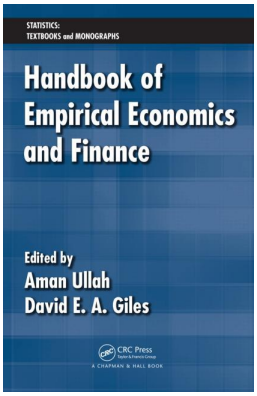

Handbook of Empirical Economics and Finance

Ullah Aman, E. A. Giles David

An Introduction to Textual Econometrics

Publication details
https://test.routledgehandbooks.com/doi/10.1201/b10440-6
Fagan Stephen, Gençay Ramazan
Published online on: 20 Dec 2010

5

An Introduction to Textual Econometrics

Stephen Fagan and Ramazan Gençay

CONTENTS

5.1 Introduction

> If we are not to get lost in the overwhelming, bewildering mass of statistical data that are now becoming available, we need the guidance and help of a powerful theoretical framework. (Frisch 1933)

This quote from Ragnar Frisch's Editor's Note in the first issue of *Econometrica* in 1933 has never seemed more timely. The phrase "data rich, information poor" is often used to characterize the current state of our digitized world. Over recent decades, data storage and availability has been growing at an exponential rate, and currently, data sets on the order of terabytes are not uncommon. While a portion of this new data is in the form of numerical or categorical data in well-structured databases, the vast majority is in the form of unstructured textual data. These news stories, government reports, blog entries, e-mails, Web pages, and the like are the medium of information flow

throughout the world. It is this unstructured data that most decision-makers turn to for information.

In the context of numerical and categorical data, Frisch's desire for powerful tools for data analysis has, to a great extant, been satiated. The field of econometrics has expanded at a rate that has well matched the increasing availability of numerical data. However, in the context of the vast amount of textual data that has become available, econometrics has barely scratched the surface of its potential. Of course, the problems involved in the analysis of textual data are much greater than those of other forms of data. The complexity and nuances of language, as well as its very-high-dimensional character, have made it an illusive target for quantitative analysis. Nevertheless, decision-making is at the core of economic behavior, and given the importance of textual data in the decision-making process, there is a clear need for more powerful econometric tools and techniques that will permit the important features of this data to be included in our economic and financial models.

While textual econometrics is still early in its development, some important progress has been made that has empirically demonstrated the importance textual data. Research has shown that even at a coarse level of sophistication, automated textual processing can extract useful knowledge from large collections of textual documents. Most of this new work has taken one of two approaches: either the textual data have been greatly simplified into a small number of dimensions so that traditional econometric techniques can be brought to bear, or new techniques have been adopted that are better able to deal with the complexity and high-dimensional character of the data. Both approaches have proven useful and will be discussed in this chapter.

This early work is important for two reasons: First, it has shown that textual data is an economically significant source of information and is not beyond quantitative analysis. Second, while the techniques employed in these studies can capture only a fraction of the linguistic sophistication contained in the documents, they do offer good baselines against which we can compare future work. The development of linguistically sophisticated techniques is a multidisciplinary endeavor pursued by researchers in linguistics, natural language processing, text mining, and other areas. New techniques are constantly being developed, and their application in econometrics will need to be tested to determine whether an increase in complexity is made worthwhile by an increased ability to quantify the information embedded in textual data.

The aim of this chapter is to introduce the reader to this new field of textual econometrics. It proceeds as follows: Section 5.2 provides a review of some recent applications of textual analysis in the economics and finance literature, Section 5.3 describes some special properties of textual data, Section 5.4 identifies various sources of textual data, Sections 5.5 through 5.7 describe the important tasks of preprocessing, creating feature vectors, and reducing dimensionality. Section 5.8 considers the application of automated document classification. Section 5.9 directs readers to popular software that can process textual data, and Section 5.10 concludes. Throughout the chapter, references

are given for further reading and public resources that will be useful for the interested researcher.

5.2 Textual Analysis in Economics and Finance

Since the introduction of event studies into economics (Fama et al. 1969), researchers have been investigating the economic impact of media releases on markets (Chan 2003; Mitchell and Mulherin 1994; Niederhoffer 1971). While manual classification of stories as positive or negative was used in many such studies, very little of the actual information content could be extracted for use in statistical analysis. To a large degree, it was often the fact that there was new information, not the information itself, that was being analyzed.

The mere fact that communication is occurring, either to or between market participants, can be useful in understanding markets. For example, Coval and Shumway (2001) look at the volume of noise in a Chicago Board of Trade futures pit and found that following a rise in the sound levels, prices become more volatile, depth declines, and information asymmetry increases. In the written domain, Wysocki (1999) examines the relationship between posting volume on Internet stock message boards and stock market activity, finding that market characteristics determine posting levels and at the same time, posting levels predict future volume and returns.

Using more linguistic content, Antweiler and Frank (2004) investigate whether there is any predictive information in Internet stock message boards. Having manually classified a training set of 1000 messages as either BUY, SELL, or HOLD, they use the Naive Bayes classification algorithm to classify a larger set of 1.5 million messages from Yahoo!Finance and Raging Bull. The classifier accepts the document as a "bag of words" without retaining any linguistic structure, and uses simple word frequencies to decide on an appropriate indicator. By aggregating these indicators within each period, they construct a bullishness sentiment indicator that has predictive power for market volatility, but not for returns.

While Antweiler and Frank (2004) use all of the words (but not their linguistic relationships) of their documents, they ultimately reduce the dimensionality of textual data significantly by identifying the tone or sentiment of documents. Similarly, Tetlock, Saar-Tsechansky, and Macskassy (2008) create a measure of news sentiment by identifying the fraction of negative words in over 350,000 firm-specific news stories from the *Wall Street Journal* and the Dow Jones News Service.[1] Using the traditional regression methodology, they find that the information captured by this simple variable has predictive power for earnings and equity returns.

These two papers represent the two approaches to textual data: either it has been greatly simplified into a small number of dimensions so that traditional

[1] Their source for the news stories was the Factiva database (www.factiva.com).

econometric techniques can be brought to bear, or new techniques have been adopted that are better able to deal with the complexity and high-dimensional character of the data.

