HARVARD | BUSINESS | SCHOOL

Machine Learning Methods for Strategy Research

Mike Horia Teodorescu

Working Paper 18-011

Machine Learning Methods for Strategy Research

Mike Horia Teodorescu
Harvard Business School

Working Paper 18-011

Machine Learning Methods for Strategy Research

Mike Horia Teodorescu[1]

Harvard Business School and USPTO

Abstract

Numerous applications of machine learning have gained acceptance in the field of strategy and management research only during the last few years. Established uses span such diverse problems as strategic foreign investments, strategic resource allocation, systemic risk analysis, and customer relationship management. This survey article covers natural language processing methods focused on text analytics and machine learning methods with their applications to management research and strategic practice. The methods are presented accessibly, with directly applicable examples, supplemented by a rich set of references crossing multiple subfields of management science. The intended audience is the strategy and management researcher with an interest in understanding the concepts, the recently established applications, and the trends of machine learning for strategy research.

Keywords: strategic decisions, machine learning, natural language processing, classification, decision trees

[1]Address correspondence to Mike Teodorescu, Harvard Business School, Morgan Hall 276, Soldiers Field Road, Boston, MA 02165. E-mail: miketeod@hbs.edu. The author is also a part-time research affiliate of the US Patent and Trademark Office, Office of the Chief Economist, Alexandria, Virginia.

INTRODUCTION

During the last few decades, various management disciplines became hugely dependent on machine learning methods and tools. Domains such as marketing (Gans *et al.*, 2017; Struhl, 2015), financial markets (Tetlock, 2007; Tan *et al*., 2007; Bollen *et al.*, 2011), risk management (Hu *et al.*, 2012, Chen *et al.*, 2012), knowledge management (Williams and Lee, 2009; Li *et al.*, 2014; Balsmeier *et al.*, 2016), and logistics (Jordan and Mitchell, 2015), among others, are inconceivable today without the use of vast quantities of data and machine learning tools. Machine learning is the study of methods that make it possible to find patterns in data and the subsequent use of these patterns to construct predictions and inferences and to make decisions. The purpose of this article is to give a survey of machine learning methods and their applications to management, providing the reader with fundamental methodological tools via steps and examples that are accessible and easily reusable. The interested reader will also find targeted references to in-depth methodological content expanding the methods surveyed here and to a set of relevant articles in our management literature that showcase some of these methods. The examples are presented so as to be usable by a broad audience. Given text-based methods' growing use in our field and their partial independence of the other machine learning methods, the first half of the article presents and exemplifies textual analysis methods (part of the field of statistical natural language processing) such as term frequency, textual similarity, corpora considerations, and sentiment analysis (Manning and Schütze, 1999). The second part of the article covers general machine learning concepts, such as the concept of classification, the decision boundary, training and testing, cross-validation, and other fundamentals. It also exemplifies typical methods in machine learning that extend beyond text, such as decision

trees, random forests, k-Nearest-Neighbors, and Naïve Bayes. Each method is presented together with an implementation in an easy-to-use machine learning toolkit[2] that requires no programming background and with a current management literature example or a potential use in the management literature.

Insert Table 1 here

For a quick orientation to the main applications and trends of the methods of machine learning in solving important problems in strategy and management, Table 1 summarizes some of these problems and provides a few relevant references.

NATURAL LANGUAGE PROCESSING: TEXTUAL ANALYSIS

Management requires vast amounts of information that must be retrieved, aggregated, filtered, correlated, and analyzed from various standpoints. Many of the main sources of information come in textual form, such as corporate filings (e.g., Li, 2010a), financial disclosures (e.g., Loughran and McDonald, 2011, 2014), customer messages (e.g., Struhl, 2015; Ngai, Xiu, and Chau, 2009; Pang and Lee, 2008; Balazs and Velásquez, 2016; Mostafa, 2013; Piryani, Madhavi, and Singh, 2017; and Gans, Goldfarb, and Lederman, 2017), internal corporate documents such as corporate emails (e.g., Srivastava *et al*., 2017) and CEO diaries (e.g., Bandiera *et al.*, 2017), and patents (e.g., Hall, Jaffe, and Trajtenberg, 2001; Trajtenberg, Shiff, and Melamed, 2006; Kaplan, 2012; Li *et al.,* 2014; and Balsmeier *et al.*¸ 2016). The use of the information contained

[2] The mentions throughout this paper of various toolkits and software packages are not an endorsement of these toolkits and software packages. The opinions expressed are solely of the author, and are based on his experience with these toolkits, languages, and packages.

in text collections is based on methods pertaining to the domain of Natural Language Processing (NLP).

NLP is the interpretation of text and speech using automated analytical methods. Claude Shannon laid the groundwork for information theory and NLP by describing a model of communication (Shannon, 1948) and introducing statistical language models (Shannon, 1951); Alan Turing laid the foundation for artificial intelligence (Turing, 1950). Later efforts by others developed linguistic corpora for statistical analysis and grammar models for parsers in the 1960s and for question-answering systems in the 1970s and 1980s. A non-exhaustive list of subfields of NLP includes language parsers and grammars, text and speech recognition, sentiment analysis (including its impacts on firm and individual behavior), document classification (including insurance fraud detection, spam detection, and news manipulation), analysis of customers' and investors' sentiment tendencies (Lugmayr, 2013), search query disambiguation (for example, handling of word associations, abbreviations and polysemy), market segmentation, customer churn modeling, and many more. Researchers in management, strategy, marketing, and accounting have all found applications of NLP relevant to understanding consumer, firm, government, and individual executive behavior.

Text analysis workflow

Text is unstructured data that requires a sequence of processing stages to be quantified into variables which can then be used in regressions or classifications. A typical workflow, including the sampling and analysis step, is depicted in Figure 1. The first step of any textual analysis is to determine the sample of interest, which is generally referred to as a collection of documents,

where a document refers to an observation. A document or observation can be as short as a tweet or as long as a financial report or comprehensive patent description. The computational complexities of processing text are driven by the data volume, as measured by the size of the collection of documents and the average length of a document in the collection. The analysis of documents requires the comparison of their features with those of corpuses, which are comprehensive bodies of text representing a field or a natural language.

'Insert Figure 1 here'

The text preprocessing steps consist of tokenization, lemmatization or stemming, and stop words removal. Tokenization means segmenting a text, which is essentially a string of symbols including letters, spaces, punctuation marks, and numbers, into words and phrases. For example, a good tokenizer transforms "he's" into "he is" and "Dr." into "doctor," treats expressions such as "business model" as a single token, and processes hyphenation (Manning, Raghavan, Schütze, 2008). Depending on the purpose of the analysis, the tokenization may remove punctuation.

The other two preprocessing steps in text analysis are used depending on the purpose of the analysis. For example, when one wishes to differentiate between the specific languages used by two authors, one may wish to determine how frequently they use common words such as "the," "and," "that." These words are called "stop words" and serve grammatical purposes only, but their frequency and distribution may help "fingerprint" an author. In contrast, when one is interested in sentiment analysis, words that carry semantic meaning matter; stop words are generally held not to carry semantic meaning, so for such analyses they should be removed

in preprocessing. This is easily achieved with any of the standard text processing packages, which maintain dictionaries of stop words. Lemmatization, the reduction of the words to their lemma (the dictionary form of the word), helps lessen both the computational task and the duration of the analysis. Lemmatization reduces the number of words by mapping all inflections and variations of a word to the same lemma. It also disambiguates the semantic meaning of the words in a text by assigning words with the same meaning to their lemma. In sentiment analysis, for example, "improve," "improved," "improvement," and "improves" all point equally to an optimistic sentiment and share the same root; differentiating them would serve no purpose for a sentiment analysis task. The lemmatizer does distinguish between different parts of speech and notes whether the word is used as a verb or a noun, as these uses resolve to two different lemmas (as one would find in any standard dictionary). For instance, "binding contract," "being in a bind," and "bind together" would resolve to distinct lemmas, although they all use forms of "bind." A typical lemmatizer is the WordNet lemmatizer; several other stemmers and lemmatizers are described in Manning *et al.* (2008).

In other cases, information on the part of speech is not relevant for the analysis, and a simple removal of the prefixes and suffixes to reach the *stem* of the word is sufficient. The stem is the root of the word, the smallest unit of text that conveys the shared semantic meaning for the word family. For example, the stem of "teaching" is "teach." Because stemmers do not look up meaning in the context of parts of speech, verbs and nouns resolve to the same root, which reduces complexity but at the cost of a loss of information. A stemmer reduces the size of the text analysis problem by reducing the text's distinct words to a dictionary of roots, a minimally-sized collection, which enables the fastest analysis. Stemmers are standard in any programming

language or toolkit that enables text analysis. The standard stemmer (Manning and Schütze, 1999) for English language texts is the Porter Stemmer (Porter, 1980). Stemming may mask valuable information. For example, the Porter Stemmer produces on the corpus of patent titles the token "autom," which when applied to the standard American English corpus used in the literature, the Brown corpus (Kučera and Francis, 1967), finds that the stem corresponds to "automobile," whereas the expected word is "automate." It also introduces ambiguity when both "automaton" and "automate" stem to "autom."

While there is no generalized rule in the literature about where to use a stemmer versus a lemmatizer, all text preprocessing workflows should include at least one of the two. In the case of well-structured texts with limited lexicons, stemming is sufficient. However, for complex technical texts, such as patents, lemmatization is recommended. Further background in grammars, lemmatizers, stemmers, and text processing in general can be found in the comprehensive and widely cited textbook by Manning and Schütze (1999) and in Pustejovsky and Stubbs (2012).

