# Taming Text: An Introduction to Text Mining

Louise A. Francis, FCAS, MAAA

**Abstract**

**Motivation.** One of the newest areas of data mining is text mining. Text mining is used to extract information from free form text data such as that in claim description fields. This paper introduces the methods used to do text mining and applies the method to a simple example.

**Method**. The paper will describe the methods used to parse data into vectors of terms for analysis. It will then show how information extracted from the vectorized data can be used to create new features for use in analysis. Focus will be placed on the method of clustering for finding patterns in unstructured text information.

**Results**. The paper shows how feature variables can be created from unstructured text information and used for prediction

**Conclusions**. Text mining has significant potential to expand the amount of information that is available to insurance analysts for exploring and modeling data

**Availability**. Free software that can be used to perform some of the analyses describes in this paper is described in the appendix.

**Keywords**. Predictive modeling, data mining, text mining, statistical analysis

## 1. INTRODUCTION

Traditional statistical analysis is performed on data arrayed in spreadsheet format. That is, the data is arrayed in two dimensional matrices where each row represents a record and each column represents a feature or variable. Table 1-1 provides a sample of such a database. In Table 1-1, each row represents a claimant. The features are the variables claim number, accident date, claim status, attorney involvement, paid loss, outstanding loss, incurred loss, incurred allocated loss adjustment expenses (ALAE) and claimant state. As seen in Table 1-1, the data contain two key types of variables, quantitative or numeric variables such as incurred losses and incurred expenses and nominal or categorical variables such as claim status and state. Each numeric value denotes a specific quantity or value for that variable. Each value or category, whether numeric or alphanumeric, of a categorical variable embeds a coding that maps the value to one and only one category.[1] This data is structured data. Structured databases result from intentional design where the variables have proscribed definitions and the values of the variables have proscribed meaning.

[1] Note that categorical variables can contain numeric codes as well as string values as in the example. Each code for the categorical variables maps to a value. That is injury '01' may denote a back strain and injury '02' may denote a broken wrist for an injury type variable.

**Sample Structured Data**

| Claim No | Accident Date | Status | Attorney | Paid | Outstanding | Incurred ALAE | Incurred Loss | State |
|---|---|---|---|---|---|---|---|---|
| 199816 | 01/08/1999 | C | Yes | 37,284 | 0 | 11,021 | 37,284 | NY |
| 199843 | 01/16/1999 | C | No | 0 | 0 | 0 | 0 | NY |
| 200229 | 12/30/2002 | O | No | 195 | 0 | 3 | 195 | CA |
| 199868 | 09/19/1998 | C | Yes | 99,852 | 0 | 31,807 | 99,852 | NJ |
| 200327 | 05/19/2003 | C | No | 286 | 0 | 72 | 286 | PA |

**Table 1-1**

Another kind of data that is also present in corporate databases is unstructured data. This data typically has the appearance of free form text data. Examples of text data are claim description fields in claim files, the content of e-mails, underwriters written evaluation of prospective policyholders contained in underwriting files and responses to open ended survey questions on customer satisfaction survey. It has been estimated that 85% of corporate data is of the unstructured type (Robb, 2004). As Mani Shabrang of Dow Chemical says, "We are drowning in information but starved for knowledge" (Robb, 2004).

When data is unstructured there is no obvious procedure for converting the data which is composed of sequences of characters that vary in length and content in apparently random ways, to information that can be used for analysis and prediction. Manual intervention on the part of human beings may be able to convert some unstructured data to structured features which can be used to perform statistical analysis. Derrig *et al.* (1994) provide an example where claims experts reviewed claims files and scored the claims on a number of indicators of suspicion of fraud. Because of the effort required and difficulty of interpreting the unstructured text data, it is typically ignored for doing analysis. If information could be automatically extracted from unstructured data, a significant new source of data could become available to corporations.

In the field of data mining, *text mining* has been attracting increased interest in the insurance industry. Text mining refers to a collection of methods used to find patterns and create intelligence from unstructured text data.

In this paper, the key methods used in text mining will be presented. A simple application to a free form claim description field will be used to illustrate the text mining procedures.

## 1.1 Research Context

While text mining is relatively new, software for analyzing text data has been available since the late 1990s from the major statistical software vendors such as SAS and SPSS. One of the most common uses of text mining procedures is in search engine technology. A user types in a word or phrase, which may include misspellings, and the search engine searches through a vast repository of documents to find the most relevant documents. Other applications include:

- Analysis of survey data
    - Text mining is used as an automated approach to coding information from open ended survey questions.
- Spam identification
    - The title line and contents of e-mails are analyzed to identify which are spam and which are legitimate (Hastie *et al.*, 2001).
- Surveillance
    - It is believed that a project referred to as ENCODA monitors telephone, internet and other communications for evidence of terrorism (Wikipedia, 2005).
- Call center routing
    - Calls to help desks and technical support lines are routed based on verbal answers to questions.
- Public health early warning
    - Global Public Health Intelligence Network (GPHIN) monitors global newspaper articles and other media to provide an early warning of potential public health threats including disease epidemics such as SARS, and chemical or radioactive threats. (Blench, 2005).
- Alias identification
    - The aliases of health care and other providers are analyzed to detect over billing and fraud. For instance, a bill may have been submitted by John Smith, J. Smith and Smith, John. The same approaches may be used to

identify abuse by claimants, where given claimants submit numerous insurance claims under different aliases.

Text mining has evolved sufficiently that web sites are devoted to it and courses focusing solely on text mining are appearing in graduate school curricula. Text mining also occurs frequently as a topic at data mining conferences. While the interest in text mining is relatively recent, Weiss *et al.* (2005) point out that text analysis dates back to at least the late 1950s where "automatic abstracting" of text information was studied. In the 1970s and 1980s, artificial intelligence researchers were interested in natural language processing. Many of these early efforts did not yield commercially useful results, so interest in text analysis declined. However, in the 1990s new developments in text mining tools led to a reawakened interest in the field.

In property and casualty insurance, literature on text mining is sparse. In 2002 Ellingsworth described the application of text mining to fraud identification (Ellingsworth, 2002). Kolyshkina (2005) described the use of text mining to create features for identifying serious claims.

This paper attempts to fill a gap in the actuarial literature on text mining. It will show that text mining combines string manipulation functions that are available in many modern programming languages, with commonly available statistical analysis methods.

Many of the statistical procedures described in this paper are described in the statistical and actuarial literature (Hastie *et al.* 2001, Kaufman, 1990) but have not heretofore been applied to unstructured text data. Derrig *et al.* (1994) and Francis (2001, 2003), Hayword (2002) have described analytical methods that can be applied to large insurance databases and are used in data mining. Berry and Linoff, (1997), Kaufman and Rousseeuw (1990) and Hastie *et al.* (2001) described some of the dimension reduction techniques that are utilized in text mining. The description of the methods used to derive information from terms in text data will make use of these dimension reduction techniques.

Much of the text mining literature focuses on search engine and other information retrieval method. This paper focuses, instead, on using text mining for prediction of important business outcomes. It will therefore not cover some of the key approaches that are applied primarily in information retrieval.

## 1.2 Objective

The objective of this paper is to introduce actuaries and other insurance professionals to the methods and applications of text mining. The paper shows that many of the procedures are straightforward to understand and utilize. Many of the procedures have been known in the statistics discipline for decades. The two methods described are k-means and hierarchical clustering.

## 1.3 Outline

The remainder of the paper proceeds as follows. Section 2 will discuss the data used in the exposition of the text mining methods. Section 2.1 presents the first phase of text mining: parsing and other string manipulations used to create terms for further analysis. Section 2.2 presents the methods used to create features or variables from the terms extracted in the first phase of the process. These features can then be used to perform additional analysis. The concept of dimension reduction is discussed in section 2.2.1. The two key methods of dimension reduction used in this paper, k-means clustering and hierarchical clustering are discussed in sections 2.2.2 and 2.2.3 respectively. Further considerations such as the number of clusters to retain and cluster naming to provide understanding of the features created by clustering are described in sections 2.2.4 and 2.2.5. Section 2.2.6 presents two simple examples of using variables derived from text mining for prediction. Results of the analysis are summarized and discussed in Section 3. Conclusions are presented in Section 4.

While many details of how text mining is performed will be presented, some analysts will want to acquire software specific for text mining. A discussion of text mining software is presented in the Appendix.

# 2. BACKGROUND AND METHODS

Text mining can be viewed as having two distinct phases: term extraction and feature creation. Term extraction makes heavy use of string manipulation functions but also applies techniques from computational linguistics. Actual content is a result of the feature creation process. Feature creation applies unsupervised learning methods that reduce many potential features into a much smaller number of final variables. These features are then potentially useable as dependent or predictor variables in an analysis.

The example employed to illustrate text mining uses simulated data from a general liability claims file[2]. The data contains one free form text field: injury description. In this simple example, there is no injury or body part code in the data and the only information about the nature of the injury is the free form text description. The injury description is a very brief description, generally containing only a few words. The data is representative of that which might be available from a small self insured exposure. While many claims databases contain larger accident and claim description fields, this data serves as a simple example of how text mining works. An example of the sample text data is shown in Table 2-1.

**Sample Claim File Text Data**

| **INJURY DESCRIPTION** |
|---|
| BROKEN ANKLE AND SPRAINED WRIST |
| FOOT CONTUSION |
| UNKNOWN |
| MOUTH AND KNEE |
| HEAD, ARM LACERATIONS |
| FOOT PUNCTURE |
| LOWER BACK AND LEGS |
| BACK STRAIN |
| KNEE |

**Table 2-1**

The sample data also contains other insurance related information: incurred losses, incurred loss adjustment expenses, accident year, status (open/closed) and whether or not an attorney is involved in the claim. There are approximately 2,000 records in the data. The values in the data are simulated, but are based on relationships observed in actual data for this line of business.