Bhattacharya et al. (2008) manually read and classified 171,488 news stories written during the Internet IPO frenzy to test whether the media "hyped" Internet stocks by giving them more positive news relative to a group of non-Internet IPOs during the same period. They find that there was indeed significant media hype, but after controlling for other return-related factors, find that the aggregated news affect from a given day would last only two days and explained only a small portion of the difference between Internet and non-Internet IPO returns.

Tetlock (2007) looks for interactions between the content of a popular *Wall Street Journal* column and the stock market over a 16-year period. After reducing the dimensionality of the textual data through dictionary classifications and principal component analysis, he uses a vector autoregressive methodology to identify the relationship between media pessimism and market returns and volumes. Findings indicate that the WSJ column does impact returns and volume, and that his textual variable depends on prior market activity.

Like Antweiler and Frank (2004), Das and Chen (2007) study the impact of stock message boards on market characteristics. Rather than using a single classification algorithm to label messages as BUY, SELL, or HOLD, they employ five different classification algorithms which get to vote for the final label. These classifiers use different data sets, extracted using various grammatical parsing and statistical techniques, to decide on a label. Within the forecasting literature, combining forecasts has been found to improve forecasting performance (Armstrong 2001; Bates and Granger 1969), and compared to the Bayes text classifier, the combined algorithms have better classification accuracy.

Financial economics is not the only field doing textual econometrics, and in fact only a minority of the work to date has been published in economic or finance journals. The majority of the work has been done in the area of Knowledge Discovery in Databases (KDD). KDD is a multidisciplinary field (drawing primarily on developments of computing sciences) whose goal is to extract nontrivial, implicit, previously unknown, and potentially useful information from databases. A critical step in the KDD process often uses data mining[2] techniques to extract hidden patterns from data. In fact KDD and data mining are already used in many economic applications including marketing, fraud detection, and credit scoring, just to name a few.

Economic and financial data are popular for KDD research because of their abundance and the complex relationships within large economic datasets. A large portion of this research focuses on stock market forecasting since the difficulty of this task is well established. Consequently, techniques that work in this area are likely to represent true innovations. Examples of this work

[2] While the phrase "data mining" has negative associations of "data snooping" within the field of econometrics, the term also refers to a respected, well-developed field of computing science.

include that by Mittermayer (2004), Fung, Yu, and Lu (2005), Kroha, Baeza-Yates, and Krellner (2006), Rachlin et al. (2007), and Schumaker and Chen (2009). Often, this research is more data-driven than work in economics with, for example, features chosen for their predictive ability (e.g., associations with trending prices) rather than their linguistic properties (e.g., negative affectivity). This may be an important difference since, for example, markets can fall on "good news" if the news was not as "good" as expected.

Beyond market forecasting, there has been a small amount of research on the impact of textual data on macroeconomic conditions (Gao and Beling 2003) and in the area of labor laws (Ticom and de Lima 2007). However, there are many other areas of economics and finance where the effects of textual information may prove illuminating. Such areas include corporate finance, bankruptcy and default, public policy, consumer behavior, among others.

5.3 Properties of Textual Data

Many researchers and philosophers have argued that human language and human intelligence are intimately linked (Dennett 1994). In fact, one of the long-standing proposals for testing genuine artificial intelligence is whether a computer could engage a human in conversation with sufficient ability that the human cannot tell, from language use alone, whether they are conversing with a human or not (Turing 1950). While this level of automated linguistic ability is still a distant goal, many less ambitious tasks have been automated through the exploitation of structural and statistical regularities of language (Jurafsky and Martin 2000). Clearly, the research outlined in the previous section was not based on a deep level of automated understanding of language, and yet many of the results are interesting and useful. On the one hand, this indicates that there is still much to be gained from further advances in language sciences and textual econometrics. However, it is necessary to be aware of the challenges that textual data presents.

The primary function of language is to encode information that other language users can extract. This is achieved through the sequential ordering of linguistic primitives, either simple sounds (phonemes) in spoken language or written symbols. In the written domain, these symbols are combined to form words, and these words combine to form phrases and sentences, which combine to form larger linguistic entities. The word is the smallest meaningful unit, and much of the current textual econometric research deals with language at the level of words, ignoring the important structural relationships between words.

While words are the smallest syntactic unit, we are ultimately interested in the intended meaning of the word. However, there is a many-to-many relationship between words and meanings. The fact that a word can have more than one meaning (or sense, as they are sometimes called) is referred to as *polysemy*. On average, English words have 1.4 meanings, with some words having

more than 70 (Fellbaum 1998). In addition to multiple meanings, words can also have multiple linguistic functions, or part-of-speech (PoS). Thus, using words as independent variables rather than their intended sense is akin to introducing measurement error into a model.

On the other side of the word-meaning relationship, a sense/meaning can often be expressed with more than one word. Two words sharing a common meaning are called *synonyms*, and in the English language, an average of 1.75 words express a single meaning (Fellbaum 1998). This phenomenon is intensified by the fact that most of us are taught not to be repetitive in our writing style, and at the same time when writing about a given topic, a small set of senses will be used multiple times. This means that synonyms can exhibit strong correlations, and lead to problems akin to multicollinearity.

Moving up to the level of sentences creates another layer of language-to-meaning difficulties. Ambiguity is one such difficulty. For example, consider the sentence "Jill saw Jack with binoculars." Does Jill or Jack have the binoculars? The prepositional phrase "with binoculars" is not unambiguously attached to either Jill of Jack. The use of metaphors and sarcasm, which are not meant to be taken literally, is also difficult to account for. Another, though not so pressing, difficulty is the fact that languages evolve over time, and grammar rules are not always strictly obeyed. Overcoming these difficulties is actively being pursued by language researchers, and there have been many developments including part-of-speech identification, grammatical parsing, word-sense disambiguation, automated translation, and others.