Vector space model

All of these preprocessing steps allow us to prepare a document for consumption by a variety of numerical methods. The standard representation of a document is called the "vector space model," as each distinct word in the document becomes a feature of the document; the text can then be represented as a vector of words, with each word assigned a value. If the collection of documents is represented in an N-dimensional space, where *N* is the total number of distinct words across the collection (its vocabulary *V*), then each individual document is represented as

a point within this *N* dimensional space. Each dimension (axis in the corresponding diagram) represents a different word from the vocabulary of this collection. The numerical values on these axes for each document may be calculated in different ways. There are four typical methods:

1. Binary weighting at the document level assigns a value of 1 for the word's presence in the document and 0 for the word's absence in the document. This is useful in document classification tasks, where the presence or absence of a term is what matters in assigning the document to a particular topic (Albright, Cox, and Daly, 2001).
2. *Raw Term Frequency* is the raw count of the word in the document, and does not look at the total number of words in the document or at the collection of documents. It is useful in applications of sentiment analysis, where counts of positive and negative words are taken to determine the overall sentiment of the text. The most widely used annotated dictionaries include SENTIWORDNET[3] (Baccianella, Esuli, and Sebastiani, 2010) and the University of Illinois at Chicago's Opinion Lexicon[4] (Hu and Liu, 2004).
3. The *Relative Term Frequency* (TF) is calculated as the ratio between the number of occurrences of a word in a document and the number of times the word appears in the entire collection of documents. The tokenizer preprocessing step is essential for creating the proper list of words for each document, as it removes punctuation and non-word text. Stop words could "drown" out other words that would carry more meaning for the texts

[3] The SENTIWORDNET annotated corpus for sentiment analysis research is available at http://sentiwordnet.isti.cnr.it/. Accessed May 28th, 2017.
[4] The Opinion Lexicon consists of 6800 English words annotated with positive and negative sentiment and is freely available at the University of Chicago's website: https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html. Accessed May 28th, 2017.

under analysis, and so are removed prior to calculating TF. Inflections of a word would artificially lower the TF, which makes lemmatization/stemming critical. Importantly / essentially, the TF measure does not account for words that are common across documents.

4. Using the TF calculated at 3, one can create a separate set of weights called *Term Frequency-Inverse Document Frequency* (TF-IDF) that takes into account the number of documents in which the word appears through a separate measure called Inverse Document Frequency (IDF). Denoting the number of documents in the collection as *D* and the number of documents containing the i^{th} word in the alphabetically ordered vocabulary vector as D_i, the IDF is $IDF[i] = \log_2(D/D_i)$. From this definition, it is apparent that words that are common to *all* documents would lead to an IDF of 0. The TF-IDF is thus defined for each word *i* in document D_i as $TFIDF[D_i, i] = TF[D_i, i] \cdot IDF[i]$. The effect of multiplying the term frequencies for each word in each document by the inverse document frequency of that word is that words that are common across documents are weighted down, as they receive a low IDF value. However, uncommon terms that reveal specifics about a document, such as the methodological and technical terms that make a particular document unique, are weighted up by multiplication by the IDF. This is particularly useful when determining the extent of the difference between pairs of documents and is the standard method used in the NLP literature. For virtually any text analysis application that targets the unique or rare features in a document TF-IDF is the method of choice. For instance, patents use a highly specialized language in which common words are generally irrelevant. Younge and Kuhn (2015) performed TF-IDF on the entire patent corpus and determined the differences

across patents using cosine similarity on the word vectors associated with each patent. Another application of TF-IDF in management is the comparison of corporate financial forms such as 10-Ks and 10-Qs (Li, 2010b), where words common to most firms or forms are not particularly useful for extracting features of the firm's strategy. For a comprehensive review of term-weighting methods, see Salton and Buckley (1988).

Textual similarity measures

The most common similarity measures used in text analysis are the cosine similarity, the Pearson correlation, the Jaccard similarity, and the Dice similarity. All four rely on the vector representation of texts and produce a coefficient that compares pairs of texts. Cosine similarity has been used to compare texts for the past 30 years (Salton and Buckley, 1988; Salton, 1991), (Manning and Schütze, 1999). The cosine similarity is computed as the cosine of the angle of the pair of word vectors representing the two texts, denoted as $\overrightarrow{w_1}$ and $\overrightarrow{w_2}$. The components of these vectors are usually word counts (Manning and Schütze, 1999, p. 301):

$$\cos(\overrightarrow{w_1}, \overrightarrow{w_2}) = \frac{\sum_i w_{1i} \cdot w_{2i}}{\sqrt{\sum_i w_{1i}^2} \cdot \sqrt{\sum_i w_{2i}^2}}$$

The cosine similarity defined above may use TF or TF-IDF as a weighting method to create the values in each vector (see Salton and Buckley (1988) for an extensive review). Unlike cosine similarity and the Pearson correlation coefficient, the Jaccard and Dice similarity indices require binary weighting for both vectors (1 for the presence of a word and 0 for the absence of the word), thus acting at the set-level of the vocabularies *W* of the texts. The Jaccard similarity

measures the number of shared components between the two sets of words, and is defined (using set notation) as:

$$Jaccard(W_1, W_2) = \frac{|W_1 \cap W_2|}{|W_1 \cup W_2|},$$

where W_1 , W_2 are the vocabularies for the two texts. Dice similarity is defined likewise, with the key difference that it rewards shared word pairs while simultaneously penalizing pairs of texts that share fewer pairs of words relative to the total text sizes (Manning and Schütze, 1999, p. 299):

$$Dice(W_1, W_2) = \frac{2 \cdot |W_1 \cap W_2|}{|W_1| + |W_2|}.$$

Both Jaccard and Dice indices are used in information retrieval tasks, such as classification of documents and querying (Willett 1988). Overviews of these and other typical measures are in Manning and Schütze (1999, pp. 294-307), Salton and Buckley (1988), and Huang (2008). A survey of these measures applied to collections of short texts, such as online reviews and tweets, is in Metzler, Dumais, and Meek (2007).

Similarity measures were key to Hoberg and Phillips's (2010) study showing that firms with products very similar in textual descriptions to those of their rivals have lower profitability, and to Younge and Kuhn's (2015) study of how patent text similarities can predict future innovation. Arts, Cassiman, and Gomez (2017) apply Jaccard similarity to the patent corpus to determine technological similarity classes and compare their classification system to the USPC patent classification system. Textual similarity measures can also be helpful in creating comparison groups and identifying new classification structures. For example, they can help find groups of

companies with similar stated strategies through analysis of their financial disclosures, companies that create comparable products despite being in different SIC codes (Hoberg and Phillips, 2010), companies with similar customer review sentiments, or companies that have received comparable news coverage. These comparison groups can then be used in regression analysis.

MACHINE LEARNING TOOLS AND PROGRAMMING LANGUAGES

Introducing toolkits to the management literature helps reduce the barriers to entry for research questions that require machine learning, web mining, or text processing, and may benefit our field by reducing the costs of collecting and analyzing data. In the category of toolkits for data mining and processing, there are several, such as WEKA, RapidMiner, and KNIME, that are convenient in both teaching and research, due to the ease of use and fast learning curve. Several, e.g. RapidMiner, KNIME, enable users to run full machine learning algorithms applying just a drag-and-drop interface, while also providing suggestions for the best parameters for the algorithms. In RapidMiner, the recommendations are based on the inputted data and also on input from a cloud-based platform to which the software is connected, which compares the performance of various algorithms on millions of datasets. While regular programming languages that support machine learning packages, such as Python, C#, and R, provide more functionality, they are not as intuitive and have a steeper learning curve than ready-to-use packages. Data collection from the Internet is automated in some packages, a further advantage for the management researcher. For example, RapidMiner is able to extract

Twitter data with a single prebuilt operator and to extract data from a custom website using another built-in operator called Process Documents from Web. In this section, I also provide an overview of two natural language processing packages useful to management researchers working with text: NLTK and AYLIEN, a cloud-based toolkit. Under languages supporting machine learning, I briefly survey Python, C#, and Java. A review of these and other languages and tools is available in Louridas and Ebert (2017).

For non-programmers, general purpose machine learning toolkits such as SAS, WEKA (the open source software Waikato Environment for Knowledge Analysis), and RapidMiner may be suitable. Some of these toolkits have built-in cloud computing capabilities, hundreds of available algorithms, and a simple drag-and-drop visual interface for programming. A programming task in these tools may reduce to selecting a sequence of prebuilt operators to create a sequence of linked operators that forms a process. Figure 2 depicts a process that computes the Term Frequency vectors for a collection of text documents in RapidMiner. The input for the collection of documents is specified by an operator from the Data Access list. The Select Attributes operator allows selections for the columns to be used as input variables to the text processing algorithm. The actual document processing occurs in the Process Documents operator, which can take a wide variety of inputs, for example from a collection of files, from Twitter, or from a custom website.

'Insert Figure 2 here'

Most toolkits include naïve-Bayes, tree-based algorithms, nearest neighbor, and support vector machine algorithms (discussed in the general-purpose machine learning section of the paper),

neural networks, and others. Toolkits also provide standard statistical models and methods, including a suite of regression, segmentation, and correlation operators. Toolkits offer a wide variety of web mining tools (Kotu and Deshpande, 2014), including tools that gather data from any website given search parameters, gather data from websites with authentication, gather data from Twitter, and collect emails from an email server. The latter two have proven especially useful data sources for recent strategy research. For instance, Gans *et al.* (2017) analyzed sentiment in customer tweets to predict firm behavior. Srivastava *et al*. (2017) applied a tree-based machine learning approach to a firm's email server and applied a tree-based approach to determine how well employees matched the firm's email culture, and how differences in culture may impact employee turnover. The methods in these two papers could be implemented in current toolkits such as RapidMiner with just a few dragged-and-dropped operators, without the need to learn a programming language.

A unique feature of some toolkits compared to programming languages with support for machine learning is the ability to incorporate into the algorithms previous successful experience and knowledge from other sources. The use of the "wisdom of crowds" has been applied in many fields, such as biology, medicine, and NLP (Savage, 2012). A "wisdom of crowds" cloud engine (see the lower part of Figure 2, RapidMiner implementation) is a useful complement; it provides suggestions for parameter values for the operators as well as a sequence of operators that construct a program to analyze the inputted data.

The ability to visualize data and results is built into many tools, such as Tableau, Qlik, SAS, MATLAB, and RapidMiner. However, RapidMiner is more limited in its visualization capabilities

than visualization tools such as Tableau and Qlik or visualization packages such as D3 or Python's Matplotlib.

MATLAB is a programming language and software package focused on manipulating matrices (a matrix computation language).It provides functions and applications for a large variety of fields. It has numerous toolboxes for statistical analysis (for example, the "Statistics and Machine Learning Toolbox"), predictive analysis, and natural language processing. It even has a Financial Toolbox that complements its more engineering-rooted toolboxes, such as the Neural Network Toolbox, Fuzzy Logic Toolbox, Signal Analysis Toolbox, and Time Series Toolbox. This software package also has a toolbox for easily interfacing with relational databases. Applications for credit scoring and trading are readily available in the Risk Management Toolbox. While it is not the easiest to learn and manipulate, MATLAB is often preferred because of the ease and efficiency of using multidimensional matrix data. Good overviews of MATLAB for finance and economics include Anderson (2004) and Brandimarte (2006).