## 2.1 Term Extraction

During term extraction, character text is first parsed into words. The term extraction process also strips away words that convey no meaning such as "a" or "the". An additional part of the process involves finding words that belong together such as "John Smith".

[2] The claim description field is very similar to actual claim descriptions in actual data from small self insurance programs. Other data, such as ultimate losses have been simulated, but are based on relationships actually observed in data.

When data is parsed, string functions are used to extract the words from the character string composing the text data. To do this, spaces, commas and other delimiters must be used to separate words. A simple example of parsing one record using Microsoft Excel string functions with blank spaces as delimiters is illustrated in Table 2.1-1. The total length of the character string is first determined using the "length" function. Then, the "find" function of Excel is used to find the first occurrence of a blank. This is shown in column (3). Next, the substring function is used to extract the first word from the text, using the position of the first blank (column (4)). The remainder of the term, after removing the first word is then extracted, again using the substring function (columns (5) and (6)). The process continues until every word has been extracted. The "iserr" function can be used to determine when no more blanks can be found in the field. The words extracted are shown in the highlighted area of the table.

**Example of Parsing Claim Description**

| Full Description | Total Length | Location of Next Blank | First Word | Remainder Length 1 |
|---|---|---|---|---|
| (1) | (2) | (3) | (4) | (5) |
| **BROKEN ANKLE AND SPRAINED WRIST** | **31** | 7 | BROKEN | 24 |
| **Remainder 1** | | **2nd Blank** | **2nd Word** | **Remainder Length 2** |
| (6) | | (7) | (8) | (9) |
| ANKLE AND SPRAINED WRIST | | 6 | ANKLE | 18 |
| **Remainder 2** | | **3rd Blank** | **3rd Word** | **Remainder Length 3** |
| (10) | | (11) | (12) | (13) |
| AND SPRAINED WRIST | | 4 | AND | 14 |
| **Remainder 3** | | **4th Blank** | **4th Word** | **Remainder Length 4** |
| (14) | | (15) | (16) | (17) |
| SPRAINED WRIST | | 9 | SPRAINED | 5 |
| **Remainder 4** | | **5th Blank** | **5th Word** | |
| (18) | | (19) | (20) | |
| WRIST | | 0 | WRIST | |

**Table 2.1-1**

The result of parsing is data organized in spreadsheet format, i.e., a rectangular matrix containing indicator variables for the words extracted from the text field. For each word found in any record in the data a variable is created. The variable carries a value of 1 if a given record contains the word and a 0 otherwise.

**Example of Terms Created**

| INJURY DESCRIPTION | BROKEN | ANKLE | AND | SPRAINED | WRIST | FOOT | CONTU-SION | UNKNOWN | NECK | BACK | STRAIN |
|---|---|---|---|---|---|---|---|---|---|---|---|
| BROKEN ANKLE AND SPRAINED WRIST | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| FOOT CONTUSION | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| UNKNOWN | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| NECK AND BACK STRAIN | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

**Table 2.1-2**

The example above displays data that could be created from an injury description text field. Each claim description is treated as a "bag of words" (Weiss *et al.*, 2005). The matrices resulting from parsing text data are typically sparse. That is, for most of the terms, most of the records contain a zero for that term and only a few records have a one.

The example shown is a relatively simple one. The claim description field is relatively short and contains no delimiters other than a blank space. However, other delimiters such as the comma and period occur frequently and need to be identified also. Some delimiters, such as a single apostrophe (as in I'll) and period (as in etc.) may be part of the words; so the complex rules for finding and using such delimiters must be coded into the program that parses the data.

Certain words occur very frequently in text data. Examples include "the" and "a". These words are referred to as "stopwords". The stopwords are words removed from the term collection because they have no meaningful content. By creating a list of such stopwords and eliminating them, the number of indicator variables created is reduced. Table 2.1-3 displays a sample of stopwords used in this analysis. Many of these stopwords do not appear in the claim description data, but appear frequently in text data.

| Stopwords |
|---|
| A |
| And |
| Able |
| About |
| Above |
| Across |
| Aforementioned |
| After |
| Again |

**Table 2.1-3**

Table 2.1-4 below presents a collection of words obtained from parsing the injury description data into single words and removing stop words.

| Parsed Words | |
|---|---|
| HEAD | INJURY |
| LACERATION | NONE |
| KNEE | BRUISED |
| UNKNOWN | TWISTED |
| L | LOWER |
| LEG | BROKEN |
| ARM | FRACTURE |
| R | FINGER |
| FOOT | INJURIES |
| HAND | LIP |
| ANKLE | RIGHT |
| HIP | KNEES |
| SHOULDER | FACE |
| LEFT | FX |
| CUT | SIDE |
| WRIST | PAIN |
| NECK | INJURED |

**Table 2.1-4**

Other issues affecting the usefulness of the data must be dealt with. One issue is multiple versions and spellings of words. Table 2.1-4 illustrates this. Both L and LEFT are used to denote left, R and RIGHT are used to denote right and the database has both the singular and plural versions of KNEE. In addition, it can be seen from the table that certain

"words" stand for the same injury. For example, as a result of abbreviations used, FX and FRACTURE as well as BROKEN all denote the same injury The process referred to as stemming is used to substitute one word, referred to as a stem (because in the example of knee and knees, both words have the same stem) for all versions of the term.

Once the words have been parsed, stopwords removed and stemming performed, the sparse matrix of term indicators is ready for the next step: feature creation. During the feature creation step, words and sequences of words are classified into groups that contain similar information.

In some text mining applications, especially those that attempt to understand the content contained in large documents, other analysis such as grammatical analysis is performed before progressing to the feature creation step. Such analysis will not be described here as it is not relevant to the example in this paper.

## 2.2 Feature Creation

Term extraction is the first step in deriving meaning or content from free form text. The next step is feature creation. Thus far, each line of text has been parsed into a "bag of words". The data can be represented as a rectangular array that has indicator variables for each term in the injury description. When we analyze the terms more carefully, we may find that some words such as "back strain" and "neck strain" denote similar injuries and are unlike "head trauma". Thus, occurrence or non-occurrence of specific words may tell us something useful about the nature and severity of the injury.

One of the most common techniques used to group records with similar values on the terms together is known as cluster analysis. Cluster analysis is an example of dimension reduction. Before describing cluster analysis, the concepts of dimension and of dimension reduction are introduced.

### 2.2.1 Dimension Reduction

Jacoby (1991) describes dimensions as "the number of separate and interesting sources of variation among objects"[3] There are two views as to sources of variation when dealing

[3] Jacoby, p. 27

with a database organized in rectangular spreadsheet format: columns (or variables) and rows (or records). Table 2.2.1-1 displays the two views of dimensionality for a sample claims database. The arrow pointing to the right indicates that each column of data can be viewed as a separate dimension. The downward pointing arrow indicates that each row or claimant also can be viewed as a dimension.

**Two Ways of Viewing Dimension in a Database**

| | CLAIM NUMBER | DATE OF LOSS | STATUS | INCURRED LOSS |
|---|---|---|---|---|
| | | | | VARIABLES |
| | 1998001 | 09/15/97 | C | 407.81 |
| | 1998002 | 09/25/97 | C | 0.00 |
| | 1998003 | 09/26/97 | C | 0.00 |
| | 1998004 | 09/29/97 | C | 8,247.16 |
| | 1998005 | 09/29/97 | C | 0.00 |
| | 1998006 | 10/02/97 | C | 0.00 |
| | 1998007 | 10/10/97 | C | 0.00 |
| | 1998008 | 10/24/97 | C | 0.00 |
| | 1998009 | 10/29/97 | C | 21,211.66 |
| | 1998010 | 10/29/97 | C | 0.00 |
| R | 1998011 | 11/03/97 | C | 0.00 |
| E | 1998012 | 11/03/97 | C | 0.00 |
| C | 1998013 | 11/04/97 | C | 451.66 |
| O | 1998014 | 11/04/97 | C | 0.00 |
| R | 1998015 | 11/04/97 | C | 0.00 |
| D | 1998016 | 11/06/97 | C | 15,903.66 |
| S | 1998017 | 11/11/97 | C | 465.10 |

Table 2.2.1-1

Each column contains information about the claimants and is a potential variable in an actuarial or modeling analysis. Each column is a separate dimension. Often in a large database containing hundreds or even thousands of variables, many variables are highly correlated with each other and contain redundant information. The large number of variables can be reduced to a smaller number of components or factors using a technique such as factor analysis. For instance, Figure 2.2-1 displays three of the dimensions related to financial information in the sample claims data; ultimate incurred loss, ultimate allocated loss adjustment expense (ALAE) and ultimate incurred loss plus ALAE. It can be seen from the graph that the three dimensions are correlated, which one would expect, particularly when

one of the variables is the sum of the other two. It is common in actuarial analysis (particularly with small databases) to work with only one of these variables; ultimate loss and ALAE. Thus the number of "dimensions" used in the analysis is reduced to one.