Such difficulties have led researchers to work with textual data at the word level. This approach, sometimes called the "bag-of-words" approach, treats documents as unordered sets of words and ignores the sequential and grammatical structure. The assumption that word occurrences are independent features of a document is clearly false and results in the loss of the vast majority of the information contained in the document. However, this assumption has been defended on pragmatic grounds. As Fung, Yu, and Lu (2005) state,

> Research shows that this assumption will not harm the system performance. Indeed, maintaining the dependency of features is not only extremely difficult, but also may easily degrade the system performance. (page 6)

Antweiler and Frank (2004) use the same defense:

> As an empirical matter it has been found that a surprisingly small amount is gained at substantial cost by attempting to exploit grammatical structure in the algorithms. (page 1264)

Simple word choice can be a useful predictor of a document's tone and the author's sentiment. That is, words alone can capture some of the emotive content of the text. Words also permit researchers to capture documents on a given subject. However, much is lost, and as advances are made in the language sciences, we should encourage the use of new, performance-enhancing

techniques. Some of these advances are discussed in later sections of this chapter.

Given the importance of words in current textual econometrics, it is important to consider some of their stylized facts. To begin, the distribution of words in a natural language can be approximated by a Zipfian (a.k.a. Yale) distribution (Zipf 1932). This characterization of language is usually referred to as Ziph's law or the rank-size law and it states that the frequency (f) of a word is inversely proportional to its statistical rank (r). Thus, within a large corpus (i.e., a large collection of texts) there is a constant k such that

$$f \cdot r = k \tag{5.1}$$

Graphically, Ziph's law predicts that a scatter plot of $\log(f)$ against $\log(r)$ will form a straight line with a slope of -1. A consequence of this property is that in a natural language, there are a few very frequent words, a relatively small group of medium frequency words, and a very large number of rarely occurring words. For example, in the Brown Corpus, consisting of over one million words, half of the word volume consists of repeated uses of only 135 words.

Zipf's law has many implications for the statistical properties of textual data. For example, observational data on word occurrence will be very sparse, with only a few words having many examples. This can impact classification and prediction problems since even in large collections of documents, there can be words that occur in only a single document. Thus, classification algorithms must be robust to overfitting since, otherwise, training documents will be classified according to their unique words.

While Zipf's law is fairly accurate over most corpora, Mandelbrot (1954) noted that the fit is poor for both very-high- and very-low-frequency words, and proposed the following alternative characterization of the frequency-rank relationship:

$$f = P(r + \rho)^{-B} \tag{5.2}$$

where P, ρ, and B are parameters describing the use of words in the text. Both Mandelbrot's and Zipf's characterization of the distribution of words is consistent with Ziph's argument that language properties develop as an efficient compromise that minimizes the efforts of listeners (who prefer large vocabularies to reduce ambiguity) and speakers (who prefer smaller vocabularies to reduce effort). In a similar argument, Zipf proposes that the number of meanings (m) of a word is related to its frequency by

$$m \propto \sqrt{f} \tag{5.3}$$

or, given Zipf's law,

$$m \propto \frac{1}{\sqrt{r}}. \tag{5.4}$$

An important partitioning of a lexicon (i.e., a dictionary or collection of words) is between "content" words and "function" words. Function words are (usually little) words that have important grammatical roles, including determiners (e.g., the, a, that, my, ...), prepositions (e.g., of, at, in, ...), and others. There are relatively few of these words (around 300 in English), but they occur very frequently. Content words, on the other hand, constitute the vast majority of words in a language. They are the nouns (e.g., Jim, house, question, ...), adjectives (e.g., happy, old, slow, ...), and full verbs (e.g., run, grow, save, ...), and others that present the informational content of texts. As expected these types of words have very different distributional properties.

While function words appear to occur fairly uniformly throughout documents, content words appear to cluster. Zipf (1932) noted this phenomenon by examining the distance (D) between occurrences of a given word, and then calculating the frequency (F) of these distances. He found that the number of observations of a given distance between word occurrences was inversely related to the magnitude of the distance. That is,

$$F \propto \frac{1}{D^p} \tag{5.5}$$

for values of p around 1.2. This implies that most content words occur near other occurrences of the same word. Thus, content words have persistence over time.

Several models have been proposed to model word distributions. Zipf's observation about the persistence of content words makes the Poisson distribution a poor choice because of its assumption of independence between word occurrences. Better models include mixture models of multiple Poissons and the K mixture model proposed by Katz (1996). For an expanded discussion of word distribution models, and other statistical properties of natural languages, interested readers should read the text by Manning and Schutze (1999).

5.4 Textual Data Sources

The first step in any textual analysis project is the collection of the relevant data. Textual econometrics will rarely, if ever, exclusively use textual data, but will combine textual data with other more traditional data types. The specific source for textual data will vary with the project, but there are many resources that researchers in the area should be aware of.

News sources are of particular importance for textual econometrics since markets, as information aggregators, will move when new and relevant information is released. A widely used, though fairly old, set of news stories is

called *Reuters Corpus Volume 1*, or *RCV1*. It has been documented by Lewis et al. (2004) and can be obtained from the National Institute of Standards and Technology.[3] *RCV1* contains about 810, 000 Reuters English language news stories from 1996-08-20 to 1997-08-19. A multilingual version, *RCV2*, is also available. Another source for collected textual data sets, or corpora as they are often called, is the Linguistic Data Consortium.[4] These sources contain relatively old material because of copyright issues, and researchers will likely want to purchase or collect more recent data. A commercial source of textual data that has been used in econometric research is the Dow Jones Factive[5] group, which collects news from over 25, 000 sources including the *Wall Street Journal*, the *Financial Times*, as well as the Dow Jones, Reuters, and Associated Press news services. There are other news-source databases to which many academic institutions subscribe such as LexisNexis,[6] Business Source Complete,[7] the Canadian Business & Current Affairs Database,[8] among others. However, these databases are aimed at specific topic searches, and not large scale downloads. Many database providers may seize access to the data when it detects such unusual activity, so it is advisable to get the vendor's permission before downloading large quantities of data.

Collecting your own data is also a viable option through the use of public Internet resources. Many Web sites, blogs, and message boards have archives that may be downloaded manually or using a special type of software known as Web crawlers (also called Web robots, Web spiders, . . .). Many open source Web crawlers[9] are available with an array of document collection properties, but the essence of each is that it browses the Internet in a structured way while creating copies of the visited Web pages and storing them for future processing. The starting URL(s) is specified and the crawler travels through other Web pages through the hyperlinks contained in previously visited pages according to a specified set of rules. There are many variants of Web crawlers that researchers may find useful including focused and topical crawlers that attempt to only visit Web pages on a given topic.