Statistical languages such as R provide machine learning packages, but their implementation time is not as fast as that of toolkits. R requires individual packages for different algorithms, as each package is relatively limited in scope (packages are available at CRAN). For example, "rpart" is used for basic classification algorithms, but ensemble methods require additional packages, like "party" or "randomforest." Other packages are built around specific algorithms, such as neural networks in "nnet" and kernel-based machine learning models in "kernLab." Generally, these require a bit more research and learning than the prebuilt packages in MATLAB, RapidMiner, or SAS. Two good resources for working with machine learning

algorithms in R are Friedman, Jastie, and Tibshirani (2001) and the associated datasets and packages and the UC Irvine Machine Learning Repository (Lichman, 2013).

SAS makes possible the statistical data analysis, data management, and visualization that are widely used in business intelligence. It claims a more accessible interface than R, with targeted packages for specific fields. Such specialized packages are not free but provide a wide array of tools, such as the case of Enterprise Miner, which provides a comprehensive set of machine learning tools, overviewed in Hall *et al.* (2014), the closest equivalent in terms of functionality to the tools already discussed. Like RapidMiner and the freeware R, SAS has a free academic edition. The general-purpose programming languages Python, C#, and Java all have a variety of machine learning, text analysis, and web mining packages. For example, in Python, the typical packages covering machine learning functionality include NLTK for natural language processing, scikit-learn and pylearn2 for machine learning methods, beautifulsoup for web parsing, pandas for data parsing from files, and Matplotlib (MATLAB-like interface) and Seaborn for data visualization. For C#, a good library for machine learning is Accord.NET, and a good library for natural language processing is Stanford's CoreNLP. Machine learning package examples for Java include the user-friendly freeware Weka and Java-ML. Using a general-purpose language may unlock more customization than would be possible with a ready-made toolkit, though the researcher has to navigate implementation challenges such as higher setup costs, as general-purpose programming languages setup involves a fairly complex patchwork of optional packages.

In terms of packages specifically targeted to natural language processing, NLTK is a comprehensive text analysis platform for Python, whereas AYLIEN is a cross-language cloud

based text processing toolkit with advanced sentiment analysis, news parsing, and named entity extraction abilities. NLTK is better for corpus analytics, as it incorporates over 100 text corpora from different fields,[5] contains a lemmatizer based on WordNet, and has extensive functionality for sentence parsing based on grammars. For an exhaustive overview of NLTK capabilities and examples, see Bird, Klein, and Loper (2009). NLTK is used in the corpora and Zipf's law section of this paper.

For the management researcher interested in easily collecting data about firms and then analyzing the data for sentiment or for entity extraction (locations, individuals, company names, product names, currency amounts, emails, or telephone numbers) from news sites, Twitter, documents, or websites in general, AYLIEN is available as a text extension for RapidMiner and as a Python, Java, and C# package. The news and Twitter parsers allow the user to connect these entities to collections of text documents, which can then be linked to events like stock prices or product launches and assigned a sentiment value through the prebuilt sentiment analyzer.

Sentiment analysis and the Naïve-Bayes classifier using NLP

Investor sentiment is known to affect stock returns (Lee, Shleifer, and Thaler, 1991), and investors themselves are known to be influenced by the sentiment of news articles (Tetlock, 2007; Devitt and Ahmad, 2007), by the sentiment of conventional media (Yu, Duan, and Cao, 2013), by social media (Bollen, Mao, and Zeng, 2010), and by nuances of optimism about future events as reported in standard financial filings (Li, 2010b). Attitudes and sentiments are

[5] For a list of current linguistic corpora included with NLTK, see http://www.nltk.org/nltk_data/. Accessed May 29th 2018.

detected by counting "positive" and "negative" words and expressions, using specific "bags" (sets) of sentiment/opinion words in the lexicon-based detection methods (such as in Taboada *et al*. (2011) or Ravi and Ravi (2015)), and calculating sentiment scores as the ratios of these counts (Struhl, 2015). The method can be enhanced using text corpora and parts of speech (POS) recognition and tagging to determine the context of word use. The second class of methods for sentiment detection pertains to machine learning. Various types of supervised classifiers are used in the literature to mine for the sentiments in a text, such as neural networks (NN), support vector machines (SVM), rule-based (RB) systems, naïve Bayes (NB), maximum entropy (ME), and hybrids. Ravi and Ravi (2015) and Tsytsarau and Palpanas (2012) provide details on the classifiers and related machine learning techniques in opinion mining. N-grams, which are uninterrupted sequences of N tokens, are often used in sentiment analysis to classify the sentiment of expressions. In the cases of online data sources, tokens may include punctuation constructs in the form of emoticons, and n-gram analysis takes into account the affect that an emoticon carries, such as through the use of bi-grams (pairs of tokens) to analyze consumer behavior and sentiment with regards to actions in the airline industry (Gans *et al.*, 2017).

In most sentiment analysis applications, a classification decision must be made regarding the type of sentiment (positive, negative, or neutral) at the document level or the sentence level. The typical classifier used in this context (Li, 2010b; Gans *et al.*, 2017) is the naïve-Bayes, a fast, general-purpose classifier popular in sentiment analysis (e.g., Pang and Lee, 2004, 2008; Melville, Gryc, and Lawrence, 2009; Dinu and Iuga, 2012).

The naïve-Bayes classifier works by using Bayes' rule for each classification decision under the assumption that *all predictors* are independent of each other. The name of the classifier is drawn from this assumption, which yields a certain naiveté in many situations but also makes this the simplest classifier, with no parameters to tune. The lack of parameter tuning makes it one of the fastest classifiers, which is especially useful in problems in which real-time analysis is needed, such as stock trading and question-answering bots. The naïve-Bayes algorithm calculates the posterior probability for each class given a predictor and picks the class with the highest posterior probability as the outcome of the classification. A broad overview of more advanced machine learning methods that outperform naïve-Bayes is presented in the General Purpose Machine Learning Methods section.

Corpora and Zipf's law

Business applications such as marketing sentiment and shareholder analysis require large corpora composed of collections of messages and documents. A linguistic corpus is a "systematically collected" set of "machine-readable texts" "representative of 'standard' varieties of [a language]" (Leech, 1991, p. 10, (Pustejovsky and Stubbs, 2012, p. 8). A corpus should be a representative sample of the overall language, which may be a natural languages or a specialized language like those used in patents, individual scientific fields, financial reports, consumer reviews, or short online texts (Tweets, product descriptions, firm mission statements). A comprehensive list of the most used and freely available corpora is provided in the appendix of Pustejovski and Stubbs (2012); the library NLTK, discussed in the prior section, also provides a growing set of free linguistic corpora. It is widely accepted that the standard American English corpus is the Brown corpus (Kučera and Francis, 1967), which was created as

a representative sample of the English language by a group at Brown University in the 1960s. The Brown corpus features balanced coverage of the different genres of the time, including specialized fields (Manning and Schütze, 1999, p. 19), and covers about 1 million words. The optimal corpus for a given field is a corpus formed by collecting a complete population of the texts that define the field under study. In certain cases, such as patent texts, all texts are indeed available in machine-readable form, and it is feasible to collect the complete population of texts.

The selection of an appropriate corpus for the research setting is essential, as using a general-purpose corpus can lead to misleading results (Li, 2010a). Linguistic corpora are traditionally characterized through the following parameters: the size of the vocabulary *V* (total number of distinct words), the total number of words, and the total number of documents in the corpus. Corpora may be tagged and annotated to enhance their analytical usefulness. For example, words may be tagged with their part of speech to enable statistical analysis of the inherent grammar of the language, as in the case of Treebanks, whose gold standard is the Penn TreeBank (Marcus, Marcinkiewicz, and Santorini, 1993). Such tags can then be used as an input to allow a classification algorithm to learn to classify a wider body of text, as is the case of the Reuters corpus, which collected a balanced sample of news articles spanning 90 topics and classified each article (for an overview of Reuters corpus uses, see Sebastiani, 2002). For a methodological overview of the composition of a linguistic corpus, see Pustejovsky and Stubbs (2012), and for a programmatic reference for working with corpora, see Bird *et al.* (2009) and Marcus *et al.* (1993).

Zipf's law (Zipf, 1949) is a descriptive instrument and a benchmark in the analysis of large text collections. Zipf's law states that if one were to order the words of a natural language by their frequency of appearance in the language and name the ordered position of each word as its *rank*, the relationship between the frequency *f* and the rank *r* would be (Manning and Schütze, 1999, p. 24):

$$f \propto \frac{1}{r^{\alpha}}, \alpha > 1.$$

In a log-log plot of rank and frequency (a standard visualization for corpora), Zipf's law would yield a line with a slope of about -1. The English language follows this approximation, with the top of the distribution dominated by stop words. The top of the distribution is not particularly helpful for comparing texts or retrieving texts; thus, it is often dropped prior to extracting the word vector for similarity analysis or information retrieval. The lower end of the distribution is dominated by the most infrequent words, which are helpful for defining a text for searches and in similarity analysis. Zipf's Law is just an approximation of a language; the Brown corpus, for example, deviates at the tails of the distribution from the ideal (see Figure 3). Despite this limitation, the log-log frequency-rank construction remains helpful for comparing the distributions of linguistic corpora and for identifying differences between a natural corpus such as Brown and corpora that are field-specific. Such differences matter for accurately understanding sentiment in financial disclosures in the form of forward-looking statements, which do not follow typical sentiment dictionaries and require training a classifier on a specialized collection of documents (Li, 2010b). Similarly, in case of the language of online reviews, a specialized corpus must be created to properly classify sentiment (Gans *et al*., 2017).

In the patent corpus, specialized technical language yields a very different distribution from that of a natural language corpus such as Brown. Figure 3 shows the comparison between the standard Brown corpus, the US-granted patent abstracts corpus, the US-granted patent titles corpus, and the US-granted patent claims corpus (all patents issued between 2007 and November 2017). The heads (due to stop words) and the tails of the distributions differ in the Brown and the three patent corpora. For the titles corpus, for example, relatively technical words such as "method," "device," "system," "apparatus," and "control" appear at the top of the distribution. This suggests that for patent similarity analysis, the most frequent (and thus least informative) words are different from those in a regular English language corpus, and that the distributions of the corpora should be determined in advance. This also holds for other fields that use specialized language, such as product descriptions, news articles, marketing materials, financial disclosures, and company mission statements. Zipf's law is a regularity that also holds for city population sizes (Gabaix, 1999; Glaeser *et al.*, 1992), firm size by employee number (Axtell, 2001), and firm bankruptcies (Fujiwara, 2004).

