42%	22.1% (17)	23.1% (18)	+
1200	83.87%	15.6% (12)	16.7% (13)	+
Panel C: Beneish alone				
	40.16%	73.8% (45)	45.9% (28)	
Panel D: text + Beneish				
10	59.02%	26.2% (16)	55.7% (34)	+
20	55.74%	41.0% (25)	47.5% (29)	−
30	53.28%	45.9% (28)	47.5% (29)	−
40	50.00%	52.5% (32)	47.5% (29)	+
50	65.57%	37.7% (23)	31.1% (19)	+
100	62.30%	41.0% (25)	34.4% (21)	+
200	74.59%	21.3% (13)	29.5% (18)	−
300	76.23%	21.3% (13)	26.2% (16)	+
400	68.85%	29.5% (18)	32.8% (20)	0
500	75.41%	24.6% (15)	24.6% (15)	0
1200	81.97%	19.7% (12)	16.4% (10)	+

Although the Beneish ratios alone did not add value, the Beneish ratios plus text (ontology size 1200) achieved the greatest overall accuracy (81.97%). This is 8.70% higher than the text results alone. This indicates that combining text with quantitative financial variables serves to create a more powerful classifier. Complementarities were achieved for six of the dictionary sizes. See Table 5, Panel D for details.

5. Conclusion

In this paper we develop a methodology for automatically analyzing financial text. We make several interrelated contributions. We develop a methodology to detect financial events. We extend technical research in computational linguistics via the dictionary creation method. This method extends the VSM model to include WordNet and tunes the weights to discriminate between documents from two different domains (event and nonevent).

To validate the methodology empirically, we test it on two financial events, bankruptcy and fraud. We find that the dictionaries we create are able to discriminate fraud from nonfraud MD&As 75% of the time and bankrupt from nonbankrupt MD&As 80% of the time. We compare our results with quantitative prediction methods (Beneish for fraud and Altman for bankruptcy). Our methodology achieved superior results using the same data. We tested for complementarities by merging the quantitative data with the text data. We achieved the best results for both bankruptcy (83.87%) and fraud (81.97%) with the combined data. This shows that the text of the MD&A contains information that is complementary to the quantitative information. Using both quantitative and textual information improved the prediction of both financial events. The methodology could be used to help investors screen out companies at risk for bankruptcy. It could also be used by government and regulatory bodies as a tool to help determine the firms that may be committing fraud, and thus be considered for auditing. The methodology can be applied to available text for any financial problem where the goal is to create a dictionary (ontology) of discriminating concepts.

The potential future directions for this research are numerous. The methodology designed in this paper is aimed as a foundation to be applied to a variety of financial phenomena that include text. In a post Sarbanes Oxley (http://www.soxlaw.com) world, the disclosure requirements for public companies continue to ratchet up. “Reading between the lines of the text” from these required financial reports will give researchers a new toolkit. The potential uses go beyond required financial reports and can include optional financial information as well (such as conference call transcripts, press releases, and press about the company). Where much prior qualitative analysis required manual coding of pre-hypothesized variables, this methodology is automatic. Future research can include gaining insight into the keywords or phrases in financial text that lead to future changes in company valuation (big gains and losses). Finding the firms that were correctly classified by one method (quantitative or text) but not another could give accounting/financial researchers valuable information (e.g., the characteristics of firms that report financial and qualitative information differently). This could be explored more thoroughly in future research. The ability to quantify text information and place alongside financial information allows researchers to extend previous econometric models that only included quantitative financial information.

Improvements to the methodology could be made to reduce noise in the ontology and improve prediction accuracy. A natural methodological extension would be to further develop the methodology so that it can be useful in analyzing continuous variables (as opposed to binary outcomes) such as are commonly researched in accounting. A possible addition to the methodology would be the inclusion of a temporal component to track changes to the structure of text over time for individual firms. Such an extension would allow researchers to better understand how firm changes are manifest in financial texts. The method could be further developed to learn the context of multi word phrases via WSD.

Acknowledgements

We gratefully acknowledge Scott Jackson, Tom Lopez, Stephen Brown and John Core for their helpful comments.