There are many other sources of textual data that are publicly available including academic literature, firm press releases and shareholder reports, analyst reports, political speeches, and government/institutional reports.

[3] trec.nist.gov/data/reuters/reuters.html

[4] www.ldc.upenn.edu

[5] www.factiva.com

[6] www.lexisnexis.com

[7] www.ebscohost.com/titleLists/bt-complete.htm

[8] www.proquest.com/en-US/catalogs/databases/detail/cbca.shtml

[9] One such Web crawler is the DataparkSearch Engine, available at www.dataparksearch.org, which will search and extract documents from within a Web site or group of Web sites.

5.5 Document Preprocessing

The technology that permits the quantification of textual data has been largely developed within the field of natural language processing (NLP) (Charniak 1996; Jurafsky and Martin 2000; Manning and Schutze 1999). Many tasks that seem trivial to a language using human are in fact frustratingly difficult to program a computer to do. The level of NLP pre-processing can vary widely depending on the particular research domain and goals. Various levels of preprocessing are described below:

- *Document format standardization:* The current standard for document formatting is the XML (Extensible Markup Language)[10] format. An advantage of XML is that it permits the placement of delimiting tags around various parts of a document such as <DOC> $\cdots$ </DOC> which indicates where a document begins and ends. Other tags indicate document components, such as titles and section headings, that may be of special importance. In cases where document components are easily identified, plain text files are easiest to work with.
- *Tokenization:* This process is breaking stream of characters into groups called tokens. The order of the original sequence is maintained, and only white-space is removed. For example, the stream "You are reading this sentence." would be transformed into the following sequence of tokens: [You][are][reading][this][sentence[.] Some multi-word expressions, such as "data set," may be treated as single tokens. These word groups include collocations, fixed expressions, and idioms.
- *Misspelling correction:* Commonly misspelt words may be automatically replaced to improve task performance. However, unsupervised replacement of all unfamiliar words (i.e., those not in a specified dictionary) can reduce performance (Malouf and Mullen 2008).
- *Sentence boundary detection:* If the intended linguistic features involve more than token occurrences, then the next step is to identify where sentences begin and end. There are some hand-crafted algorithms for sentences boundary detection that exploit language regularities which can achieve greater than 90% accuracy. Given sufficient training data, classification learning algorithms can achieve more than 98% accuracy (Weiss et al. 2005).
- *Part-of-speech (PoS) tagging:* Each token has a linguistic function (called its part-of-speech) that can be identified (or "tagged"). This will be useful if you are only interested in certain types of tokens, such as adjectives or nouns, or if you wish to do further NLP processing. The number of classes of PoS objects varies considerably depending on the level of detail. As an example, the CLAWS PoS Tagger[11] takes the sentence, "You are reading this sentence." and outputs "`[You_PNP]`

[10] www.w3.org/XML

[11] ucrel.lancs.ac.uk/claws

`[are_VBB] [reading_VVG] [this_DT0] [sentence_NN1] [._PUN]`" where PNP indicates a personal pronoun, VBB indicates the "base forms" of the verb "BE," VVG identifies the -ing form of a lexical verb, DT0 indicates a general determiner, NN1 indicates a singular noun, and PUN indicates punctuation. Another well known and publicly available PoS tagger is the Brill tagger.[12]

- *Phrase identification:* Also known as "text-chunking," identifies important word sequences. Primarily these sequences are noun phrases (e.g., "the economic crisis"), verb phrases (e.g., "has worsened"), or prepositional phrases (e.g., "because of"). Phrases can be used as features, and also they can be used in parsing and named entity recognition.
- *Named entities:* Identifying proper noun phrases (named entities) such as people, organizations, and locations, is important in many textual processing applications. This is a subtask of phrase recognition, but as 90% of new lexemes encountered NLP systems are proper nouns (do Prado and Ferneda 2008), there is considerable focus on identifying and classifying them.
- *Parsing:* Parsing is the process of linking words within a sentence by grammatical relationships. For example, the Stanford Parser[13] identifies the following relationships in the sentence, "You are reading this sentence."
  1. nsubj(reading-3, You-1) – "You" is a nominal subject of a clause of which "reading" is the governor.
  2. aux(reading-3, are-2) – "are" is an auxiliary of a clause whose main verb is "reading".
  3. det(sentence-5, this-4) – "this" is the determiner of a noun phrase whose head is "sentence".
  4. dobj(reading-3, sentence-5) – "sentence" is the direct object of a verb phrase whose main verb is "reading".

  The output of a parser is often in the form of a tree representing the dependency between the parts of the sentence. Parsing permits us to automatically associate descriptions and actions to named entities. Without parsing, textual data such as "So XYZ's earnings were not bad this year." and "Not so! XYZ's earnings were bad this year." would likely be treated the same despite their very different meaning.
- *Normalization/lemmatization/stemming:* Performance may be improved by aggregating different types of tokens. For example we may want to treat the words "book" and "books" as instances of the same thing, and so we may transform both into a single standard form. There are many types and degrees of such standardization. Inflectional

[12] www.tech.plym.ac.uk/soc/staff/guidbugm/software/RULE_BASED_TAGGER_V.1.14.tar.Z

[13] http://nlp.stanford.edu/software/lex-parser.shtml

stemming involves aggregating across grammatical variation from factors such as tense and plurality. Lemmatization replaces words with their primitive form. "Stemming to a Root" removes all prefixes and suffixes from a word. This last form of stemming is very strong and since it often removes important information, it should not often be used. While such normalization can be done immediately after tokenization, if further NLP processing such as parsing is intended, it should be postponed until later.

- *Synonyms and coreferences:* Collapsing synonyms and coreferences involves aggregating all synonyms to their common sense (meaning). An important tool in dealing with synonyms is WordNet,[14] which organizes English nouns, verbs, adjectives, and adverbs into synonym sets.