'Insert Figure 3 here'

GENERAL-PURPOSE MACHINE LEARNING METHODS

Learning types: taught versus self-educated

The "learning" part in machine learning is that of an algorithm that tweaks the parameters of a model of data to reach a goal based on an optimization criterion. The goal varies according to the application—it could be winning a game of chess, predicting customer purchasing behavior with an accuracy above a set requirement, segmenting the data until a mathematical

minimization criterion is achieved, or reaching a certain population composition after evolving over thousands of generations.

In the machine learning context, learning falls into four main categories: unsupervised learning, supervised learning, reinforcement learning, and evolutionary learning, with the last two often included in the supervised learning. Machine learning uses a model that ties the outcome variable (which may be referred to as the *target*) to the explanatory variables (which are referred to as *parameters* in the model). In the case of *supervised learning*, the algorithm has knowledge of the values of the explanatory variables that lead to a given outcome in the existing data, or a "teacher" gives input that corrects erroneous decisions taken by the algorithm. For instance, in the case of a loan decision, a bank employee may override an algorithm decision. Direct human intervention in the form of teaching is frequently seen in artificial intelligence applications (image classification with the subset field of handwriting recognition, which is now ubiquitous in the postal and banking sectors; speech recognition; spam filtering), but is less relevant for management applications. In the case of social science research, one might have a small subset of the data manually coded (outcome variable known) and need the algorithm to code the remainder of the data (outcome unknown), or the data may already be coded in terms of the correct outcome variable values, yet the mechanism producing these outcomes may be unknown.

In the case of unsupervised learning, none of the data are tagged and there is no human intervention to help the algorithm adjust along the way. Unsupervised learning covers data segmentation problems familiar in statistics such as clustering (k-means is a typical method here) or principal component analysis (PCA). Clustering is popular in marketing, particularly in

consumer market segmentation questions (as in Punj and Stewart, 1983; Schaffer and Green, 1998; Wedel and Kamakura, 2012), and an approach similar to PCA has been used recently to determine components of CEO behavior (Bandiera *et al*., 2017). Both k-means and PCA are reviewed below.

Reinforcement learning rewards the machine if it reaches the correct outcome, but does not reward individual steps. It is a form of machine learning for multistage games, such as chess or checkers, in which no one move necessarily leads to the desired outcome, but there exist multiple winning paths. The machine becomes better after each completed game.

Evolutionary learning is inspired by biological mutation, selection, and reproduction in populations. The genes in a population act as the parameters of a fitness function that evaluates the quality of a solution given a set of population characteristics. Genes that improve solution quality compared to the previous iteration are selected and promoted through to the next generation, while mutation is the mechanism used to prevent the algorithm from being trapped in a local optimum. The drawbacks of evolutionary algorithms include the high mathematical complexity that makes computation time a factor to consider and the fact that the solutions may not necessarily be as intuitive or easily interpretable as those of other methods. Their applications include financial trading (Allen and Karjalainen, 1999).

K-means: an example of unsupervised learning

K-means is a typical case of unsupervised learning and thus of knowledge discovery. The method requires as an input a good guess of the number of classes, k, which the researcher should make before running the algorithm. In this respect, it is a method of partial knowledge

discovery with partial supervision, as the number of clusters is "taught" to the machine. It is similar in principle to k-Nearest-Neighbors (also summarized in this paper) as it uses distances, yet it is based on cluster centers called centroids. Initially, the k clusters consist of a single element each; these elements are picked randomly from the data (if no good guess can be made about how to pick them) in the initialization step. At this stage, the selected data points become the "centers" of the newly formed clusters. In the next step of the most elementary form of the procedure, a new data point is randomly selected, the distances between the newly selected data point and the centers of the clusters are determined, and the new element is assigned to the closest cluster. The cluster now contains two data points, and its centroid is computed as the average coordinate values of the two members of the cluster. Subsequently, new data are selected randomly and the previous steps are repeated. Alternatively and more frequently, in the second step all data points are assigned to one of the clusters and contribute to the computation of the new centroids. During the run of the algorithm, the centers of the clusters continuously migrate, eventually tending toward fixed positions. Because the early-stage assignment of the data points to clusters may be wrong, the process is retested several times. When the centers of the clusters stabilize, the machine has learned the statistics of the data and the process is stopped. The k-means method is relatively simple algorithmically, but time-consuming.

The uses of k-means in the management literature have primarily been in the preparatory stages of analysis. In their study of the relationships between knowledge management strategies and organizational performance, Choi, Poon, and Davis (2008) used k-means to determine clusters of companies as a first step. Another example is Ngai *et al.* (2009), which

discussed k-means in the context of customer clustering, while Chiu *et al.* (2009) used the method for market segmentation. Wang (2009) performed a detailed analysis of various clustering methods in market segmentation based on published research and concluded that k-means is not the most robust technique, although it behaves reasonably. Wang suggested the use of hybrid kernel-based methods for customer relationship management applications, especially when the target clusters are overlapped and outliers are present.

Principal component analysis: unsupervised

Principal component analysis (PCA) is a method used in multivariate data analysis to sort out the input variables that play the most important role in explaining the variance of the results. It is used primarily when there are numerous input variables with varied degrees of importance in explaining a process, and when one suspects that some of these variables play no or little role. The technique is commonly used *before* machine learning methods are applied to reduce the dimensionality of the problem and accelerate computations by projecting a massive multidimensional space into a subspace of much lower dimension. The applications of PCA in management are numerous. For example, Van de Vrande *et al.* (2009) used PCA "to reduce the number of dimensions in data and applied cluster analytic techniques to find homogeneous groups of enterprises." Elenkov, Judge, and Wright (2005) recalled two other applications of PCA, the first in developing "measures for Product-Market Innovation and Administrative Innovation" and the second providing "support to the typology of product-market and administrative innovations." Lamberg *et al*. (2009) "employed PCA to illustrate the movement of … organizations in the competitive landscape." PCA has also been used to analyze relationships between corporate vision statements and employee satisfaction (Slack *et al*.,

2010), extract factors from survey questions in investment exit decisions (Elfenbein *et al*., 2016), analyze board decisions regarding CEO compensation (Zhu, 2013), and create board independence measures (Lange *et al.*, 2014). Further, virtually all clustering and regression studies that work with large numbers of variables need to apply PCA before the core analysis is performed.

Training, testing, cross-validation in supervised learning: core concepts

A supervised algorithm requires a pre-tagged dataset in which the correct outcome for each data point has already been made available. This pre-tagged dataset is called the training set, as it is used by the algorithm to learn and update its parameters. The second dataset is called the test set, and is used for validating the model determined during the training portion. The training and testing sets have to be mutually exclusive— the training data may not be reused for testing, as the model will simply fit the training data and perform poorly on new data. Both the training and the testing data must include the outcome variable. The outcome variable may either be a continuous variable, as in the case of regression, or a discrete variable (mutually exclusive categories) in the case of classification. The available data are typically divided between training and testing at a split of 80:20 or 90:10. Running a learning algorithm once does not eliminate the possibility that the model may avoid generalizing to new data and instead merely overfit the training data. To solve this problem, the data are randomly sampled into different subsets and the model is run multiple times on different splits of the data, a method called cross-validation.

Cross-validation involves measuring algorithm accuracy and comparing different runs of the algorithm across the various splits of the data to optimize model parameters. Multiple flavors of cross-validation are standard in any machine learning toolkit, including holdout cross validation, k-fold cross-validation, and leave-some-out multi-fold cross-validation.

The k-fold cross-validation approach is the typical one in most applications (Manning and Schütze, 1999). In this method, the data are randomly partitioned into k mutually-exclusive subsets and the algorithm is run k times, with each run on a different set of k-1 subsets joined as a training set and with testing done on the remaining subset. The k runs thus produce k different parameter sets for the algorithm, and the classification performances of these runs can be compared to each other. K is normally selected to be at least ten folds. An example is illustrated in Figure 4.

'Insert Figure 4 here'

Out of the k runs, the best-performing algorithm is selected as the outcome of the cross-validation process. The performance of a machine learning algorithm, however, involves more than just accuracy. Measuring it requires a few more concepts, which I discuss in the following section.

Classification and accuracy measures

The simplest measure is *classification accuracy*, defined as the sum of the true positives and true negatives divided by the total number of classification decisions (sum of the true positives, true negatives, false positives, and false negatives.) A frequently used measure is *precision*, which is defined as the true positives divided by the sum of the true positives and false

positives. *Recall* is defined as the true positives divided by the sum of the true positives and false negatives. The *Confusion Matrix* is a simple representation of the four standard accuracy metrics by which to measure the performance of a classifier, as in Figure 5.

'Insert Figure 5 here'

The typical method used to compare different runs of the same classifier (e.g., in cross-validation) or different classifiers run on the same data is the Receiver Operating Characteristic (ROC) curve, a plot of the true positive rate versus the true negative rate. It is also applied to compare the performance of classifiers against chance, which in an ROC plot is the diagonal. The area under the ROC curve is one of the most frequently encountered measures of classifier performance, and is called the *Area Under the Curve* (AUC). The best possible model would have an AUC of 1, as it would be a perfect detector of true positives and return no false positives.

K-Nearest-Neighbors: a simple supervised learning method

The k-Nearest-Neighbors (kNN) algorithm is one of the simplest of the supervised learning methods. It uses a distance metric such as Euclidean, city-block, Mahalanobis, or Chebyshev to classify each data point computing the distance between the data point with unknown class and each data point in the training set. The closest k training data are chosen, and the majority vote across these k wins and is assigned as the classification for the new data point. The choice of k matters, as values that are too small, such as k=1, overfit, whereas large values of k (k=21, for example) take considerable computational time. Using cross-validation, one can easily find the value of k with the lowest misclassification error. Unlike the simplest classifier, naïve-Bayes,

the kNN does not assume independence between the predictors and can take on any decision boundary shape (the decision boundary separates distinct classes in the predictor space). If speed of prediction as measured by time to classify new data matters for the application, kNN is a worse choice than naïve-Bayes, as it is slow at prediction, which matters in financial market applications and other applications where near-real-time responses are needed. Although kNN is biased, corrections can be applied (Magnussen *et al.*, 2010).