**Scatterplot of Correlated Dimensions (Variables)**

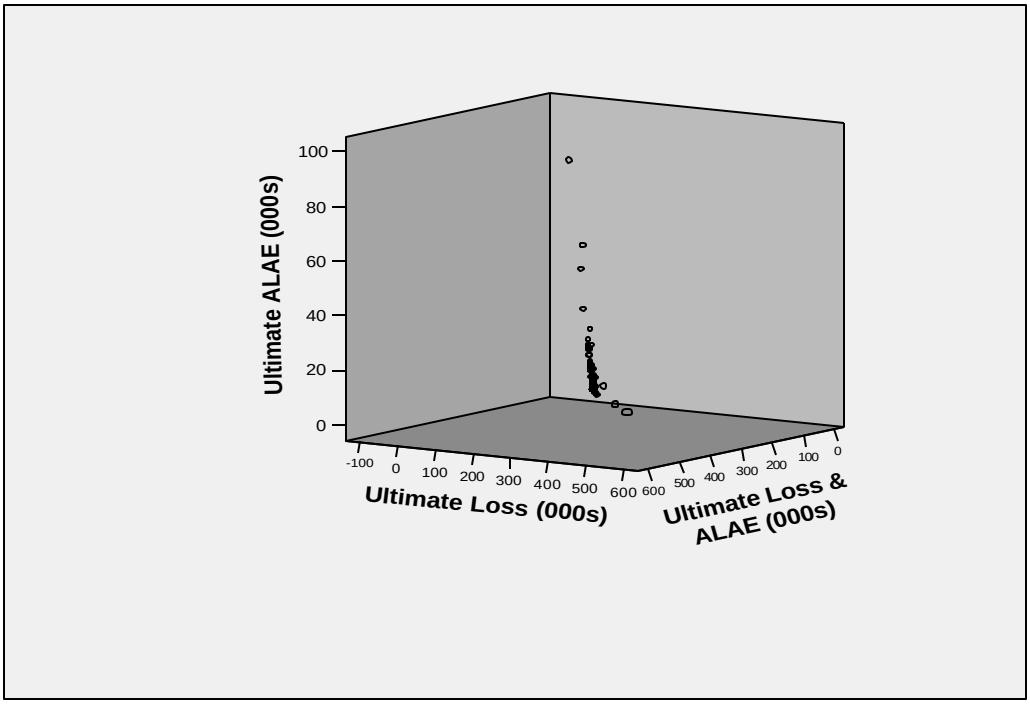


**Figure 2.2-1**

Each record in the data is also a dimension. When data is aggregated by accident year and development age in order to construct a loss development triangle, row-wise dimension reduction is taking place. The number of dimensions is reduced from the total number of records in the database to the number of cells in the loss development triangle.

### 2.2.2 K-means Clustering

In statistics, a formal procedure known as clustering is often used to perform dimension reduction along the rows. The objective of the technique is to group like records together. A common application of this method in property and casualty insurance is territory development. Policyholders are grouped into territories according to where they live and the territories are used in ratemaking. Both geographic information such as longitude and

latitude and demographic information such as population density can be used for the territorial clustering. Cluster analysis is an unsupervised learning method; there is no dependent variable. Rather, records with similar values on the variables used for clustering are grouped together. In the territory example, policyholders living in high population density zip codes in the southeastern part of a state might be grouped together into one territory. In text mining, clustering is used to group together records with similar words or words with similar meanings.

Many different techniques for clustering exist. One of the most common methods is k-means clustering. When using k-means clustering the analyst specifies the number of clusters he/she wants (a discussion of how to make this choice is deferred until later). A statistical measure of dissimilarity between records is used to separate records that are the most dissimilar and group together records that are the most similar. Different measures of dissimilarity are used for numeric data as opposed to categorical data. Text data is generally viewed as categorical. However, when terms are coded as binary indicator variables, it is possible to apply the techniques that are used on numeric data. Moreover, text mining is commonly applied to documents containing large collections of words, such as academic papers and e-mail messages. Some words appear multiple times in such text data and the number of times a word appears may be recorded and used for text analysis instead of a binary indicator variable. Dissimilarity measures for both numeric and categorical data will be presented.

One of the most common measures of dissimilarity for numeric variables is Euclidian distance. The formula for Euclidian distance is shown below in equation 2.1. The Euclidian distance between two records is based on the variable-wise squared deviation between the values of the variables of the two records.

$$d_{i,j} = \left(\sum_{k=1}^{m} (x_{i,k} - x_{j,k})^2\right)^{1/2} \quad \text{i, j = records, m = number of variables} \tag{2.1}$$

The second dissimilarity measure Manhattan distance is shown in equation 2.2. This measure uses absolute deviations rather than squared deviations.

$$d_{i,j} = \sum_{k=1}^{m} | x_{i,k} - x_{j,k} | \quad \text{i, j = records, m = number of variables} \tag{2.2}$$

Table 2.2.2-1 displays a calculation of both measures using two sample records. The first record has injury "broken ankle and sprained wrist" and the second record has injury "contusion to back of leg". Binary variables indicating the presence or absence of words after the parsing of text are the variables used in the measure.

**Euclidian and Manhattan Distance Between Two Records**

| Variable | **Record 1** BROKEN ANKLE AND SPRAINED WRIST | **Record 2** CONTUSION TO BACK OF LEG | **Squared** Difference | **Absolute** Difference |
|---|---|---|---|---|
| Back | 0.000000 | 1.000000 | 1 | 1 |
| Contusion | 0.000000 | 1.000000 | 1 | 1 |
| Head | 0.000000 | 0.000000 | 0 | 0 |
| Knee | 0.000000 | 0.000000 | 0 | 0 |
| Strain | 0.000000 | 0.000000 | 0 | 0 |
| Unknown | 0.000000 | 0.000000 | 0 | 0 |
| Laceration | 0.000000 | 0.000000 | 0 | 0 |
| Leg | 0.000000 | 1.000000 | 1 | 1 |
| Arm | 0.000000 | 0.000000 | 0 | 0 |
| Foot | 0.000000 | 0.000000 | 0 | 0 |
| Hand | 0.000000 | 0.000000 | 0 | 0 |
| Ankle | 1.000000 | 0.000000 | 1 | 1 |
| Shoulder | 0.000000 | 0.000000 | 0 | 0 |
| Hip | 0.000000 | 0.000000 | 0 | 0 |
| Left | 0.000000 | 0.000000 | 0 | 0 |
| Neck | 0.000000 | 0.000000 | 0 | 0 |
| Wrist | 1.000000 | 0.000000 | 1 | 1 |
| Cut | 0.000000 | 0.000000 | 0 | 0 |
| Fracture | 1.000000 | 0.000000 | 1 | 1 |
| Surgery | 0.000000 | 0.000000 | 0 | 0 |
| Finger | 0.000000 | 0.000000 | 0 | 0 |
| None | 0.000000 | 0.000000 | 0 | 0 |
| Broken | 1.000000 | 0.000000 | 1 | 1 |
| Trauma | 0.000000 | 0.000000 | 0 | 0 |
| Lower | 0.000000 | 0.000000 | 0 | 0 |
| Right | 0.000000 | 0.000000 | 0 | 0 |
| Total | | | 7 | 7 |
| Distance Measure | | | 2.65 | 7 |

**Table 2.2.2-1**

Dissimilarity measures specific for categorical variables also exist: Table 2.2.2-2 displays the notation for comparing two records on all their binary categorical variables. For instance, the sum of all variables for which both records have a one is shown as "a"

on the table. The counts of variables on which the two records agree are denoted "a" and "d". The counts of variables on which the two records disagree are denoted "b" and "c".

**Crosstabulation of Counts for Two Records Binary Variables**

| | | Record 1 | |
|---|---|---|---|
| | | 1 | 0 |
| Record 2 | 1 | a | b |
| | 0 | c | d |

**Table 2.2.2-2**

Simple matching is a dissimilarity measure that compares the total number of non matches to the total number of variables as shown in equation 2.3.

$$d_{i,j} = \frac{b+c}{a+b+c+d} \quad \text{i,j = records} \tag{2.3}$$

Another dissimilarity measure, shown in equation 2.4, is Rogers and Tanimoto. This measure gives more weight to disagreements than to agreements. In the example above (Table 2.2.2-1) where there are 7 disagreements and 19 agreements, the Rogers and Tanimoto dissimilarity measure is 0.43.

$$d_{i,j} = \frac{2(b+c)}{a+d+2(b+c)} \quad \text{i,j = records} \tag{2.4}$$

Instead of using a dissimilarity measure, some clustering procedures use a measure of similarity. A common measure of similarity is the cosine measure. The cosine statistic is a measure of covariance, but it is applied to records rather than to variables.

(2.5)

$$\text{cosine}_{i,j} = \frac{\sum_{k=1}^{m} (x_{i,k} * x_{j,k})}{\sqrt{\sum_{k=1}^{m} (x_{i,k}{}^2)} \sqrt{\sum_{k=1}^{m} (x_{j,k}{}^2)}} \quad \text{i,j = records, m = number of variables}$$

Rather than use binary indicator variables in the cosine calculation, this statistic typically uses a value referred to as the tf-idf statistic as $x_{ij}$ in equation 2.5. The tf-idf (term frequency – inverse document frequency) statistic is based on the frequency of a given term in the record. The static is normalized by being divided by the total number of times term appears in all records[4].

$$tf = \frac{n_i}{\sum_k n_k} \quad \text{n}_\text{i} = \text{number of times term i occurs,}$$
$$\text{tf-idf} = \frac{\text{tf}}{Df} \quad Df \text{ is the document frequency} \tag{2.6}$$

There are several ways to count the document frequency (denoted *Df* in equation 2.6) or the frequency of a term in a database (Wikipedia, 2005). A common method counts the number of records[5] in which the term appears divided by the total number of records. Sometimes the log of the inverse of the document frequency is used in the calculation. This statistic is more appropriate for applications involving larger collections of words, i.e., where each record is an entire document. The tf-idf method was not used in the analysis in this paper.