5.6 Quantifying Textual Information

With documents in a standardized format, the most critical question of quantitative textual analysis must be considered: How do we transform textual data into numerical data? The general strategy is to define a feature vector for each document, where each vector element is associated with a linguistic feature (such as a word, type of word, phrase, relationship, etc.) and the numerical entry in this vector element measures the extant to which the feature is present in the document. The choice of the type of features is critical and may depend on the intended task, the properties of the textual data, or the level of linguistic sophistication required. The number of features generated from textual data can easily reach into the tens and even hundreds of thousands. Consequently, combinations of dimensionality reduction and appropriate classification and prediction methods are also needed. The following list indicates some of the possible choices of types of features:

- *Word occurrence:* For this type of feature, the feature vector will have a length that is the size of the dictionary. That is, for every distinct word in the training data (or corpus), there is a corresponding feature element. The feature vector for a given document will have a 1 in the nth element if the document contains the nth word in the dictionary, otherwise it contains a 0.
- *Word frequencies:* Rather than just having a 1 or 0 in each entry, this feature type uses some measure of the frequency of words in the document to weight the entry. There are several potential frequency measures that can be used. The simplest method is just the number of times each word occurs in the document; however, there are

[14] `wordnet.princeton.edu`

more effective measures. There are two primitives that are used in the construction of term weighting measures: term frequency ($tf_{i,j}$: the number of occurrences of word w_i in document d_j) and document frequency (df_i: the number of documents in the collection in which w_i occurs). A common weighting scheme based on these primitives is called the inverse document frequency (*idf*)

$$idf_{i,j} = \begin{cases} (1 + \log(tf_{i,j})) \log \frac{N}{df_i} & \text{if} \quad tf_{i,j} \geq 1 \\ 0 & \text{if} \quad tf_{i,j} = 0 \end{cases} \tag{5.6}$$

where N is the number of documents. The idea of this type of weighting scheme is that words that occur frequently are informative, but words that occur in many documents are less informative. There are several other weighting schemes based on this idea that use different functions for term frequency, document frequency, and normalization. Both word-occurrence and word-frequency type features are often referred to as the bag-of-words representation since all structural properties of the textual data are lost.

- *n-Grams/multi-word features:* An n-gram is a sequence of n words. Thus, the word-occurrence and word-frequency type features are examples of 1-gram, or unigram, features. We could extend these types of features to include pairs of words that occur together to generate 2-gram, or bigram, features. Tan, Wang, and Lee (2002) show that bigrams plus unigrams improve performance in a Web page classification task compared to unigrams alone.
- *Word+PoS features:* Occurrence or frequency features could also be generated for words paired with their parts of speech.
- *Parsed features:* Word n-grams together with their parsed dependencies would provide considerable linguistic sophistication to the feature vectors. Such vectors would very large, with lengths on the order of hundreds of thousands.

5.7 Dimensionality Reduction

With such large feature vectors, dimensionality reduction (also called feature selection) can improve classification performance as well as increase computational speed. An introduction to feature selection in high-dimensional settings is given by Guyon and Elisseeff (2003), and experimental results for various feature selection strategies is given by Yang and Pedersen (1997) and Formen (2003).

The idea behind dimensionality reduction is that not all features are equally informative, so we would like to score each potential feature by some

"informativeness" metric, and then select only the best k features. Yang and Pedersen (1997) have shown that in some cases removal of up to 98% of the features can improve classification performance. Before reviewing some scoring procedures, it is useful to review some of the informal filters that are often applied.

In bag-of-words feature sets, words can be aggregated into broad categories to capture more general features rather than specific meanings. For example, Tetlock (2007) classifies all words in his documents as belonging to one of 77 categories using the Harvard-IV-4 dictionary. He then further reduces the dimensionality down to one feature using principal component analysis (PCA) to extract the single factor that captures the maximum variation in the dictionary categories. In Tetlock, Saar-Tsechansky, and Macskassy (2008), only one of the 77 categories is used to measure the negative sentiment of a document, thereby again reducing the dimensionality to one.

A common feature filter is to eliminate rare words on the grounds that they will not help with classification. In many cases, up to half of the distinct words in a collection will occur only once. Removing these features will greatly improve the processing speed as well as reduce potential for over-fitting.

Similarly, removing the most common words will not likely harm classification performance since these function words (such as "the", "a", and "of") are purely grammatical rather than informative. A frequency threshold can be used to identify these words, or a "stopword" list may be provided. Also, as mentioned earlier, collapsing synonyms and coreferences, as well as lemmatization and stemming can aggregate words into broader classes thereby reducing dimensionality and potentially improving classification and prediction performance.

There are many formal feature selection metrics that can be used to rank potential features. According to Formen (2003), the Bi-Normal Separation metric outperforms other more common metrics. Other metrics include Information Gain, Mutual Information, Chi-Square, Odds Ratio, and others. Formal descriptions of these metrics can be found in Formen (2003) and Yang and Pedersen (1997).

5.8 Classification Problems

The growth of unsolicited and unwanted e-mails that are sent out in a bulk or automatic fashion, also known as spam, has posed a great threat to Internet communications. Today, estimates put the volume of spam in the range of 88%–92% of all e-mails (MAAWG 2010). In order to maintain the useability of e-mail communications, spam detection systems were developed to classify e-mails as spam or not. This is one of the most successful examples of automated textual classification, but similar technology can be applied to a wide array of classification and prediction problems.

5.8.1 Classification

Textual classification (also known as categorization or supervised learning) is the task of assigning documents $\mathcal{D} = \{d_1, \ldots, d_{|\mathcal{D}|}\}$ to predefined[15] classes $\mathcal{C} = \{c_1, \ldots, c_{|\mathcal{C}|}\}$. Thus, a classifier is a function

$$\mathcal{F} : \mathcal{D} \times \mathcal{C} \to \{T, F\} \tag{5.7}$$

where $\mathcal{F}(d_i, c_j) = T$ if and only if document d_i belongs to class c_j. Under this general specification, a document may simultaneously belong to multiple classes, as may be appropriate when, for example, an e-mail is not spam, about a particular project, and from a particular sender. Such a problem is usually broken down into $|\mathcal{C}|$ simpler binary classification problems. That is, for each class $c_i \in \mathcal{C}$, we define a binary classifier

$$\mathcal{F}_i : \mathcal{D} \to \{T, F\} \tag{5.8}$$

where $\mathcal{F}_i(d_j) = T$ if and only if document d_j belongs to class c_i. When exactly one class can be assigned to each document, as in the case of BUY, SELL, and HOLD recommendations, the classifier will be of the form

$$\mathcal{F} : \mathcal{D} \to \mathcal{C} \tag{5.9}$$

As before, depending on the classification algorithm chosen, this type of problem may be reduced to a set of binary classifiers with rules for dealing with multiple class assignments.