Decision trees: supervised learning

A tree is a graph with a node designated as a *root*, any two nodes of which are connected by only one path. All nodes except the terminal nodes, called *leaves,* branch out. Each branching represents a decision. A typical type of decision tree is the binary tree, where each decision yields exactly two choices. Trees with more than two branches per node are used as well, though this discussion centers on the binary classification tree. The decision tree can be used to classify any kind of categorical outcome. Key benefits of decision trees compared to other methods are that they are intuitive and can be used to generate an *induction rule set*, the set of mutually exclusive classification rules that yield every possible outcome in the tree. These rules are easy to interpret, as they are simple if-then/else-then rules that can also be constructed manually just by tracing the path from the root of the tree to each leaf, keeping every node split as a condition. In Figure 6, I exemplify a few decision paths for an organization looking at an acquisition. This is a simplified example of the decision paths a company might take in acquiring another: some of the decisions might involve binary variables (e.g., is the target company profitable or not?) or numerical variables (e.g., debt load). Such a combination poses

no difficulties for a decision tree model, which splits the feature space of the data based on the individual variables.

'Insert Figure 6 here'

Each path through the tree leads to a *leaf node* that allots all data satisfying that decision path to a particular class. In the crude example from Figure 6, there are two decisions, to acquire or not acquire, and a set of variables, some categorical and one numerical, that have different orders of importance depending on the path. Leaf nodes *can occur at any depth in the tree*, as some decision paths may be longer than others. On each path, the nodes closer to the root are of greater importance. If the leaf nodes are *pure*, the elements found in each of these leaf nodes are homogenous in their characteristics. To understand a standard decision tree algorithm, it is helpful to introduce the notion of *entropy* as the minimum amount of information necessary to transmit all possible outcomes in a random variable X (in other words, the minimum size of a binary message):

$$H(X) = -\sum_{x \in X} p(x) \cdot \log_2 p(x)$$

The entropy for a binary classification with two classes x_1 and x_2 is

$$H(X) = -p(x_1) \cdot \log_2 p(x_1) - p(x_2) \cdot \log_2 p(x_2)$$

with a maximum entropy for $p(x_1) = p(x_2) = 0.5$ corresponding to a random toss. The amount of entropy increases with the number of possible classes. The splitting of nodes is decided by the maximal reduction in the entropy, i.e., for which the largest information gain (IG) is achieved:

$$IG(X, A) = H(X) - H(X, A).$$

For a complete derivation, see Chapter 16 of Manning and Schütze (1999).

The process of node splitting is iterative: at each step the algorithm decides whether to split on a new attribute based on whether the entropy post-split is smaller than the entropy pre-split. This is the principle of the simplest decision tree algorithm, the ID3 iterative algorithm (Quinlan, 1979). There are numerous variations of decision tree algorithms (Marsland, 2015).

The depth of the tree is learned from the data. In the absence of a stop criterion, however, the tree could generate enough splits to perfectly model the input test data, thus resulting in an overfit model. *Pruning* is a method by which the depth of the tree is limited to avoid overfitting. Pruning involves setting a stop criterion that, once reached, will prevent further splits. This can mean, for example, that the trees stop splitting once they reach a certain depth (number of splits in the longest path), once the purity gained from an additional split falls below a preset threshold, or when the number of datapoints per node falls below a pre-set number. As described earlier, cross-validation is a typical approach to prevent overfitting (Elith, Leathwick, and Hastie, 2008).

Decision trees have been used for survival analysis as an alternative to logistic regression; a typical example is the Titanic survival data. For a comprehensive overview of this example and a comparison of logistic regression to decision trees, see Varian (2014). Recent work has also shown that decision trees can be used as an alternative to propensity score matching, as in Westereich, Lessler, and Funk (2010).

Decision trees and their resulting if-then *rule sets* can help both to understand processes in the data and to design new studies. For instance, past legal decisions may be modeled using decision trees, and the results applied to predict how firms may respond to a legal decision or use litigation as a tool in competitive behavior. The newly-published PACER patent litigation data set may be amenable to such analysis (Marco, Tesfayeus, and Toole, 2017). Decision trees can be employed to classify decisions in corporate documents if combined with NLP tools to extract variables from emails, memos, and financial filings. The lessons learned from such analyses may help organizational behaviorists design surveys based on features and decision paths extracted from data.

Forests, bagging, boosting: supervised learning

Some machine learning models are dependent on changes in the initial conditions in their data or their input parameters. Ensemble learning methods aggregate many runs of these models to generate a more generalizable model. One of the simplest such approaches is the *bagging* method, which combines bootstrapping with aggregation. Essentially, many different runs using different training data for each run through bootstrapping are aggregated through a majority voting system (or other criterion) to generate a new model based on these aggregated parameters. *Boosting* assigns higher weights to misclassified data, such that subsequent runs of the algorithm sample more of the misclassified data points and thus focus on reducing these misclassifications (for an approach using boosted regression trees, see Elith *et al.*, 2008). The different runs are aggregated through voting. Bagging and boosting are general-purpose ensemble methods. The *random forest* technique, by contrast, focuses solely on decision trees, generating a number of pre-specified decision tree models, each with a randomly selected

number of attributes from the data. The number of attributes chosen is less than the dimensionality of the data. Each decision tree receives an identically sized but randomly sampled training set. Finally, for each data point, classification is determined as the majority vote of all the trees' decisions for that data point. This ensemble method often outperforms simple cross-validation for decision trees. Random forests do this, and are also highly resilient to outliers and noise in the data. An in-depth comparative discussion of these method and their variants applied to credit scoring is in (Wang et al, 2012).

Application issues and examples

Managers and strategists need to work closely with their teams to direct data processing in view of extracting the pertinent information and for understanding the limits of the methods and the degree of validity of the results. Consider the case of the strategists of a landing firm who wish to have a simplified procedure for the selection of the borrowers; it seems that it is enough that they instruct their analysis team to determine how to categorize borrowers. However, such a basic request would reveal a limited understanding of the clustering methods and even an unclear purpose of the strategists. The latter should be aware that the clustering procedures, such as k-means, require the specification of the number of classes; also, the analysts may wonder if the strategists consider some features of the borrowers more important than the others, that implying a weighting of the components of the feature vectors; further, the choice of the distance in the k-means algorithm may have reasons better appreciated by a cognizant strategist. Also, knowing the limits of the clustering procedures, the analysts may wonder if the strategists need a result with more classes, but with less accuracy of clustering, or prefer the result with the best accuracy (e.g., in the sense of the silhouette method). For a

fruitful collaboration with the analysts, strategists need to know the principles of the methods, their main variants, and their limits.

The potential and limits of ML tools are illustrated for the case of methods based on local similarity, including k-means, k-NN classification, and the k-NN predictor, in the Appendix. The data comes from a freely available resource for machine learning training data at the UC Irvine Machine Learning Repository[6] (UCI ML). The first example in the appendix involves the methods of decision tree and kNN discussed above, with implementations in RapidMiner and Matlab, and the source scripts and links to the data from UCI ML included as online attachments to this paper. The example shows the influence of the choice of the parameters of the kNN classifier on the results, including the choice of the distance and indirectly illustrates the effect of the local inhomogeneity of the data distribution. The second example is a text analysis of firm slogans to determine clusters of competitors. The examples are intended to be easily reproduced.

DISCUSSION AND CONCLUSIONS

Methods pertaining to natural language processing, decision trees, clustering and classification have become necessary instruments for strategy and management in domains such as multinational corporations, international commerce, financial markets, alliances and mergers, corporate governance, supply chain optimization, transport management, banking, and knowledge transfers. This article surveyed part of the field of machine learning for methods relevant to recently accepted practices in strategy and more broadly in management research.

[6] UC Irvine Machine Learning Repository, https://archive.ics.uci.edu/ml/datasets.html, Accessed July 5th 2017.

It addressed tools offered by the subfield of natural language processing, such as sentiment analysis, textual similarity, and vocabulary analysis. It discussed their applicability for revealing firm culture, firm interests, coverage, goals, management objectives, and methods, and for the management of relationships with customers, employees, and stakeholders.

Next, the paper focused on clustering, classification and decision methods and their relation to prediction and other decision tools as used in strategy and management. It provided an accessible theoretical overview of each method, listed research topics for which each method would be useful, gave examples of machine learning toolkits and of their use, and discussed the predictive power of various machine-based instruments. It also listed the theoretical and applied limits of the methods in strategic management and mentioned current and future research directions.

Today, the inability to use large data collections is a handicap for companies and for researchers alike. For firms, it translates into wrong decisions, misunderstandings of the market, competitors and customers, and lack of competitiveness. To maintain a competitive advantage, managing teams need the understanding of the tools and methods of machine learning suitable to generate from the row data information about the categories of customers and competitors, the patterns of their behavior in the market, their relationship and sentiments toward the firms, and the trends of the above. Today the competitive edge of a firm may have the roots in the amount of data that it can obtain and process, in the quality and depth of the data processing, and in the ability of the management teams and strategists to ask questions to and cooperate with the data analysts. In turn, a fruitful collaboration between them requires an understanding by the strategists of the methods and tools in machine learning.

This article may help researchers and managers alike to master some of the tools in machine learning, particularly in using natural language processing methods, decision trees, clustering, and classification methods. Further, the article aimed to show how to combine language processing with other methods and how to determine the accuracy of the results. The method survey part of the paper may also serve as an orientation in the current state of the machine learning use in strategy and management, complementing previous reviews of the rapidly evolving business intelligence domain.

I acknowledge that several types of ML methods, including neural networks, graph-based techniques, decision maps, and Bayesian networks, among others had to be omitted in this survey in order to attain a minimal depth in the discussion of the other methods, especially those based on NLP, classification, and clustering. Methods based on networks deserve an entirely separate treatment and may be followed up in a separate paper.

ACKNOWLEDGMENTS

The author thanks Editor Alfonso Gambardella for his feedback, which was essential in shaping this paper. The author is grateful to Shane Greenstein, Tarun Khanna, John Deighton, William Kerr, Andy Wu, Michael Toffel, Frank Nagle, Stephen Hansen, Andrew Toole, Asrat Tesfayesus, Raffaella Sadun, Neil Thompson, Debbie Teodorescu, Ingo Mierswa, Knut Makowski, Andrei Hagiu, Aaron Yoon, Yo-Jud Cheng, and Daniel Brown for valuable feedback and reference suggestions. The author is solely responsible for any errors or omissions. The author states that there are no conflicts of interest regarding the content or production of this work.

REFERENCES

Albright R, Cox J and Daly K. 2001. Skinning the cat: comparing alternative text mining algorithms for categorization. In *Proceedings of the 2nd Data Mining Conference of DiaMondSUG*, Chicago, IL. DM Paper (Vol. 113).

Allen F, Karjalainen R. 1999. Using genetic algorithms to find technical trading rules. *Journal of Financial Economics* **51**(2): 245–271.

Alpaydin E. 2014. *Introduction to Machine Learning*. MIT Press: Cambridge, MA.