K-means clustering using Euclidian distance was applied to the matrix of extracted terms from the injury descriptions. Each cluster that is created from a k-means clustering procedure has a center referred to as the centroid. The centroid is the vector of average values for the cluster for each variable entering the clustering procedure. In the case of binary variables coded as either zero or one, the centroid is the cluster's frequency for each term or the proportion of all records in the cluster which contain the term. For example, the clustering procedure was used to create two classes or clusters. The clusters' frequencies for each term are displayed in the Table 2.2.2-3. From the table, it can be seen that none of the claims in Cluster 1 contain the word "back" and all of the claims in Cluster 2 contain the word. In addition, Cluster 1 contains a much higher percentage of claims with the words "contusion" "unknown" and "laceration" while Cluster 2 contains a much higher proportion

[4] In much of the text mining literature, the term "document" is a synonym for "record", because the unit of observation is often an entire document, such as a newswire article

[5] Frequently when this statistic is used, each record is a document. See footnote 4 above.

of records with the word "strain". Thus, when k-means clustering is used to create two clusters, a cluster with a high representation of claims with back injuries is partitioned from claims with other injuries.

**Frequencies for Two Clusters**

| Cluster Number | back | contusion | head | knee | strain | unknown | laceration |
|---|---|---|---|---|---|---|---|
| 1 | 0.00 | 0.15 | 0.12 | 0.13 | 0.05 | 0.13 | 0.17 |
| 2 | 1.00 | 0.04 | 0.11 | 0.05 | 0.40 | 0.00 | 0.00 |

**Table 2.2.2-3**

Frequency statistics for three clusters are displayed in Table 2.2.2-4. Again, one group, Cluster 2, is a cluster with 100% back injuries. Cluster 3 contains a high proportion of claims with knee injuries, while contusions and unknown are the most common injuries in Cluster 1. As will be discussed in more detail in Section 2-5, examining duster statistics such as those in Tables 2.2.2-3 and 2.2.2-4 assist the analyst in assigning labels to cluster.

**Frequencies for Three Clusters**

| Cluster Number | back | contusion | head | knee | strain | unknown | laceration |
|---|---|---|---|---|---|---|---|
| 1 | 0.00 | 0.17 | 0.14 | 0.04 | 0.05 | 0.16 | 0.19 |
| 2 | 1.00 | 0.04 | 0.11 | 0.05 | 0.40 | 0.00 | 0.00 |
| 3 | 0.00 | 0.07 | 0.04 | 0.48 | 0.09 | 0.00 | 0.05 |

**Table 2.2.2-4**

Viewing the statistics for three clusters versus two clusters, it is clear that there is refinement in the definition of the injury clusters when progressing from two to three clusters. Determining how many clusters to use is something of an art. If too many clusters are estimated, the model is over parameterized and is fitting noise as well as pattern. If too few clusters are created, the data are not adequately modeled. This topic is discussed in more detail in Section 2.2.4.

### 2.2.3 Hierarchical Clustering

Though less common than k-means clustering, hierarchical clustering is another common method applied in text mining to cluster terms in order to discover content (in this case, to

create features that can be used for further analysis). Hierarchical clustering is a stepwise procedure that begins with many clusters and sequentially combines clusters in close proximity to each other until no further clusters can be created. Typically, hierarchical clustering begins with every observation as a single cluster and terminates with one cluster containing all the records.[6] Hierarchical clustering procedures produce dendograms or tree-like visualizations of the stages of clustering which assist the analyst in determining the final number of clusters to select. Hierarchical clustering can be applied to either records or variables. Because the visualization of the results is easier to display, the results of clustering by variable are displayed in Figure 2.2-2. The figure displays the tree like figure or dendogram that results from clustering ten of the injury terms. For this dendogram, Euclidian distance was used.

The dendogram displays the cluster groupings created at each step of the clustering process. The left-hand side of the dendogram under the label CASE lists the variables and their numbers (based on order in the database). The left-most portion of the dendogram is a line representing a terminal branch of the tree. There is one branch for each variable, as each variable is its own cluster at the beginning of the clustering process. Moving to the right, the branch for arm and the branch for foot are connected, indicating that a new cluster is created in step one by combing arm and foot. Forming a cluster composed of these two variables indicates that the distances between arm and foot are smaller than the distances between any other possible combination of two variables. Next on the dendogram, the branch for leg is connected to the branch containing arm and foot, indicating that at the second step, a new cluster is created by combining leg with the arm-foot cluster. The stepwise process of creating new clusters by combining together smaller clusters at each step continues until there is only one cluster containing all the variables. This is shown at the right side of the dendogram where a branch containing back and strain are connected to a branch containing all other variables (i.e., at the next to last step the two group cluster partitions the terms "back" and/or "strain" from all other injuries).

Table 2.2.3-1 presents a matrix of proximity (i.e. distance) measures which underlie the dendogram clusters. These are the distances used to cluster the variables. For example, the distances between arm and foot, which are clustered together in the first step is 6.708. This

[6] Hierarchical clustering can also proceed in the opposite direction, from one cluster with all the data to many clusters

compares to the distance of 8.888 between arm and back, which only cluster together in the last step, where all variables are combined into one cluster.

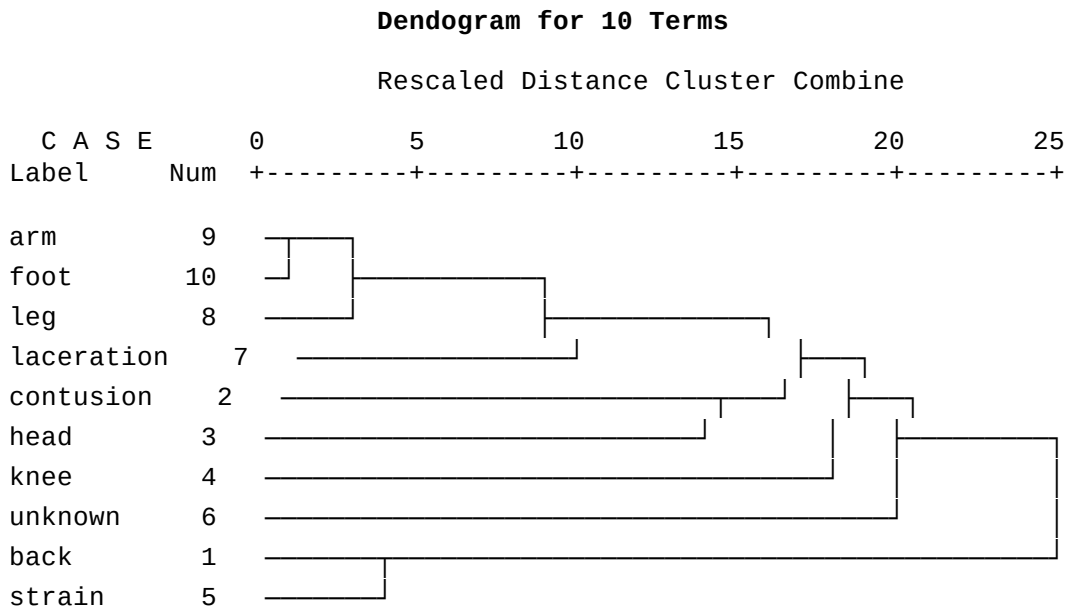


**Figure 2.2-2**

**Proximity Matrix**

| Case | Matrix File Input | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Back | Contusion | Head | Knee | Strain | Unknown | Laceration | Leg | Arm | Foot |
| back | 0.000 | 10.000 | 9.327 | 9.539 | 7.071 | 9.747 | 9.747 | 9.055 | 8.888 | 8.602 |
| contusion | 10.000 | 0.000 | 7.937 | 8.062 | 9.274 | 9.220 | 9.110 | 8.246 | 8.185 | 7.211 |
| head | 9.327 | 7.937 | 0.000 | 9.055 | 9.000 | 8.944 | 8.124 | 8.185 | 8.000 | 7.810 |
| knee | 9.539 | 8.062 | 9.055 | .000 | 8.307 | 8.832 | 8.602 | 8.307 | 8.000 | 7.681 |
| strain | 7.071 | 9.274 | 9.000 | 8.307 | 0.000 | 8.775 | 8.775 | 7.746 | 7.810 | 7.616 |
| unknown | 9.747 | 9.220 | 8.944 | 8.832 | 8.775 | 0.000 | 8.718 | 8.185 | 8.000 | 7.550 |
| laceration | 9.747 | 9.110 | 8.124 | 8.602 | 8.775 | 8.718 | 0.000 | 7.550 | 8.000 | 7.141 |
| leg | 9.055 | 8.246 | 8.185 | 8.307 | 7.746 | 8.185 | 7.550 | 0.000 | 7.000 | 6.928 |
| arm | 8.888 | 8.185 | 8.000 | 8.000 | 7.810 | 8.000 | 8.000 | 7.000 | 0.000 | 6.708 |
| foot | 8.602 | 7.211 | 7.810 | 7.681 | 7.616 | 7.550 | 7.141 | 6.928 | 6.708 | 0.000 |

**Table 2.2.3-1**

The hierarchical clustering of the terms in the data provides insight into word combinations in the data that tend to occur together and or tend to be associated with similar injuries. However, for the purpose of classifying records into injury categories, it is more typical to cluster case-wise rather than variable-wise. Thus, hierarchical clustering was also used to cluster the injury description records.

### 2.2.4 Number of Clusters

Determination of the number of clusters to retain is often something of art. One approach involves viewing the cluster centers to determine if the clusters from a given grouping appear meaningful. Another procedure for determining the number of clusters involves comparing the performance of different clustering schemes on an auxiliary target variable of interest. Here, ultimate incurred losses and ALAE is one variable of interest that may help in the decision. Figure 2.2-3 displays how the mean ultimate loss and ALAE varies by cluster for four and eight cluster groupings.

A forward stepwise regression was run to determine the best cluster size. Stepwise regression is an automated procedure for selecting variables in a regression model. Forward stepwise regression begins with a null model or model that has no predictors. The procedure then tests all possible independent variables that can be used on a one-variable regression model. The variable which improves the goodness of fit measure the most is the variable entered in step one. In step two, all 2-variable regressions are fit using the variable selected in step one and the variables not selected in step one. The variable which produces the largest improvement in goodness of fit is then selected and entered into the model. The process continues until no further significant improvement in fit can be obtained.