If $\mathcal{F}$ is the correct or authoritative classifier, then we wish to approximate this function with $\widehat{\mathcal{F}}$. Approximating classifiers has been extensively studied in the field of machine learning (Mitchell 1997). The specific classification scheme of $\widehat{\mathcal{F}}$ is determined by a set of training documents for which the correct classifications are known. These training documents can be classified by a domain expert according to their linguistic properties, or they can be classified according to some specific data that is aligned with the text document. Fung, Yu, and Lu (2005) uses this latter approach and aligns news stories in a time series with stock market performance.

5.8.2 Classifier Evaluation

A classifier can be evaluated across many dimensions. Its ability to correctly classify documents is of paramount importance; however, its speed and scalability in both the training and classification phases are also important. Additionally, a good classifier should be relatively easy to use and understand. The focus in this section, however, is limited to the evaluation of a classifier's primary task.

Recall that the classifier $\mathcal{F}$ maps $\mathcal{D} \times \mathcal{C}$ into the $\{T, F\}$ such that $\mathcal{F}(d_i, c_j) = T$ if and only if document d_i belongs to class c_j. In this case, we call d_i a positive

[15] When the classes are not known in advance, then the task is referred to as clustering.

example of class c_j. When $\mathcal{F}(d_i, c_j) = F$ then document d_i is not a member of the class c_j and so we call d_i a negative example of c_j. To capture the correctness of classifications from a trained classifier $\widehat{\mathcal{F}}$, we introduce the following four basic evaluation functions:

$$TP_{\widehat{\mathcal{F}}}(d_i, c_j) = \begin{cases} 1 & \text{if } \widehat{\mathcal{F}}(d_i, c_j) = T \text{ and } \mathcal{F}(d_i, c_j) = T \\ 0 & \text{otherwise} \end{cases} \tag{5.10}$$

$$TN_{\widehat{\mathcal{F}}}(d_i, c_j) = \begin{cases} 1 & \text{if } \widehat{\mathcal{F}}(d_i, c_j) = F \text{ and } \mathcal{F}(d_i, c_j) = F \\ 0 & \text{otherwise} \end{cases} \tag{5.11}$$

$$FP_{\widehat{\mathcal{F}}}(d_i, c_j) = \begin{cases} 1 & \text{if } \widehat{\mathcal{F}}(d_i, c_j) = T \text{ and } \mathcal{F}(d_i, c_j) = F \\ 0 & \text{otherwise} \end{cases} \tag{5.12}$$

$$FN_{\widehat{\mathcal{F}}}(d_i, c_j) = \begin{cases} 1 & \text{if } \widehat{\mathcal{F}}(d_i, c_j) = F \text{ and } \mathcal{F}(d_i, c_j) = T \\ 0 & \text{otherwise} \end{cases} \tag{5.13}$$

Thus, every classification by $\widehat{\mathcal{F}}$ will either be a true positive (*TP*), a true negative (*TN*), a false positive (*FP*), or a false negative (*FN*). The *TP*s and *TN*s indicate correct classifications, and the *FP*s and *FN*s indicate incorrect classifications.

A simple, and sometimes overused, performance measure of classifiers is accuracy (*A*) measured as the proportion of correct classifications.

$$A = \frac{\sum_{i=1}^{|\mathcal{D}|} \sum_{j=1}^{|\mathcal{C}|} \left(TP_{\widehat{\mathcal{F}}}(d_i, c_j) + TN_{\widehat{\mathcal{F}}}(d_i, c_j) \right)}{\sum_{i=1}^{|\mathcal{D}|} \sum_{j=1}^{|\mathcal{C}|} \left(TP_{\widehat{\mathcal{F}}}(d_i, c_j) + TN_{\widehat{\mathcal{F}}}(d_i, c_j) + FP_{\widehat{\mathcal{F}}}(d_i, c_j) + FN_{\widehat{\mathcal{F}}}(d_i, c_j) \right)} \tag{5.14}$$

The converse of accuracy is the error rate (E) measured as

$$E = 1 - A \tag{5.15}$$

Accuracy, and error, are often useful performance measures, but they do not always capture the intended notion of correctness. For example, when trying to classify rare events, positive and negative examples will be strongly imbalanced with far more negative than positive examples. In this case, a universal rejector (i.e., $\widehat{\mathcal{F}}(d_i, c_j) = F, \forall d_i, c_j$) will have a high accuracy while being of no practical use.

To address such concerns, two other performance measures have become popular: precision and recall. Precision, with respect to a class c_j, is the proportion of documents assigned to class c_j that actually belong to that class.

$$P_{\widehat{\mathcal{F}}}(c_j) = \frac{\sum_{i=1}^{|\mathcal{D}|} TP_{\widehat{\mathcal{F}}}(d_i, c_j)}{\sum_{i=1}^{|\mathcal{D}|} \left(TP_{\widehat{\mathcal{F}}}(d_i, c_j) + FP_{\widehat{\mathcal{F}}}(d_i, c_j) \right)} \tag{5.16}$$

So, given a particular class, precision is the ratio of correct positive classifications to the total number of positive classifications. Precision can be aggregated across classes in two ways. First, microaveraged precision averages the precision of $\widehat{\mathcal{F}}$ for each class, weighted by the number of positive documents.

$$P_{\widehat{\mathcal{F}}}^{\text{Micro}} = \frac{\sum_{j=1}^{|\mathcal{C}|} \sum_{i=1}^{|\mathcal{D}|} TP_{\widehat{\mathcal{F}}}(d_i, c_j)}{\sum_{j=1}^{|\mathcal{C}|} \sum_{i=1}^{|\mathcal{D}|} \left(TP_{\widehat{\mathcal{F}}}(d_i, c_j) + FP_{\widehat{\mathcal{F}}}(d_i, c_j)\right)} \tag{5.17}$$