Arts, S., Cassiman, B. and Gomez, J.C., 2017. Text matching to measure patent similarity. *Strategic Management Journal*: pre-print.

Axtell RL. 2001. Zipf distribution of US firm sizes. *Science* **293**(5536): 1818–1820.

Baccianella S, Esuli A,Sebastiani F. 2010. SentiWordNet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. *Proceedings of LREC 2010*: 2200–2204.

Balazs JA, Velásquez JD. 2016. Opinion mining and information fusion: a survey. *Information Fusion* **27**: 95–110.

Balsmeier B, Li GC, Chesebro T, Zang G, Fierro G, Johnson K, Kaulagi A, Lück S, O'Reagan D, Yeh B, Fleming L. 2016. Machine learning and natural language processing on the patent corpus: data, tools, and new measures. Working Paper, UC Berkeley Fung Institute, Berkeley, CA. Available at: http://funginstitute.berkeley.edu/wp-content/uploads/2016/11/Machine_learning_and_natural_language_processing_on_the_patent_corpus.pdf

Bandiera O, Hansen S, Prat A, Sadun R. 2016. CEO behavior and firm performance. HBS Working Paper 17-083, Harvard Business School, Boston, MA. Available at: https://dash.harvard.edu/handle/1/30838134.

Bettis R, Gambardella A, Helfat C, Mitchell W. 2014. Quantitative empirical analysis in strategic management. *Strategic Management Journal* **35**(7): 949–953.

Bird S, Klein E, Loper E. 2009. Learning to classify text. In *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*, Bird S, Klein E, Loper E. (eds). O'Reilly Media, Inc.: Sebastopol, CA; 221–259.

Bloomfield R. 2008. Discussion of "annual report readability, current earnings, and earnings persistence." *Journal of Accounting and Economics* **45**(2):248–252.

Bollen J, Mao H, Zeng X. 2011. Twitter mood predicts the stock market. *Journal of Computational Science* **2**(1): 1–8.

Brink H, Richards J, Fetherolf M. 2014. *Real-World Machine Learning*. Manning: Shelter Island, NY.

Cavusgil ST, Kiyak T, Yeniyurt S. 2004. Complementary approaches to preliminary foreign market opportunity assessment: Country clustering and country ranking. *Industrial Marketing Management* **33**(7): 607–617.

Chen H, Chiang RHL, Storey VC. 2012. Business intelligence and analytics: From big data to big impact. *MIS Quarterly*, **36** (4): 1165-1188.

Chiu CY, Chen YF, Kuo, IT, Ku, HC. 2009. An intelligent market segmentation system using k-means and particle swarm optimization. *Expert Systems with Applications* **36**(3):4558–4565.

Choi B, Poon SK, Davis JG. 2008. Effects of knowledge management strategy on organizational performance: a complementarity theory-based approach. *Omega* **36**(2):235–251.

Cubiles-De-La-Vega M-D, Blanco-Oliver A, Pino-Mejías R, Lara-Rubio J. 2013. Improving the management of microfinance institutions by using credit scoring models based on Statistical Learning techniques. *Expert Systems with Applications*, **40**(17): 6910–6917.

Debaere P, Lee H, Lee J. 2010. It matters where you go. Outward foreign direct investment and multinational employment growth at home. *Journal of Development Economics*. **91**(2): 301–309.

Devitt A, Ahmad K. 2007. Sentiment polarity identification in financial news: a cohesion-based approach. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics,* Vol. 7, Prague, Czech Republic, 1–8.

Dinu L, Iuga I. 2012. The Naïve-Bayes classifier in opinion mining: in search of the best feature set. In *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, Berlin, 556–567.

Eggers JP and Kaplan S, 2009. Cognition and renewal: Comparing CEO and organizational effects on incumbent adaptation to technical change. *Organization Science* **20**(2):461-477.

Elenkov DS, Judge W, Wright P. 2005. Strategic leadership and executive innovation influence: an international multi-cluster comparative study. *Strategic Management Journal* **26**(7): 665–682.

Elfenbein DW, Knott AM and Croson R. 2017. Equity stakes and exit: An experimental approach to decomposing exit delay. *Strategic Management Journal* **38**(2): 278-299.

Elith J, Leathwick JR, Hastie T. 2008. A working guide to boosted regression trees. *Journal of Animal Ecology* **77**(4):1365–2656.

Ertek G, Tapuku D, Arin I. 2013. Text mining with RapidMiner. In *RapidMiner: Data Mining Use Cases and Business Analytics Applications*, Hofmann M, Klinkenberg R. (eds). Chapman and Hall/CRC: Boca Raton, FL; 241–288.

Friedman J, Hastie T, Tibshirani R. 2001. *The Elements of Statistical Learning.* Springer: Berlin.

Fujiwara Y. 2004. Zipf law in firms bankruptcy. *Physica A: Statistical Mechanics and its Applications* **337**(1):219–230.

Gabaix X. 1999. Zipf's law for cities: an explanation. *The Quarterly Journal of Economics* **114**(3):739–767.

Gamache DL, McNamara G, Mannor MJ and Johnson RE, 2015. Motivated to acquire? The impact of CEO regulatory focus on firm acquisitions. *Academy of Management Journal* **58**(4):1261-1282.

Gans JS, Goldfarb A, Lederman M. 2017. Exit, tweets and loyalty. NBER Working Paper 23046, National Bureau of Economic Research, Cambridge MA. Available at: http://www.nber.org/papers/w23046.

Glaeser EL, Kallal HD, Scheinkman JA,Shleifer A. 1992. Growth in cities. *Journal of Political Economy* **100**(6): 1126–1152.

Gow ID, Kaplan SN, Larcker DF and Zakolyukina AA, 2016. CEO personality and firm policies. NBER Working Paper 22435). National Bureau of Economic Research. Available at: http://www.nber.org/papers/w22435.

Hall BH, Jaffe AB, Trajtenberg M. 2001. The NBER patent citation data file: lessons, insights and methodological tools. NBER Working Paper 8498, National Bureau of Economic Research, Cambridge, MA. Available at: http://www.nber.org/papers/w8498.

Hall P, Dean J, Kabul IK, Silva J. 2014. An overview of machine learning with SAS Enterprise Miner. In *Proceedings of the SAS Global Forum 2014 Conference.* Available at: https://support.sas.com/resources/papers/proceedings14/SAS313-2014.pdf.

Hoberg G, Phillips G. 2010. Product market synergies and competition in mergers and acquisitions: a text-based analysis. *Review of Financial Studies* **23**(10): 3773–3811.

Hu D, Zhao JL, Hua Z, Wong MCS. 2012. Network-based modeling and analysis of systemic risk in banking systems. *MIS Quarterly* **36**(4): 1269-1291.

Hu M, Liu B. 2004. Mining and summarizing customer reviews. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Seattle, WA, 168–177. Available at: http://dl.acm.org/citation.cfm?id=1014052&picked=prox.

Huang A. 2008. Similarity measures for text document clustering. In *Proceedings of the 6th New Zealand Computer Science Research Student Conference* (NZCSRSC2008), Christchurch, New Zealand, 49–56.

Jordan MI, Mitchell TM. 2015. Machine learning: trends, perspectives, and prospects. *Science* **349**(6245): 255–260.

Kanze D, Huang L, Conley MA, Higgins ET, 2017. We ask men to win & women not to lose: closing the gender gap in startup funding. *Academy of Management Journal*, preprint amj-2016.

Kaplan S. 2012. Identifying breakthroughs: using topic modeling to distinguish the cognitive from the economic. *Academy of Management Proceedings* **2012**(1).

Kotu V, Deshpande B. 2014. *Predictive Analytics and Data Mining: Concepts and Practice with Rapidminer*. Morgan Kaufmann: Waltham, MA.

Kučera H, Francis WN. 1967. *Computational Analysis of Present-day American English*. Brown University Press: Providence, RI.

Lamberg JA, Tikkanen H, Nokelainen T, Suur-Inkeroinen H. 2009. Competitive dynamics, strategic consistency, and organizational survival. *Strategic Management Journal* **30**(1): 45–60.

Lange D, Boivie S and Westphal JD, 2015. Predicting organizational identification at the CEO level. *Strategic Management Journal* **36**(8):1224-1244.

Lau RYK, Liao SSY, Wong KF, and Chiu DKW. 2012. Web 2.0 environmental scanning and adaptive decision support. *MIS Quarterly* **36**(4): 1239-1268.

Lazer D, Kennedy R, King G, Vespignani A. 2014. The parable of Google Flu: traps in big data analysis. *Science* **343**(6176): 1203–1205.

Lee C, Shleifer A, Thaler R. 1991. Investor sentiment and the closed-end fund puzzle. *The Journal of Finance* **46** (1): 75–109.

Lee PM and James EH, 2007. She'-e-os: gender effects and investor reactions to the announcements of top executive appointments. *Strategic Management Journal* **28**(3): 227-241.

Leech G. 1991. The state of the art in corpus linguistics. In *English Corpus Linguistics: Studies in Honour of Jan Svartvik*, Svartvik J, Aijmer K, Altenberg B (eds). Longman: London; 8–29.

Li F. 2008. Annual report readability, current earnings, and earnings persistence. *Journal of Accounting and Economics* **45**(2):221–247.

Li F. 2010a. Textual analysis of corporate disclosures: a survey of the literature. *Journal of Accounting Literature* **29**: 143–165.

Li F. 2010b. The information content of forward-looking statements in corporate filings—a naïve Bayesian machine learning approach. *Journal of Accounting Research* **48**(5): 1049–1102.

Li GC, Lai R, D'Amour A, Doolin DM, Sun Y, Torvik VI, Amy ZY, Fleming L. 2014. Disambiguation and co-authorship networks of the US patent inventor database (1975–2010). *Research Policy* **43**(6): 941–955.

Liab H, Sun J. 2011. Principal component case-based reasoning ensemble for business failure prediction. *Information & Management* **48**(6): 220-227.

Lichman M. 2013. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. University of California, School of Information and Computer Science: Irvine, CA.

Loughran T, McDonald B. 2011. When is a liability not a liability? textual analysis, dictionaries, and 10-Ks. *The Journal of Finance* **66**(1):35–65.

Loughran TIM, McDonald B. 2014. Measuring readability in financial disclosures. *The Journal of Finance* **69**(4): 1643–1671.

Loughran T, McDonald B. 2016. Textual analysis in accounting and finance: a survey. *Journal of Accounting Research* **54**(4):1187–1230.