References

[1] C. Aasheim, G.J. Koehler, Scanning World Wide Web documents with the vector space model, Decision Support Systems 42 (2006) 690–699.
[2] E. Altman, Financial ratios, discriminant analysis and the prediction of corporate bankruptcy, Journal of Finance 23 (1968) 193–194.
[3] W. Antweiler, M.Z. Frank, Is all that talk just noise? The information content of internet stock message boards, The Journal of Finance LIX (3) (2004) 1259–1294.
[4] R. Ball, S.P. Kothari, R.L. Watts, Economic determinants of the relation between earnings changes and stock returns, The Accounting Review 68 (1993) 622–638.
[5] M. Beneish, The detection of earnings manipulation, Financial Analysts Journal 55 (1999) 24–36.
[6] E. Brill, A simple rule-based part of speech tagger, Proceedings of the Third Conference on Applied Natural Language Processing, 1992, pp. 152–155, Trento, Italy.
[7] P. Buitelaar, B. Sacaleanu, Extending synsets with medical terms, Proceedings of the First International WordNet Conference, Mysore, India, January 21–25, 2002.
[8] C.W. Choo, Information Management for the Intelligent Organization: The Art of Scanning the Environment, Information Today, Inc., Medford, NJ, 1995.
[9] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines and other Kernel-Based Learning Methods, Cambridge University Press, Cambridge, United Kingdom, 2000.
[10] S.R. Das, M.Y. Chen, Yahoo! For Amazon: Sentiment Extraction from Small Talk on the Web, Management Science 53 (9) (2007) 1375–1388.
[11] A.K. Davis, J.M. Piger, L.M. Sedor, “Beyond the Numbers: Managers' Use of Optimistic and Pessimistic Tone in Earnings Press Releases” AAA 2008 Financial Accounting and Reporting Section (FARS) PaperAvailable at SSRN:, http://ssrn.com/abstract=8753992008.
[12] Patricia M. Dechow, Weili Ge, Chad R. Larson, Richard G. Sloan, “Predicting Material Accounting Manipulations” AAA 2008 Financial Accounting and Reporting Section (FARS) PaperAvailable at SSRN:, http://ssrn.com/abstract=9974832007.
[13] John J. Gerdes, EDGAR-Analyzer: automating the analysis of corporate data contained in the SEC's EDGAR database, Decision Support Systems, 35, 2003, pp. 7–29.
[14] C. Goutte, Note on free lunches and cross-validation, Neural Computation 9 (1997) 1245–1249.
[15] C. Holton, Identifying disgruntled employee systems fraud risk through text mining: a simple solution for a multi-billion dollar problem, Decision Support Systems 46 (4) (2009) 853–864.
[16] D. Jurafsky, J. Martin, Speech and Language Processing, Prentice-Hall, Inc., Upper Saddle River, NJ, 2000.
[17] L. Khan, F. Luo, Ontology Construction for Information Selection, International Conference on Tools with Artificial Intelligence (2002) p. 122.
[18] D.D. Lewis, R.E. Schapire, J.P. Callan, R. Papka, Training algorithms for linear text classifiers, Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1996, pp. 298–306.
[19] Feng Li, Do stock market investors understand the risk sentiment of corporate annual reports? University of Michigan Working Paper, 2006.
[20] Feng Li, Annual report readability, current earnings, and earnings persistence, Journal of Accounting and Economics 45 (2–3) (2008) 221–247.
[21] N. Li, D.D. Wu, Using text mining and sentiment analysis for online forums hotspot detection and forecast, Decision Support Systems 48 (2010) 354–368.
[22] R.J.A. Little, D.B. Rubin, Statistical Analysis with Missing Data, John Wiley & Sons, New York, 1987.
[23] Andrew McCallum, Bow: a toolkit for statistical language modeling, text retrieval, classification and clustering, Working Paper, School of Computer Science, Carnegie-Mellon University, 1996.
[24] Z. Ma, O. Sheng, G. Pant, Discovering company revenue relations from news: a network approach, Decision Support Systems 47 (2009) 408–414.
[25] R. Navigli, P. Velardi, Automatic adaptation of WordNet to domains, Proc. of 3rd International Conference on Language Resources and Evaluation, Las Palmas, Canary Island, Spain, 2002.
[26] Princeton University, WordNet (2008) http://www.cogsci.princeton.edu/~wn/index.shtml.
[27] Z. Qu, H. Zhang, H. Li, Determinants of online merchant rating: content analysis of consumer comments about Yahoo merchants, Decision Support Systems 46 (2008) 440–449.
[28] Sarkar, A., Computational linguistics (course notes), 2004, http://www.sfu.ca/anoop/courses/CMPT-413-Spring-2004/lexicalsem.pdf, January, 2005.
[29] K. Sparck Jones, A statistical interpretation of term specificity and its application in retrieval, Journal of Documentation 28 (1) (1972) 11–21.