**Average Ultimate Loss and ALAE by Cluster for 4 and 8 Clusters**

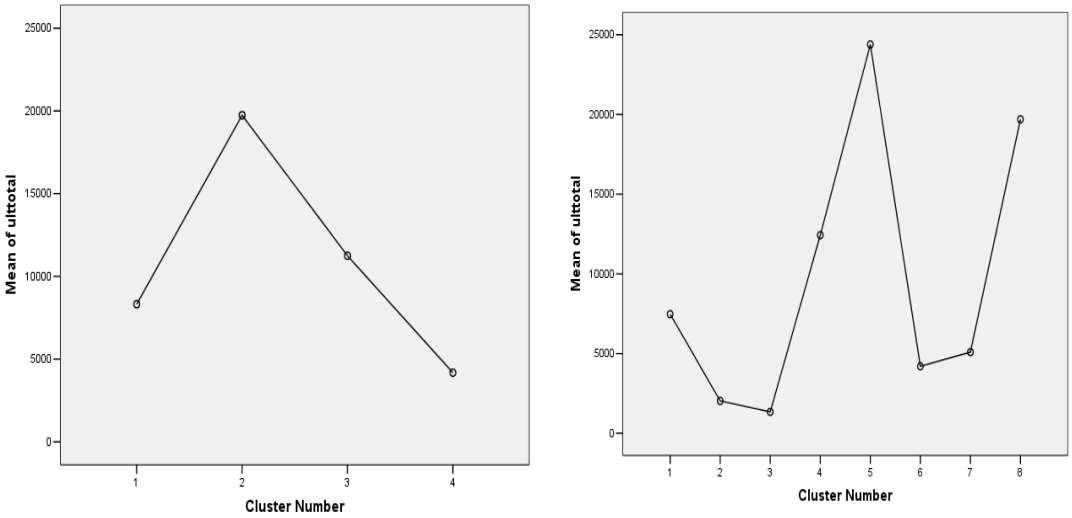


**Figure 2.2-3**

A common goodness of fit measure used in stepwise regression is the F-statistic:

$$F = \frac{\text{MS Regression}}{\text{MS Residual}} = \frac{SS_{reg} / p}{SS_{res} /(N - p - 1)} \qquad (2.7)$$

where p=number of variables,
N=number of observations,
SS=sum of squared deviation

The F statistic is the ratio of mean squared error of the regression (the amount of variance explained by the regression) divided by the mean square error of the residual (the amount of unexplained variation). When used in stepwise regression, after the first variable is entered, the change in F statistic is used. The user typically selects a significance level such as 5% that is used as a threshold for entering variables into the regression.

When using stepwise regression to select the number of clusters to use, the possible predictor variables in the regression are the clusters created by 2 category cluster, 3 category

cluster, etc. Since the objective is to find the optimal number of clusters, the regression is run on each of the category cluster variables and the category cluster with the best fit is selected. For the purposes of this paper, only the first step of the stepwise regression was performed, i.e., only the one variable supplying the best fit of all the one variable regressions was retained.[7] Stepwise regression provides a quick and efficient method for determining the number of clusters. The stepwise procedure determined that a regression with seven groups produced the best fit. Regression also ascertained that k-means clustering produced clusters that were better predictors of ultimate losses and ALAE than hierarchical clustering.

A more formal approach is to use a statistical test to determine the optimum number of clusters. One such test is the BIC (Swartz Bayesian Information Criterion) (Chen and Gopalakrishnan, 2004). The statistic is used to compare two models at a time. The statistic chooses between a simpler model and a more complex model by comparing their adjusted or penalized likelihood function. A penalty related to the number of variables in the model is applied in order to control for overfitting. When applying the statistic, it is common to treat the data as if from a multivariate normal distribution:

$$\mathbf{X} \sim N(\boldsymbol{\mu}, \mathbf{S}) \qquad (2.8)$$

where $\mathbf{X}$ is a vector of random variables $\boldsymbol{\mu}$ is the centroid (mean) of the data and $\mathbf{S}$ is the variance-covariance matrix

[7] Because of the small size of the data only one injury code variable was created.

The formula for the BIC statistic is:

$$BIC = \log L(X, M) - l\,\frac{1}{2}\text{p*log(N)} \qquad (2.9)$$

where log(L(X,M)) is the logliklihood function for a model, p is the number of parameters, N the number of records, $l$ is a penalty parameter, often equal to 1

For a cluster analysis, each cluster has a likelihood function based on the cluster's centroid and variance-covariance matrix. For instance, in comparing a one-group cluster to a two-group cluster, a likelihood based on the overall centroid of all the data and the overall variance-covariance matrix is compared to a two group likelihood based on the centroids and variance-covariance matrices of the two clusters. The second model has twice as many parameters as the first. If the BIC increases significantly using two clusters compared to one cluster, a two group clustering is preferred.

Most of the software used in this analysis did not implement the BIC statistic to determine cluster size. However, one procedure, the SPSS two-step clustering procedure intended for categorical and mixed categorical-numeric data did implement the procedure. A two-step clustering procedure breaks the clustering process into two steps 1) create a dissimilarity matrix which may be done differently for categorical as opposed to numeric data and 2) use the dissimilarity matrix to cluster the data. When this procedure was applied, it produced a clustering with three groups. The two-step clusters had a significant correlation with ultimate losses and ALAE, though this correlation was not as high as that for the best k-means cluster.

The end result of clustering of the claim description field in the data is to introduce one new feature or variable. This variable is a categorical variable indicating to which of the cluster groupings or classes a record is assigned. This new variable can be viewed as an injury type coding. In the application in section 2.2.6, the seven cluster grouping will be used, but other choices could have been made. Note that while only one cluster grouping of the injury descriptions was selected, there may be situations where the analyst prefers to use multiple new features derived from the clustering procedure, each with a different number of groups.

### 2.2.5 Naming the Clusters

For each cluster, it can be informative to determine which word or words are important

in defining the cluster. Examining the frequencies of each word for each of the clusters can be used to gain insight into the clusters. Figure 2.2-4 displays the frequencies of the words "back" and "strain" for the seven-group cluster. The graph is a population pyramid. The graph displays visually with bars a crosstabulation of back versus strain by cluster group. That is, a bar displays the count for a zero or one on back, versus zero or one on strain for each of the seven injury cluster groupings. The bars appearing under one for back, one for strain or one for both back and strain denote injury groups that contain the words back, strain or both. From the graph it can be seen that Cluster 4 has a relatively high count of both the words back and strain and Cluster 6 has a high representation of the word back, but not strain.

Table 2.2.5-1 presents frequencies of key words for the seven-group cluster. The table displays the proportion of records for each cluster which contain the words. Words that have high representation within a cluster have been highlighted. From the table it can be seen that Cluster 1 has a high representation of the word unknown. Cluster 2 has a high representation of the word contusion. Cluster 4 has a high representation of the words back and strain. Cluster 7 also has a high representation of the word strain, but the word back has a low representation. A conclusion is that Cluster 4 appears to be back strains while Cluster 7 is largely other strains, and includes a high representation of the word leg.

**Frequencies of the Words Back and Strain by Cluster**

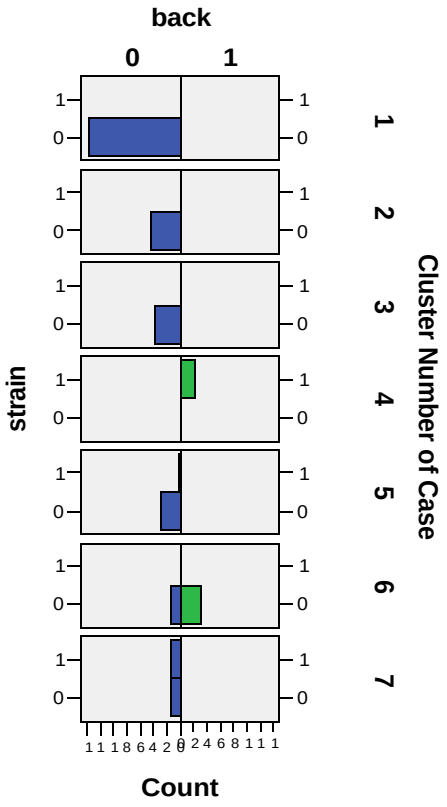


**Figure 2.2-4**

**Frequency of Terms by Cluster**

| Cluster | Back | Contusion | head | knee | strain | unknown | laceration | Leg |
|---|---|---|---|---|---|---|---|---|
| 1 | 0.000 | 0.000 | 0.000 | 0.095 | 0.000 | 0.277 | 0.000 | 0.000 |
| 2 | 0.022 | 1.000 | 0.261 | 0.239 | 0.000 | 0.000 | 0.022 | 0.087 |
| 3 | 0.000 | 0.000 | 0.162 | 0.054 | 0.000 | 0.000 | 1.000 | 0.135 |
| 4 | 1.000 | 0.000 | 0.000 | 0.043 | 1.000 | 0.000 | 0.000 | 0.000 |
| 5 | 0.000 | 0.000 | 0.065 | 0.258 | 0.065 | 0.000 | 0.000 | 0.032 |
| 6 | 0.681 | 0.021 | 0.447 | 0.043 | 0.000 | 0.000 | 0.000 | 0.000 |
| 7 | 0.034 | 0.000 | 0.034 | 0.103 | 0.483 | 0.000 | 0.000 | 0.655 |
| Weighted Average | 0.163 | 0.134 | 0.120 | 0.114 | 0.114 | 0.108 | 0.109 | 0.083 |

**Table 2.2.5-1**

A procedure involving tabulation of the frequency of the words within the cluster can be automated. The most commonly occurring words can be identified and used to label the cluster.