Alternatively, macroaveraged precision averages the precision for each class with equal weights.

$$P_{\widehat{\mathcal{F}}}^{\text{Macro}} = \frac{\sum_{j=1}^{|\mathcal{C}|} P_{\widehat{\mathcal{F}}}(c_j)}{|\mathcal{C}|} \tag{5.18}$$

The second performance measure is recall. Recall, for a given class c_j, is the proportion of documents that truly belong to c_j that are classified as belonging to that class by $\widehat{\mathcal{F}}$.

$$R_{\widehat{\mathcal{F}}}(c_j) = \frac{\sum_{i=1}^{|\mathcal{D}|} TP_{\widehat{\mathcal{F}}}(d_i, c_j)}{\sum_{i=1}^{|\mathcal{D}|} \left(TP_{\widehat{\mathcal{F}}}(d_i, c_j) + FN_{\widehat{\mathcal{F}}}(d_i, c_j)\right)} \tag{5.19}$$

So, given a particular class, precision is the ratio of the number of correct positive classifications by the total number of truly positive class documents. As with precisions, we can use microaveraging to define a measure of recall across all classes.

$$R_{\widehat{\mathcal{F}}}^{\text{Micro}} = \frac{\sum_{j=1}^{|\mathcal{C}|} \sum_{i=1}^{|\mathcal{D}|} TP_{\widehat{\mathcal{F}}}(d_i, c_j)}{\sum_{j=1}^{|\mathcal{C}|} \sum_{i=1}^{|\mathcal{D}|} \left(TP_{\widehat{\mathcal{F}}}(d_i, c_j) + FN_{\widehat{\mathcal{F}}}(d_i, c_j)\right)} \tag{5.20}$$

Alternatively, we may define a macroaverage measure of recall.

$$R_{\widehat{\mathcal{F}}}^{\text{Macro}} = \frac{\sum_{j=1}^{|\mathcal{C}|} R_{\widehat{\mathcal{F}}}(c_j)}{|\mathcal{C}|} \tag{5.21}$$

Most classifiers can be set up to tradeoff precision for recall, or vice versa. Consequently, it is useful to present a combined measure of performance, the F_1 score, which is the harmonic mean of precision and recall:

$$F_1 = \frac{2 \times P \times R}{P + R} \tag{5.22}$$

where P and R are either micro- or macroaveraged.

5.8.3 Classification Algorithms

There is a multitude of classification algorithms to choose from including decision trees, Bayesian classifiers, Bayesian belief networks, rule-based classifiers, backpropagation, genetic algorithms, k-nearest neighbor classifiers, and others. Detailed treatments of these methods can be found in many textbooks including the one by Han and Kamber (2006). For text classification,

however, support vector machines (SVMs) deserve special mention since they consistently rank as or among the best classification methods (Joachims 1998) and can handle very-high-dimensional data.

A complete description of SVMs is given by Burges (1998) and Vapnik (1998), and there are many public resources[16] available to those who wish to learn about and use these classifiers. Thorsten Joachims has created a popular implementation of the SVM algorithm called SVMlight,[17] which is publicly available. The basic SMV is a binary classifier that finds a hyperplane with the maximum margin between positive and negative training documents. There are now SMVs for multi-class classification as well as for regression. SMVs have three unique features:

1. Not all training documents are used to train the SVM. Instead, only documents near the classification boarder are used to train the SMV.
2. Not all features from the training documents are used, so excessive feature reduction is not needed.
3. SMVs can construct irregular boarders between positive and negative training documents.

Another classification technique that is worthy of mention is the use of ensemble methods that combine several different classification methods. Two examples of ensemble methods are bagging (i.e., bootstrap aggregation) and boosting (a series of classifications that weight previously misclassified training examples more heavily). Han and Kamber (2006) describe implementations of these techniques, and Das and Chen (2007) employ an ensemble voting method.

As a final note, regression techniques are also available for textual data represented as a feature vector. Traditional regression methods can be used when the number of features has been greatly reduced as in Tetlock (2007). For larger feature vectors, support vector regressions (SVRs) can be used.

5.9 Software

There are many software packages available to facilitate textual econometric research. In addition to those listed throughout this chapter, some popular statistical packages have textual analysis modules:

- *SAS text miner*[18]*:* This package provides tools for transforming textual data into a usable a format, as well as for classifying documents, finding relationships and associations between documents, and clustering documents into categories.

[16] www.support-vector-machines.org

[17] svmlight.joachims.org

[18] www.sas.com/technologies/analytics/datamining/textminer

- *SPSS text mining for clementine*[19]*:* This package uses NLP techniques to extract key concepts, sentiments, and relationships from textual data. Feature vectors can be created and used in SPSS for predictive modeling.

There are many other packages available. One package that is very comprehensive, freely available, and well documented is the Natural Language Toolkit[20] for the Python programming language. It contains open source Python modules, linguistic data, and documentation for many of the tasks described in this chapter. In addition to the documentation available, a book has been written by Bird, Klein, and Loper (2009) as a guide to the toolkit. Other programming resources are described by Bilisoly (2008), Konchandy (2006), and Chakrabarti (2003).

5.10 Conclusion

The aim of this chapter has been to introduce econometricians to tools and techniques that allow textual data to be analyzed in a quantitative and statistical manner. This new area of textual econometrics is in its early stages of development and draws heavily from the fields of natural language processing and text mining (Feldman and Sanger 2007; Weiss et al. 2005). Early work in the field has proven that useful information is embedded in textual data that can be extracted using these techniques. As these tools improve and the areas of application expand, textual econometrics is a field bound to expand.

5.11 Acknowledgment

The authors wish to thank Anoop Sarkar of SFU's Natural Language Laboratory for introducing them to this interesting field.

References

Antweiler, W., and M. Frank. 2004. Is all that talk just noise? the information content of internet stock message boards. *Journal of Finance* LIX(3): 1259–1294.

Armstrong, J. 2001. Combining forecasts. *Principles of Forecasting*. Norwell, MA: Kluwer, pp. 417–439.

[19] www.spss.com/text_mining_for_clementine

[20] www.nltk.org.

Bates, J., and C. Granger. 1969. The combination of forecasts. *Operations Research Quarterly* 20: 451–468.