[30] P. Tetlock, Giving content to investor sentiment: the role of media in the stock market, Journal of Finance 62 (2007) 1139–1168.
[31] P.C. Tetlock, S.-T. Maytal, S. Macskassy, More Than Words: Quantifying Language to Measure Firms' Fundamentals, Working Paper, , 2007.
[32] V. Vapnik, Statistical Learning Theory, Springer Verlag, New York NY, 1995.
[33] C.P. Wei, Y.H. Lee, Event detection from online news documents for supporting environmental scanning, Decision Support Systems 36 (2004) 385–401.

Mark Cecchini (PhD, University of Florida 2005), is an assistant professor in the School of Accounting at the Darla Moore School of Business at the University of South Carolina. He teaches accounting information systems and information systems strategy to both undergraduate accounting students and students in the Master of Accountancy Program. Professor Cecchini conducts research at the intersection of accounting and information systems as well as capital market research and analytical management accounting research. Mark has published his research in Management Science and Journal of Accounting and Economics. Mark has presented his research at the Annual Meeting of the American Accounting Association and the INFORMS annual meeting. Professor Cecchini has served as an ad hoc reviewer for The Accounting Review, Auditing: A Journal of Practice and Theory, Decision Support Systems, and Journal of Information Technology Management.

Haldun Aytug is an associate professor in the Warrington School of Business at the University of Florida. Dr. Aytug's current research focuses on machine learning, and e-commerce. His research has been funded by the National Science Foundation, Intel Corporation and Applied Materials. He has published in INFORMS Journal on Computing, Information Systems Research, Decision Support Systems, and European Journal of Operational Research among others.

Gary J. Koehler is the John B. Higdon Eminent Scholar of Management Information Systems at the University of Florida. He received his PhD. from Purdue University in 1974. He has held academic positions at Northwestern University and Purdue University and between 1979 and 1987 was a cofounder and CEO of a high-tech company which grew to over 260 employees during that period. His research interests are in areas formed by the intersection of the Operations Research, Artificial Intelligence and Information Systems and include such topics as genetic algorithm theory, machine learning, e-commerce, quantum computing and decision support systems. He has published in journals including *Management Science*, *Operations Research,* Informs *Journal on Computing*, *Evolutionary Computation*, *Decision Sciences, Decision Support Systems, the European Journal on Operational Research, the Journal of Management Information Systems, Information Systems and e-Business Management, SIAM Journal on Control and Optimization, Discrete Applied Mathematics, Journal of Finance*, and others. He is an area editor for *Decision Support System* and is on several other editorial boards. He has served as an expert witness for many large firms (including AT&T), has been an External Examiner for several Universities and has worked under grants from IBM and the National Science Foundation.

Dr. Praveen Pathak is an Associate Professor of Decision and Information Sciences at the Warrington College of Business at the University of Florida. He received his PhD in Information Systems from the Ross School of Business, University of Michigan, Ann Arbor, in 2000. He also holds an MBA (PGDM) from the Indian Institute of Management, Calcutta, and an Engineering degree, B. Tech. (Hons.), from the Indian Institute of Technology, Kharagpur. His research interests include information retrieval, web mining, offshore outsourcing and business intelligence. His research has appeared in many journals such as Journal of Management Information Systems (JMIS), Decision Support Systems (DSS), IEEE Transactions on Knowledge and Data Engineering (TKDE), Information Processing and Management (IP&M), Journal of the American Society for Information Science and Technology (JASIST), and in leading information technology conferences such as ICIS, HICSS, WITS, and INFORMS.