### 2.2.6 Using the Features Derived from Text Mining

A major objective of text mining is to create new information that has predictive value. The simple illustration in this paper mined an injury description field and assigned each claim to one of seven cluster groups based on the words in the injury description. The cluster group is a new independent variable that can be used to predict a dependent variable of interest to the analyst. Potential variables of interest in a claims database include financial variables such as losses and loss adjustment expenses, whether or not there has been subrogation or recovery on the claim and whether or not the claim is likely a fraud or abuse claim. The database used in this exercise is representative of what might be available in cases where a third party claims adjuster supplied data to a self insured entity. It is therefore smaller and less rich than what one would find in a large insurance company database. This simple example focuses on the financial variables in the data.

The example uses the new injury feature added by the text mining procedure to predict the likelihood that a claim will be a serious claim. One application of data mining in the literature (Derrig, 2004) uses models to score claims early in the life of the claim. The objective is to identify claims that are likely to be the most costly to the company and apply more resources to those claims. For this analysis, a serious claim is defined as a claim whose total losses plus allocated loss adjustment expenses exceeds $10,000. Approximately 15% of the claims in the data exceed this threshold. A histogram of claim severities is shown in Figure 2.2-5. The histogram indicates that the severity distribution is right skewed and heavy tailed. Approximately 98% of loss dollars are due to claims defined as serious. (See the pie chart in Figure 2.2-6).

**Histogram of Claim Severity**

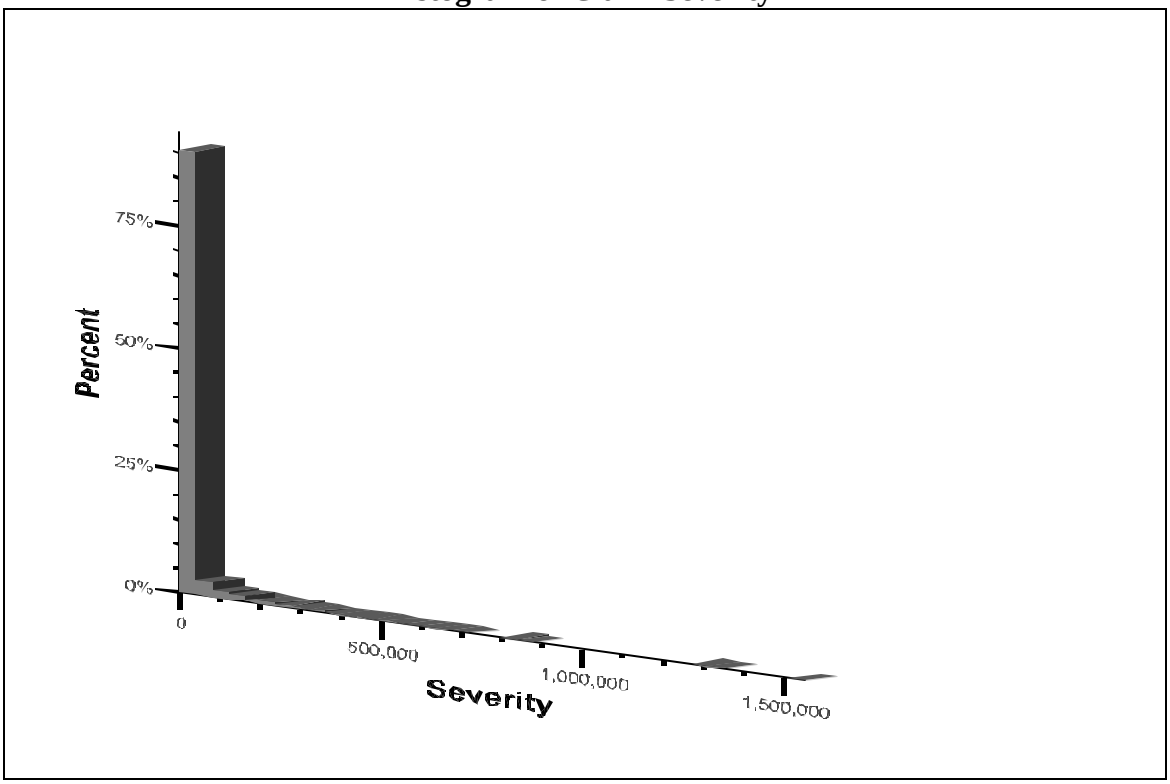


**Figure 2.2-5**

**Percent of Loss Dollars: Serious vs. Non Serious Claims**

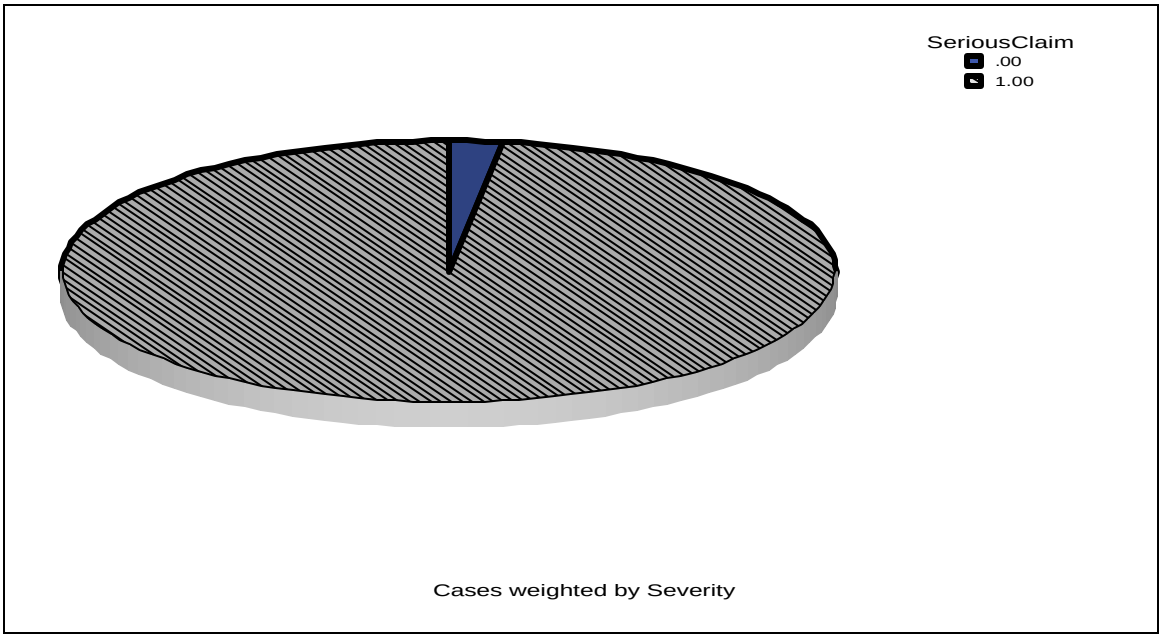


**Figure 2.2-6**

Logistic regression was used to predict the likelihood of a claim being a serious claim. Much of the current actuarial literature dealing with modeling large complex databases focuses on generalized linear models (See CAS Discussion Paper Program, 2004). A modeling procedure that is one of the options found within the family of generalized linear models is logistic regression. The logistic regression procedure functions much like ordinary linear regression, but under logistic regression the dependent variable is categorical or ordered categorical, not numeric. Logistic regression is a popular choice from the family of generalized linear models for performing classification. With categorical variables, a value of one can be assigned to observations with a category value of interest to the researcher (i.e., serious claims) and zero to all other claims. Typically the objective is to score each observation with a probability the claim will fall into the target category, category one. The probability the claim will have a value of 1 lies in the range 0 to 1. This probability is denoted p(y). The model relating p(y) to a vector of independent variables **x** is:

$$\ln\left(\frac{p(y)}{1-p(y)}\right) = b_0 + b_1 X_1 + b_2 X_2 \ldots + b_n X_n \qquad (2.10)$$

The ratio $\frac{p(y)}{1-p(y)}$ is referred to as the odds ratio and the quantity $\ln\left(\frac{p(y)}{1-p(y)}\right)$ is known as the logit function or logit transformation.

The reader is referred to the extensive literature on logistic regression for further details (Hosmer 1989, Venibles and Ripley 1999). Once a linear model has been fit, the predicted value will be on the logit transformed scale. To use the predictions as probabilities, they must be transformed back to the original scale. If $\hat{f}(\mathbf{x})$ is the logistic predicted value, the transformation $e^{\hat{f}(\mathbf{x})}/(1+e^{\hat{f}(\mathbf{x})})$ must be applied.

Other analytical methods such as CART (Brieman *et al*., 1990) could also be applied although the data in this example likely does not lend itself to complex approaches meant for larger, complex databases. Two variables were used to predict the probability of a serious claim: attorney involvement and the injury variable derived from text mining. Because the sample database is relatively small only a main effects[8] model was fit (a model with interaction terms was tested and found not to be significant). This means the model fit was of the following form:

$$Y = B_0 + B_1 Attorney + B_2 Injury\_Group \tag{2.11}$$

The injury group used in the model is the injury grouping of seven clusters created by the text mining analysis. The attorney variable denotes whether an attorney is involved in the claim.

The logistic regression found both variables to be significant. The table below shows the average model probability of a serious claim for both the serious and non-serious claims. It can be seen that the model predicts a much higher probability, on average, for the serious groups of claims than the non-serious group of claims.

[8] In a main effects model there are no interactions incorporated into the model.