Bhattacharya, U., N. Galpin, R. Ray, and X. Yu. 2009. The role of the media in the internet IPO bubble. *Journal of Financial and Quantitative Analysis* 44(3): 657–682.

Bilisoly, R. 2008. *Practical Text Mining with Perl*. Hoboken, NJ: Wiley.

Bird, S., E. Klein, and E. Loper. 2009. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. Sebastopol, CA: O'Reilly Media.

Burges, C. 1998. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* 2: 121–168.

Chakrabarti, S. 2003. *Mining the Web: Discovering Knowledge from Hypertext Data*. San Francisco: Morgan Kaufmann Publishers.

Chan, W. 2003. Stock price reaction to news and no-news: Drift and reversal after headlines. *Journal of Financial Economics* 70(2): 223–260.

Charniak, E. 1996. *Statistical Language Learning*. Cambridge, MA: MIT Press.

Coval, J., and T. Shumway. 2001. Is sound just noise? *Journal of Finance* LVI(5): 1887–1910.

Das, S., and M. Chen. 2007. Yahoo! for Amazon: Sentiment extraction from small talk on the web. *Management Science* 53(9): 1375–1388.

Dennett, D. 1994. The role of language in intelligence. In J. Khalfa (ed.), *What is Intelligence? The Darwin College Lectures*. Cambridge, U.K.: Cambridge University Press.

do Prado, H., and E. Ferneda. 2008. *Emerging Technologies of Text Mining: Techniques and Applications*. New York: Information Science Reference.

Fama, E., L. Fisher, M. Jensen, and R. Roll. 1969. The adjustment of stock prices to new information. *International Economic Review* 10.

Feldman, R., and J. Sanger. 2007. *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. New York: Cambridge University Press.

Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

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

Frisch, R. 1933. Editor's note. *Econometrica* 1(1): 1–4.

Fung, G., J. Yu, and H. Lu. 2005. The predicting power of textual information on financial markets. *IEEE Intelligent Informatics Bulletin* 5(1): 1–10.

Gao, L., and P. Beling. 2003. Machine quantification of text-based economic reports for use in predictive modeling. *IEEE International Conference on Systems, Man and Cybernetics* 4: 3536–3541.

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

Han, J., and M. Kamber. 2006. *Text Mining: Concepts and Techniques*. 2nd ed. San Francisco: Morgan Kaufmann.

Joachims, T. 1998. Text categorization with support vector machines: Learning with many relevant features. *Proceedings of the Tenth European Conference on Machine Learning*. Heidelberg: Springer-Verlag, pp. 137–142.

Jurafsky, D., and J. Martin. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Upper Saddle River, NJ: Prentice Hall.

Katz, S. 1996. Distribution of content words and phrases in text and language modelling. *Natural Language Engineering* 2: 15–59.

Konchandy, M. 2006. *Text Mining Application Programming*. Boston: Charles River Media.

Kroha, P., R. Baeza-Yates, and B. Krellner. 2006. Text mining of business news for forecasting. *Proceedings of the 17th International Conference on Database and Expert Systems Applications*. Berlin: Springer, pp. 171–175.

Lewis, D., Y. Yang, T. Rose, and F. Li. 2004. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research* 5: 361–397.

MAAWG. 2010. Report 12 of the Messaging Anti-Abuse Working Group: Third and fourth quarter 2009. *Email Metrics Program: The Network Operators Perspective*. August 20, 2010.

Malouf, R., and T. Mullen. 2008. Taking sides: User classification for informal online political discourse. *Internet Research* 18: 177–190.

Mandelbrot, B. 1954. Structure formelle des textes et communication. *Word*, 10: 1–27.

Manning, C., and H. Schutze. 1999. *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press.

Mitchell, T. 1997. *Machine Learning*. New York: McGraw-Hill.

Mitchell, M., and J. Mulherin. 1994. The impact of public information on the stock market. *Journal of Finance* 49(3): 923–950.

Mittermayer, M. 2004. Forecasting intraday stock price trends with text mining techniques. *Proceedings of the 37th Hawaii International Conference on System Sciences*, pp. 1–10. IEEE Computer Society Press.

Niederhoffer, V. 1971. The analysis of world events and stock prices. *Journal of Business* 44(2), 193–219.

Rachlin, G., M. Last, D. Alberg, and A. Kandel. 2007. ADMIRAL: A data mining based financial trading system. *Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Data Mining*, pp. 720–725. IEEE Computer Society Press.

Schumaker, R., and H. Chen. 2009. Textual analysis of stock market prediction using breaking financial news: The azfintext system. *Association for Computing Machinery Transactions on Information Systems* 27(2): 1–19.

Tan, C., Y. Wang, and C. Lee. 2002. Using bi-grams to enhance text categorization. *Information Processing and Management* 38(4): 529–546.

Tetlock, P. 2007. Giving content to investor sentiment: The role of media in the stock market. *Journal of Finance* LXII(3): 1139–1168.

Tetlock, P., M. Saar-Tsechansky, and S. Macskassy. 2008. More than words: Quantifying language to measure firms' fundamentals. *Journal of Finance* LXIII(3): 1437–1467.

Ticom, A., and B. de Lima. 2007. Text mining and expert systems applied in labor laws. *Seventh International Conference on Intelligent Systems Design and Applications*, pp. 788–792. IEEE Computer Society Press.

Turing, A. 1950. Computing machinery and intelligence. *Mind* LIX: 433–460.

Vapnik, V. 1998. *Statistical Learning Theory*. New York: John Wiley & Sons.

Weiss, S., N. Indurkhya, T. Zhang, and F. Damerau. 2005. *Text Mining: Predictive Methods for Analyzing Unstructured Information*. New York: Springer.

Wysocki, P. 1999. Cheap talk on the Web: The determinants of posting on stock message boards. *University of Michigan Business School Working Paper* (No. 98025), Ann Arbor.

Yang, Y., and J. Pedersen. 1997. A comparative study of feature selection in text categorization. *Proceedings of the Fourteenth International Conference on Machine Learning* 3: 412–420.

Zipf, G. 1932. *Selective Studies and the Principle of Relative Frequency in Language*. Boston: Cambridge University Press.