Louridas P, Ebert C. 2017. Machine learning. *Computing Edge*, April 2017: 8–13.

Lugmayr A. 2013. Predicting the future of investor sentiment with social media in stock exchange investments: a basic framework for the DAX Performance Index. In *Handbook of Social Media Management*, Friedrichsen M, Mühl-Benninghaus W (eds). Springer Media Business and Innovation: Berlin, Heidelberg; 565–589.

Magnussen S, Tomppo E, McRoberts RE. 2010. A model-assisted k-nearest neighbour approach to remove extrapolation bias. *Scandinavian Journal of Forest Research* **25**(2): 74 — 184.

Manning CD, Schütze H. 1999. Topics in information retrieval. In *Foundations of Statistical Natural Language Processing*. MIT Press: Cambridge, MA; 539–554.

Manning CD, Raghavan P, Schütze H. 2008. *Introduction to Information Retrieval.* Cambridge University Press: Cambridge, U.K.

Marco AC, Tesfayesus A, Toole AA. 2017. Patent Litigation Data from US District Court Electronic Records (1963–2015). USPTO Economic Working Paper No. 2017-06. Available at SSRN: https://ssrn.com/abstract=2942295 or http://dx.doi.org/10.2139/ssrn.2942295.

Marcus MP, Marcinkiewicz, MA, Santorini B. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics* **19**(2):313–330.

Marsland S. 2015. Learning with trees. In *Machine Learning: An Algorithmic Perspective*. CRC Press: Boca Raton, FL; 249–266.

Matej M, Miroslav P. 2013. Medical data mining. In *RapidMiner: Data Mining Use Cases and Business Analytics Applications*, Hofmann M, Klinkenberg R (eds). Chapman and Hall/CRC: Boca Raton, FL; 241–288.

Melville P, Gryc W, Lawrence RD. 2009. Sentiment analysis of blogs by combining lexical knowledge with text classification. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Paris, France, 1275–1284.

Metzler D, Dumais S, Meek C. 2007. Similarity measures for short segments of text. In *Advances in Information Retrieval*, Amati G, Carpineto C, Romano G. (eds). ECIR 2007. Lecture Notes in Computer Science, vol 4425. Springer: Berlin, Heidelberg.

Mostafa MM. 2013. More than words: social networks' text mining for consumer brand sentiments. *Expert Systems with Applications* **40**(10): 4241–4251.

Nadkarni S and Barr PS, 2008. Environmental context, managerial cognition, and strategic action: an integrated view. *Strategic Management Journal* **29**(13):1395-1427.

Ngai EW, Xiu L, Chau DC. 2009. Application of data mining techniques in customer relationship management: a literature review and classification. *Expert Systems with Applications* **36**(2):2592–2602.

Nikolic N, Zarkic-Joksimovic N, Stojanovski D, Joksimovic I. 2013. The application of brute force logistic regression to corporate credit scoring models: Evidence from Serbian financial statements. *Expert Systems with Applications* **40**(15): 5932–5944.

Pang B, Lee L. 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, Barcelona, Spain, 271.

Pang B, Lee L. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval* **2**(1–2): 1–135.

Piryani R, Madhavi D, Singh VK. 2017. Analytical mapping of opinion mining and sentiment analysis research during 2000–2015. *Information Processing & Management* **53**(1): 122–150.

Porter MF. 1980. An algorithm for suffix stripping. *Program* **14**(3): 130–137.

Punj G, Stewart DW. 1983. Cluster analysis in marketing research: review and suggestions for application. *Journal of Marketing Research* **20**:134–148.

Pustejovsky J, Stubbs A. 2012. Corpus analytics. In *Natural Language Annotation for Machine Learning*, Pustejovsky J, Stubbs A (eds). O'Reilly Media, Inc.: Sebastopol, CA; 53–65.

Quinlan JR. 1979. Discovering rules by induction from large collections of examples. *Expert Systems in the Microelectronic Age*, Michie D (ed). Edinburgh University Press: Edinburgh, U.K.; 168–201.

Ravi K, Ravi V, 2015. A survey on opinion mining and sentiment analysis: tasks, approaches and applications. *Knowledge-Based Systems* **89:**14-46.

Roth K. 1992. International Configuration And Coordination Archetypes For Medium-Sized Firms In Global Industries. *Journal of International Business Studies* **23**(3): 533–549.

Salton G, Buckley C. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing &Management* **24**(5):513–523.

Salton G. 1991. Developments in automatic text retrieval. *Science* **253**(5023):974–980.

Savage N. 2012. Gaining wisdom from crowds. *Communications of the ACM* **55**(3): 13–15.

Schaffer, CM, Green, PE, 1998. Cluster-based market segmentation: some further comparisons of alternative approaches. *Journal of the Market Research Society* **40**(2):155-164.

Sebastiani F. 2002. Machine learning in automated text categorization. *ACM Computing Surveys* (CSUR) **34**(1): 1–47.

Shannon CE. 1948. A mathematical theory of communication. *Bell System Technical Journal* **27**:379–423.

Shannon CE. 1951. Prediction and entropy of printed English. *Bell System Technical Journal* **30**: 50–64.

Siegel, E., 2016. Predictive analytics: The power to predict who will click, buy, lie, or die (pp. 103-110). Hoboken (NJ): Wiley.

Singh J, Verbeke W, Rhoads GK. 1996. Do Organizational Practices Matter in Role Stress Processes? A Study of Direct and Moderating Effects for Marketing-Oriented Boundary Spanners. *Journal of Marketing*, **60**(3): 69-86.

Slack FJ, Orife JN, Anderson FP. 2010. Effects of commitment to corporate vision on employee satisfaction with their organization: an empirical study in the United States. *International Journal of Management* **27**(3): 421-436.

Sohn MH, You T, Lee SL, Lee H. 2003. Corporate strategies, environmental forces, and performance measures: a weighting decision support system using the k-nearest neighbor technique. *Expert Systems with Applications* **25**(3): 279-292.

Sohn SY, Kim JW. 2012. Decision tree-based technology credit scoring for start-up firms: Korean case. *Expert Systems with Applications*, **39**(4): 4007-4012.

Srivastava SB, Goldberg A, Manian VG, Potts, C. 2017. Enculturation trajectories: language, cultural adaptation, and individual outcomes in organizations. *Management Science.* (Articles in Advance, doi:10.1287/mnsc.2016.2671).

Stock GN, Greis NP, Kasarda JD. 2000. Enterprise logistics and supply chain structure: the role of fit. *Journal of Operations Management*, **18**(5): 531–547.

Struhl S. 2013. *Market Segmentation.* American Marketing Association Press: Chicago, IL.

Struhl S. 2015. In the mood for sentiment. In *Practical Text Analytics: Interpreting Text and Unstructured Data for Business Intelligence*. Kogan Page Publishers: London, U.K.; 120–143.

Taboada M, Brooke J, Tofiloski M, Voll K, Stede M. 2011. Lexicon-based methods for sentiment analysis. *Computational Linguistics* **37**(2):267–307.

Tam KY, Kiang MY. 1992. Managerial Applications of neural networks: the case of bank failure predictions. *Management Science* **38**(7):926 – 947.

Tan T Z, Quek C, Ng GS. 2007. Biological brain-inspired genetic complementary learning for stock market and bank failure prediction, *Computational Intelligence* **23**(2):236-261.

Tetlock PC. 2007. Giving content to investor sentiment: the role of media in the stock market. *The Journal of Finance* **62**(3):1139–1168.

Trajtenberg M, Shiff G, Melamed R. 2006. The “names game”: harnessing inventors’ patent data for economic research (No. w12479). National Bureau of Economic Research.

Tsai CF, Chen ML. 2010. Credit rating by hybrid machine learning techniques. *Applied Soft Computing* **10**(2): 374–380.

Tsytsarau M, Palpanas T. 2012. Survey on mining subjective data on the web. *Data Mining and Knowledge Discovery* **24**(3): 478–514.

Turing AM. 1950. Computing machinery and intelligence. *Mind* **59**: 433–460.

Van de Vrande V, De Jong JP, Vanhaverbeke W, De Rochemont M. 2009. Open innovation in SMEs: trends, motives and management challenges. *Technovation* **29**(6): 423–437.

Varian HR. 2014. Big data: new tricks for econometrics. *Journal of Economic Perspectives* **28**(2): 3–28.

Wang CH. 2009. Outlier identification and market segmentation using kernel-based clustering techniques. *Expert Systems with Applications* **36**(2): 3744–3750.

Wang G, Ma J, Huang L, Xu K. 2012. Two credit scoring models based on dual strategy ensemble trees. *Knowledge-Based Systems* **26**(2): 61-68.

Wedel M, Kamakura WA. 2012. *Market Segmentation: Conceptual and Methodological Foundations* (Vol. 8). Springer Science & Business Media: New York.

Westreich D, Lessler J, Funk MJ. 2010. Propensity score estimation: neural networks, support vector machines, decision trees (CART), and meta-classifiers as alternatives to logistic regression. *Journal of Clinical Epidemiology* **63**(8): 826–833.

Willett P. 1988. Recent trends in hierarchic document clustering: a critical review. *Information Processing & Management* **24**(5): 577–597.

Williams C, Lee SH. 2009. Resource allocations, knowledge network characteristics and entrepreneurial orientation of multinational corporations. *Research Policy* **38**(8):1376–1387.

Yeh, I.C., Lien, C.H., 2009. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, **36**(2):2473-2480.

Younge KA, Kuhn JM. 2015. Patent-to-patent similarity: a vector space model. Available at SSRN https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2709238.

Yu Y, Duan W, Cao Q. 2013. The impact of social and conventional media on firm equity value: a sentiment analysis approach. *Decision Support Systems* **55**(4):919–926.

Zhu DH, 2013. Group polarization in board decisions about CEO compensation. Organization Science **25**(2):552-571.

Zipf GK. 1949. *Human Behavior and the Principle of Least Effort.* Addison-Wesley*:* Cambridge, MA.