**Mean Probability of Serious Claim vs. Actual Value**

| Avg Prob | Actual Value | |
|---|---|---|
| | 1 | 0 |
| | 0.31 | 0.01 |

**Table 2.2.6-1**

One other application of the text variable is illustrated. A simple analysis of variance (ANOVA) was used to predict ultimate losses and ALAE. An ANOVA is a linear model where the dependent variable is numeric and the independent variables are categorical. ANOVA is like a linear regression with categorical predictor variables. The form of the model is:

$$Y = B_0 + B_1 Attorney + B_2 Injury\_Group + B_3 Attorney * Injury\_Group \quad (2.12)$$

where Y is ultimate losses and ALAE trended to a common date

Note this model includes the interaction term attorney * injury group. The results are displayed in the Table 2.2.6-2. In this regression, both attorney involvement and injury group as well as the interaction between attorney and injury are significant. As an alternative to the classification procedure illustrated in the previous example, such a regression could be used to identify serious claims (i.e., the claims with high predicted values for the ultimate losses and ALAE). Another application of models that predict ultimate losses and ALAE is estimating reserves for insurance exposures. Heckman (1999) and Taylor (2004) introduced methods of reserving that utilized individual claims data. There are two components to using claim data to estimate ultimate losses for a reserving application:

- Estimate ultimate losses for claims already reported using the individual information for each claim's independent variables. Historic information on more mature claims is used to develop a model for less mature claims
- Estimate ultimate losses for claims that have occurred, but because of a long reporting lag, have not yet been reported. In order to estimate ultimate values for unreported claims the actuary needs:

- An estimate of unreported claims (perhaps derived from a claim development triangle)
- An estimate of the proportion of claims in each category of the key predictor variables (i.e., an estimate of the proportion within each attorney/injury type combination). Recent historical patterns could be used to derive such estimates

**Results for Regression of Ultimate Losses and ALAE on Attorney and Injury**

*Parameter Estimates*

*Dependent Variable: Ultimate Loss & ALAE*

| *Parameter* | *B* | *Std. Error* | *t* | *Sig.* | *95% Confidence Interval* | |
|---|---|---|---|---|---|---|
| | | | | | Lower Bound | Upper Bound |
| Intercept | 2975.08 | 74.56 | 39.90 | 0.00 | 2790.20 | 3159.97 |
| [attorney=.000000] | -2924.58 | 453.27 | -6.45 | 0.00 | -4048.49 | -1800.67 |
| [attorney=1.000000] | 0.00 | . | . | . | . | . |
| [QCL6= 1] | 18426.20 | 80.08 | 230.10 | 0.00 | 18227.64 | 18624.77 |
| [QCL6= 2] | 10504.67 | 153.40 | 68.48 | 0.00 | 10124.31 | 10885.03 |
| [QCL6= 3] | 6506.90 | 214.04 | 30.40 | 0.00 | 5976.17 | 7037.63 |
| [QCL6= 4] | 1175.95 | 112.17 | 10.48 | 0.00 | 897.81 | 1454.08 |
| [QCL6= 5] | 37081.94 | 89.64 | 413.67 | 0.00 | 36859.67 | 37304.22 |
| [QCL6= 6] | 74620.90 | 79.82 | 934.92 | 0.00 | 74422.99 | 74818.81 |
| [QCL6= 7] | 0.00 | . | . | . | . | . |
| [attorney=.000000] * [QCL6= 1] | -16537.20 | 530.17 | -31.19 | 0.00 | -17851.81 | -15222.59 |
| [attorney=.000000] * [QCL6= 2] | -10123.91 | 556.53 | -18.19 | 0.00 | -11503.88 | -8743.95 |
| [attorney=.000000] * [QCL6= 3] | -3934.19 | 607.54 | -6.48 | 0.00 | -5440.64 | -2427.74 |
| [attorney=.000000] * [QCL6= 4] | -675.90 | 719.76 | -0.94 | 0.35 | -2460.60 | 1108.80 |
| [attorney=.000000] * [QCL6= 5] | -36860.96 | 673.40 | -54.74 | 0.00 | -38530.71 | -35191.21 |
| [attorney=.000000] * [QCL6= 6] | -63147.92 | 567.22 | -111.33 | 0.00 | -64554.39 | -61741.44 |
| [attorney=.000000] * [QCL6= 7] | 0 | . | . | . | . | . |
| [attorney=1.000000] * [QCL6= 1] | 0 | . | . | . | . | . |
| [attorney=1.000000] * [QCL6= 2] | 0 | . | . | . | . | . |
| [attorney=1.000000] * [QCL6= 3] | 0 | . | . | . | . | . |
| [attorney=1.000000] * [QCL6= 4] | 0 | . | . | . | . | . |
| [attorney=1.000000] * [QCL6= 5] | 0 | . | . | . | . | . |
| [attorney=1.000000] * [QCL6= 6] | 0 | . | . | . | . | . |
| [attorney=1.000000] * [QCL6= 7] | 0 | . | . | . | . | . |
| A | This parameter is set to zero because it is redundant. | | | | | |

**Table 2.2.6-2**

## 3. RESULTS AND DISCUSSION

In this paper, a very simple example of text mining was used as an illustration of the underlying concepts and methods. The illustration has shown that the basic procedures underlying text mining are straightforward to understand and implement. The two key

technologies that are used are 1) string manipulation and processing functions that are part of nearly all programming languages and 2) classical statistical procedures for dimension reduction, such as clustering, that are included within nearly all statistical software packages.

In the illustration, text mining was used to add an injury type code to a database that contained only a free form text field describing the injury.

The injury description was then used as an independent variable in two simple predictive models. In more realistic situations text mining has the potential to add significantly to the information available to the analyst in large modeling projects. For instance, many large insurance company databases contain one or more free form claim description fields or narratives describing the accident and the circumstances associated with the accident. These narratives often contain information not contained in injury, cause of loss or other coding. This may be particularly true when new types of claims or new patterns of claiming behavior are beginning to emerge Ellingsworth and Sullivan (2003) describe applications used to provide an understanding of rising homeowner claims and suspicious and possibly fraudulent auto claims at a large insurance company. When analytical approaches using only structured information coded into the company's database were unsuccessful in explaining the patterns, they turned to text mining. Ellingsworth and Sullivan provided the following hypothetical example of text from a claim description field:

> "The claimant is anxious to settle; mentioned his attorney is willing to negotiate. Also willing to work with us on loss adjustment expenses (LAE) and calculating actual cash value. Unusually familiar with insurance industry terms. Claimant provided unusual level of details about accident, road conditions, weather, etc. Need more detail to calculate the LAE."

Certain terms in the text such as "anxious", "settle" and "familiar" may provide clues to suspicious claims that cannot be found in the structured data in the claims database. Mining the text data for such terms significantly improved the ability of Ellingsworth and Sullivan to model the patterns in the data.

Text mining has become sufficiently prominent that the major vendors of statistical and data mining software tools (such as SAS, SPSS and Insightful) offer text mining products. Some of these tools are very powerful and are capable of processing data from large document collections. While a discussion of software tools for text mining is postponed to the Appendix of this paper, acquisition of powerful text mining software may be unnecessary for smaller applications such as in this paper. That is, when the "documents"

being analyzed are relatively modest in size, as many claim description data are, methods developed for applications on larger documents such as academic papers and news service articles may be more than is needed. The analyses in this paper were performed using free text mining software along with statistical procedures available in SPSS13.0 and S-PLUS 6.2. The author believes that there are many situations where text mining can be used to augment the amount of information available for analysis and that for smaller applications, it is unnecessary to acquire expensive specialized tools.

## 4. CONCLUSIONS

The purpose of this paper is to educate actuaries on the potential for using text mining for insurance applications. That is, the intent is to provide a basic introduction to the new area of text mining. It was shown that relatively uncomplicated methods underlie the main procedures used to perform text mining. It is widely believed that a large percentage of data is contained in unstructured form. Text mining has the potential to add significantly to the amount of data available for analysis. Some of this data includes adjuster claim description notes, loss prevention specialist notes and underwriter notes.

The field of text mining is one that is undergoing rapid development. New methods are being developed to improve on simple clustering as a means of classifying text data. These include methods based on discriminant analysis (Howland and Park, 2004), methods that use principal components analysis and single value decomposition (Snellert and Blondel, 2004), and linkage based methods that dynamically update (Aggarwal, 2005). Note that the methods used in this paper perform row-wise dimension reduction and cluster similar records. Methods based on factor analysis, principal components analysis and single value decomposition can perform column-wise or term-wise dimension reduction. While these methods were not described or illustrated in this paper, they show promise for improving the classification of text information. Another area under development that may expand the applicability of text mining is handwriting recognition and optical character recognition (Wikipedia, 2005). Many PDAs read handwritten entries. Microsoft Windows XP and Office XP also have handwriting recognition capability. Improvements in handwriting and optical character recognition could permit scanning and mining of handwritten and typed notes currently stored in paper files and not currently accessible from computerized databases.

**Acknowledgment**

The author acknowledges the assistance of Michael Francis, Virginia Lambert, Rudy Palenik and Jane Taylor in editing this paper.

## Appendix A – Software for Text Mining

This appendix describes the author's experiences with several text mining tools. The tools covered are 1) commercial text mining products, 2) a free text mining tool and 3) programming languages. The author found only categories 2) and 3) to be useful in this analysis, although commercial text mining products may prove invaluable in tasks involving larger, more complex data sets.

The task of locating appropriate software for use in text mining proved to be something of a challenge. A number of software options were investigated in preparation for undertaking the analysis in this paper. Davi *et al.* (2005) gave a favorable review to two text mining packages; WordStat and SAS Text Miner. As the SAS Text Miner package is sold bundled with the Enterprise Miner, a large, relatively expensive data mining suite intended for large application, no attempt was made to acquire or test it. Therefore WordStat, a modestly price product was investigated. The WordStat web site allows prospective customers to download a trial version of the software. The user can use the demo software for 30 days or 10 uses. The latter limitation of 10 uses proved to be the more severe limiting factor. During this study about 5 of the 10 uses were consumed in figuring our how to read data into the text mining module of the software. Once the data were read, simple term extraction was performed and some simple descriptive statistics were created. However, the author was unable to apply clustering procedures to create an injury description feature or to output terms for analysis in other software. Thus, other options were investigated as WordStat was unable to provide the functionality needed for this study.

Other vendors of text mining software (SPSS and Insightful) felt their text mining software was inappropriate for the purposes of this study[9]. After relatively little success with other options, the free package TMSK was used to perform many of the tasks for the text mining analysis in this paper.

TMSK is a free product available to purchasers of the book Text Mining (Weiss *et al.*

[9] The software is intended primarily for much larger scale complex applications, and is intended for use on a server making it difficult to install and use initially. Many of these are not major issues when being applied to large scale applications for which these packages are intended.

2005). It can be downloaded from the author's web site using passwords supplied with the book. This software is very handy for performing term extraction. It comes with lists containing stop words and stem words that are automatically applied during running of the program and can be used to do feature creation using k-means clustering. Certain other analytical tasks not covered in this paper are also included. However, a certain amount of persistence is required to obtain useful results from the software. Some of these features of this program the user needs to be aware of are:

- The user must have the programming language Java on his/her computer. Java can be downloaded for free from the Sun Microsystems web site: http://java.sun.com/.
- The program will only run in DOS mode, i.e., in the command window. On many windows systems the command prompt is accessed by looking under accessories in the program listing.
- The program will only read xml files. For this analysis, the injury description field of the example data was saved to an xml file format within Microsoft Excel. More recent versions of Adobe Acrobat can also save text in xml format.
- The results of term extraction are output to what is referred to as a "sparse vector". Table A-1 displays a snapshot of what a sparse vector looks like. The sparse vector is a condensed representation of the terms extracted, containing an entry only when the term is present for the record. The notation on the first row of Table A-1 indicates that for record 1 of the example data, Term 15 occurred once, Term 20 occurred once and Term 21 occurred once. The analytical procedures included with TMSK read and process the sparse vector data. However, in order to use a statistical procedure other than the ones that come with TMSK, it is necessary to read and parse this output in some other programming language and associate the correct term and correct record with the position indicator and row from the table.
- The manual indicates that the user can add additional stem words to the list maintained by TMSK. However, during this analysis, this feature did not appear to function, so some additional stemming was performed in other software.

**Sparse Vector Representation of Terms Extracted**

| | | |
|---|---|---|
| 15@1 | 20@1 | 21@1 |
| 1@1 | 2@1 | 8@1 |
| 6@1 | | |
| 1@1 | 23@1 | |
| 1@1 | | |

**Table A-1**

Most of analysis after term extraction was performed in SPSS and S-PLUS. (TMSK could have been used for clustering, but more output and analysis than this package provides was needed for this paper). Most general purpose statistical packages provide clustering procedures that can be used in feature creation.

Text miners may also want to program the steps required for term extraction themselves. Most programming languages, including those popular for statistical applications, such as S-PLUS, R (an open source analysis package), and SAS contain string manipulation function that can be used to parse words from text data. Initially, some investment in programming effort would be required to eliminate stopwords and perform stemming. The book *Text Mining* (Weiss *et al.*, 2005) contains pseudo code that can be referenced for programming many of the text mining procedures.

Two programming languages, Perl and Python have become popular for processing text data. Both languages are free and can be downloaded from the appropriate web site (www.perl.com and www.python.org). Because these languages are used so frequently for text processing, functions have already been developed and made available to users that handle many of the term extraction tasks.

In summary, text mining is a relatively new application, and software for performing text mining is relatively undeveloped compared to other data mining applications. When using one the data mining suites, the text miner may want to use text mining capabilities sold with the suite. These have not been tested as part of this study. The text miner may also wish to use free software or one of the programming languages that specialize in text processing.

# 5. REFERENCES


[1] Aggarawal, C, "On Learning Startegies for Topic Specific Web Crawling" in *Next Generation of Data Mining Applications*, Wiley, pp. 447 – 468, 2005

[2] Blench, M and Proulx, L, "Global Public Health Intelligence Network (GPHIN): An Early Warning Intelligence Information Management Model", Government Texhnologies Forum and Health Technologies Forum, August 30, 2005

[3] Casualty Actuarial Society Discussion Paper Program on Applying and Evaluating Generalized Linear Models, 2004

[4] Chen, S and Gopalakrishnan, "Speaker, Environment and Channel Change Detection and Clustering via the Bayesian Information Criterion", www.nist.gov/speech/publications/darpa98/html/bn20/bn20.htm , 1998

[5] Davi, A, Haughton, D, Nasr, N, Shah, G, Skaletsky , M and Spack, R., "A Review of Two Text Mining Packages: SAS Text Mining and WordStat", *American Statistician*, Feb 2005

[6] Derrig, R, Weisberg H, and Chen, X, 1994, "Behavioral Factors and Lotteries Under No-Fault with a Monetary Threshold: A Study of Massachusetts Automobile Claims", *Journal of Risk and Insurance*, June, 1994, 61:2: 245-275

[7] Derrig, R and Ostaszewski, "Fuzzy Techniques of Pattern Recognition in Risk and Claim Classification", *Journal of Risk and Insurance*, pp. 447-482, 1995

[8] Derrig, R, "Fraud Fighting Actuaries: Mathematical Models for Insurance Fraud Detection", 2004 CAS Predictive Modeling Seminar

[9] De Vel, "Mining E-mail Authorship", KDD Workshop on Text Mining, 2000

[10] Howland, P, and Park, H, "Cluster Preserving Dimension Reduction Methods for Efficient Classification of Text Data", in *Survey of Text Mining*, Springer, 2004

[11] Ellingsworth, M, "An Overview of Fraud Modeling", Insurance Fraud Conference, November 2002

[12] Ellingsworth, M and Sullivan, D, "Text Mining Improves Business Intelligence and Predictive Modeling in Insurance", DMReview, July, 2003

[13] Francis, L, "Neural Networks Demystified", CAS Discussion Paper Program, Winter 2001

[14] Hayward, G, "Mining Insurance Data to Promote Traffic Safety and Better Match Rates to Risk", *Casualty Actuarial Society Forum*, Winter 2002, *pp. 31 – 56.*

[15] Hastie, T, Tibshirani, R and Friedman, *J.*, *The Elements of Statistical Learning*, Springer, 2001

[16] Heckman, P., "Aggregate Reserve Distribution from Detailed Process Model for Individual Claims", Casualty Loss Reserve Seminar, 2000

[17] Hoffman, P, *Perl for Dummies*, Wiley, 2003

[18] Holler, K, Somner, D, and Trahair, G, "Something Old, Something New in Classification Ratemaking With a New Use of GLMs for Credit Insurance", *Casualty Actuarial Society Forum, Winter 1999, pp. 31-84*.

[19] Hosmer, D, and Lemshow, S, *Applied Logistic Regression*, John Wiley and Sons, 1989

[20] Jacoby, William, *Data Theory and Dimension* Analysis, Sage Publications 1991.

[21] Ellingsworth, M and Sullivan, D, "Text Mining Improves Business Intelligence and Predictive Modeling in Insurance", DMReview, July, 2003

[22] Kauffman, L, Rousseeuw, *Finding Groups in Data, Wiley*, 1990

[23] Kolyshkina, I, "Text Mining Challenges and Treenet Impact of Textural Information on Claims Cost Prediction", presentation at Second International Salford Systems Data Mining Conference, March 2005

[24] Manning, C, Schutze, H, *Foundations of Statistical Natural Language Processing*, MIT Pres, 1999

[25] Metz, D, *Text Processing in Python*, Addison Wesley, 2003

[26] Miller, T, *Data and Text Mining*, Pearson Prentice Hall, 2003

[27] Mitchell, R, "Anticipation Game", Computer World, 2005

[28] Robb, D, "Text Mining Tools Take on Unstructured Data, *Computer World*, June 2004

[29] Senellart, P and Blondel, V, "Automatic Discovery of Similar Words" in *Survey of Text Mining*, Springer, 2004

[30] Sullivan, D, *Document Warehousing and Text Mining*, Wiley, 2001

[31] Taylor, G and McGuire, G, "Loss Reserving with GLMs: A Case Study", *2004 CAS Discussion Paper Program*
[32] Venebles, W.N. and Ripley, B.D., Modern *Applied Statistics with S-PLUS*, third edition, Springer, 1999
*[33]* Weisberg, H and Derrig, R, "Pricing No-Fault and Bodily Injury Coverages Using Micro-Data and Statistical Models", *Casualty Actuarial Society Forum, 1993*
[34] Weiss, Shalom, Indurkhya, Nitin, Zhang, Tong and Damerau, Fred, *Text Mining*, Springer, 2005
[35] Wikipedia, www.wikipedia.org, 2005

## Biography of the Author

**Louise Francis** is a Consulting Principal at Francis Analytics and Actuarial Data Mining, Inc. She is involved data mining projects as well as conventional actuarial analyses. She has a BA degree from William Smith College and an MS in Health Sciences from SUNY at Stony Brook. She is a Fellow of the CAS and a Member of the American Academy of Actuaries. She is currently chair of the CAS Committee on the Theory of Risk, and is a frequent presenter at actuarial and industry symposia. She published three previous papers in the Data Management, Quality and Technology Call Paper Program: "Neural Networks Demystified" (2001) and "Martian Chronicles: Is MARS Better than Neural Networks" (2003) and "Dancing with Dirty Data: Methods for Exploring and Cleaning Data (2005).