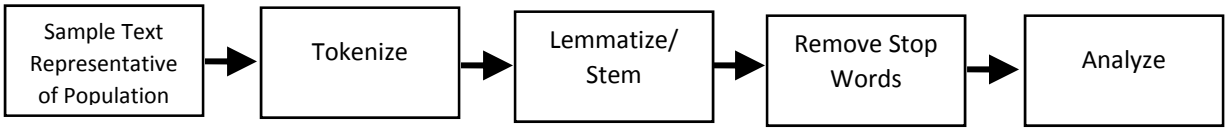

Figure 1. Typical workflow for processing text

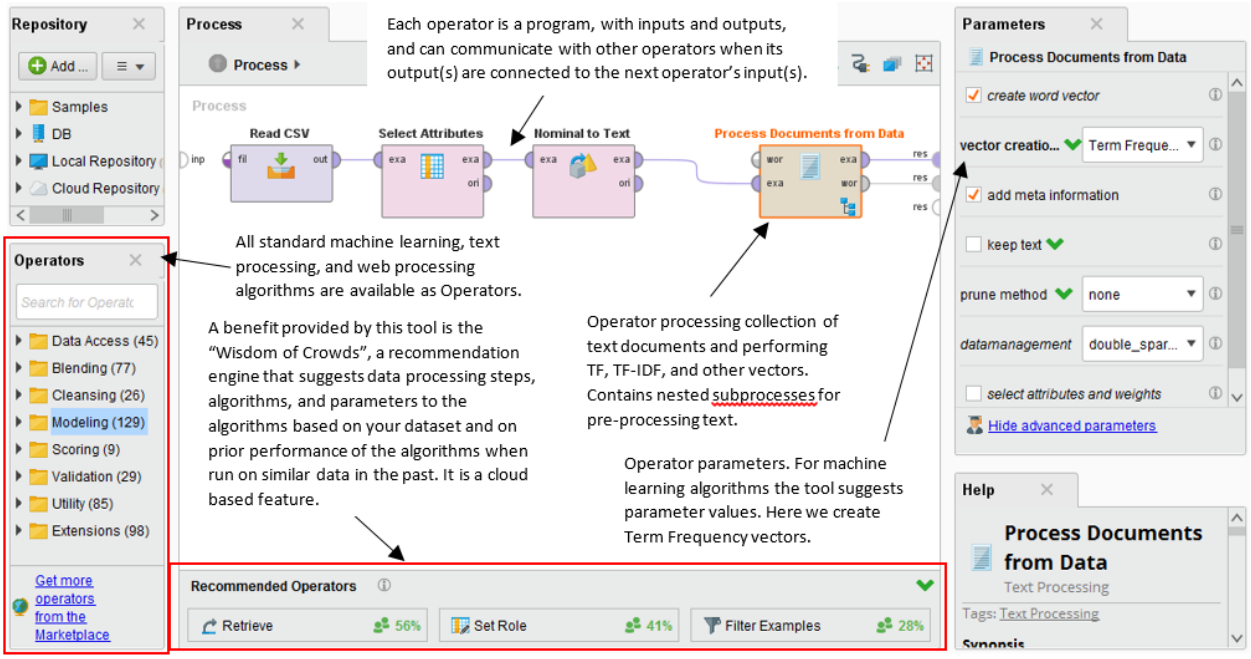

Figure 2. Example of a typical workflow for processing a collection of text documents (with overview of the RapidMiner interface).

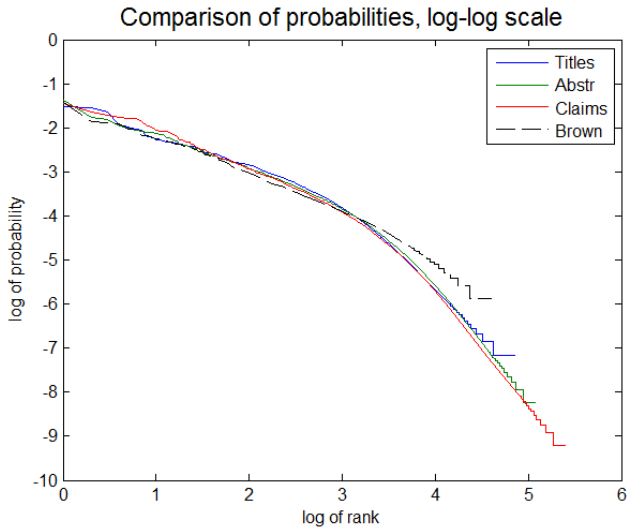

Figure 3. Log-Log representation of the Brown Corpus exhibiting Zipf's law, as compared to the patent titles corpus, the patent abstracts corpus, and the patent claims corpus.

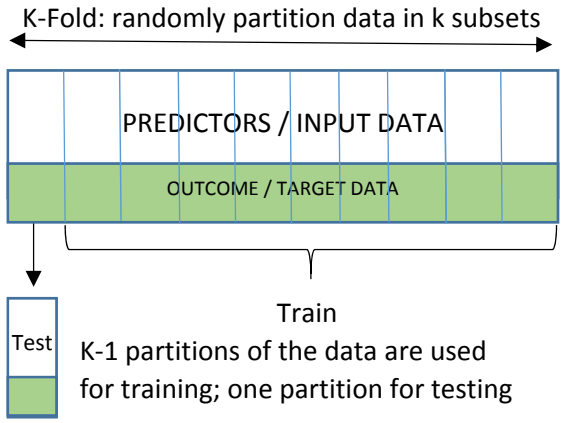

Figure 4. Cross-validation illustrated for the k-fold method

Confusion Matrix 2 by 2

True Positive Rate	**False Negative Rate**
False Positive Rate	**True Negative Rate**

Figure 5. The "Confusion Matrix" is a standard measure to compare the performance of classification algorithms. This 2-by-2 illustrates the tradeoffs when running a machine learning algorithm.

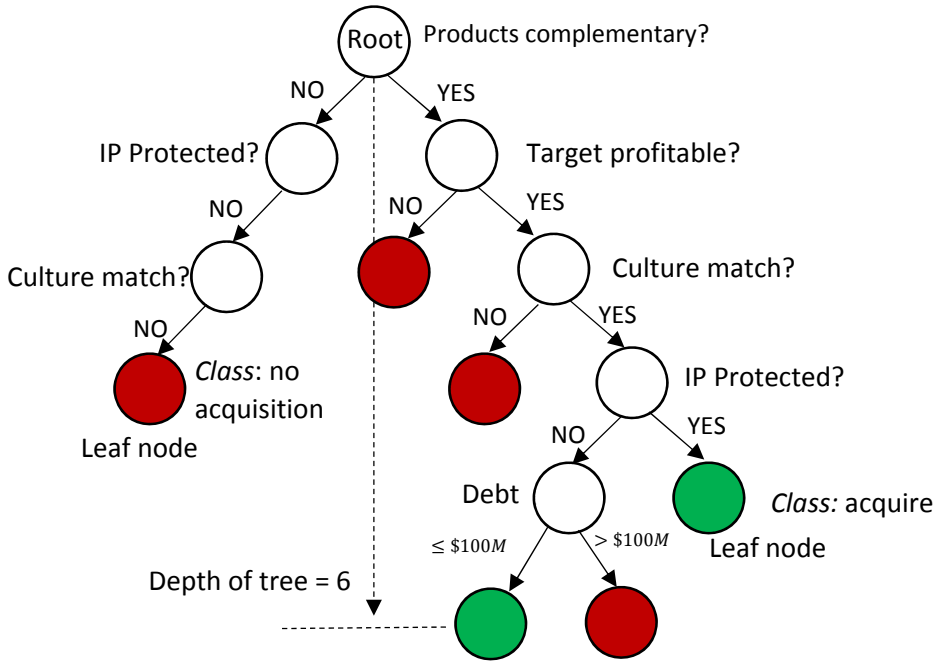

Figure 6. An example of a decision tree with a variety of path lengths

Table 1. ML in strategy and management research

Domain	Problem treated	Method used
Multinationals	Strategy of foreign investments (Debaere *et al.*, 2010), (Roth, 1992). Strategy of resource allocation (Williams and Lee, 2009). Strategy of international marketing, foreign market opportunity assessment (Cavusgil *et al.*, 2004) , (Hu M, Liu B. 2004), (Singh *et al.*, 1996), (Punj G, Stewart DW. 1983), (Wedel M, Kamakura WA. 2012). Supply chains (Stock *et al.*, 2000). Analysis of strategic leadership and executive innovation (Elenkov and Wright, 2005).	Cluster analysis; K-means validated with Ward's method (Singh *et al.*, 1996); (Williams and Lee, 2009) hierarchical clustering (Williams and Lee, 2009); (Stock *et al.*, 2000), (Singh *et al.*, 1996). Establishing control groups for firms in DID models, using kNN, for establishing models for foreign investments (Debaere *et al.*, 2010). Web mining, NLP (web intelligence) (Lau *et al.*, 2012) Clustering (Elenkov and Wright, 2005).
Corporate Governance	Assessing CEO personality (Gow *et al.*, 2016) ; Managerial attention/cognition (Nadkarni and Barr, 2008), (Eggers and Kaplan, 2009); CEO strategy and acquisitions (Gamache *et al.*, 2015); Gender effects (Lee, 2007), (Kanze *et al.*, 2017).	PCA (Slack *et al.*, 2010), (Zhu, 2013), (Lange *et al.*, 2014) Text frequency analysis (Kanze *et al.*, 2017), (Gamache *et al.*, 2015); (Eggers *et al.*, 2017); Text based clustering (Gow *et al.*, 2016)
Financial Markets	Stock market prediction (Bollen *et al.*, 2011), (Tan *et al.*, 2007), (Lugmayr, 2013) Investor sentiment analysis (Tetlock, 2007) Legal issues in finance – liabilities (Loughran, and McDonald, 2011).	Classification, prediction, NLP, Web analysis (Bollen *et al.*, 2011), (Tan *et al.*, 2007). NLP and Web mining (Lugmayr, 2013), (Loughran, and McDonald, 2011), (Tetlock, 2007).
Banking System	Systemic risk, contagious bank failures, system failure prediction (Hu et al. 2012), (Chen et al., 2012) Bank failure prediction (Tan *et al.*, 2007).	Classification, prediction, network model: "Network Approach to Risk Management (NARM)", "Rank-In-Network Principle", and "Link-Aware Systemic Estimation of Risks" (Hu *et al.*, 2012) Network based modeling (Chen et al., 2012)
Credit Prediction	Individuals and corporate credit scoring, credit worthiness prediction, business failure prediction, (Liab and Sun, 2011), (Sohn and Kim, 2012), (Ju and Sohn, 2014), and (Nikoloc *et al.,* 2013).	Decision tree, SVM (Sohn and Kim, 2012) (Cubiles-De-La-Vega *et al.*, 2013). Logistic regression (Nikoloc *et al.*, 2013). (Liab and Sun, 2011). Neural network (NN) (Liab and Sun, 2011), (Cubiles-De-La-Vega *et al.*, 2013). Classification trees (Cubiles-De-La-Vega *et al.*, 2013) and decision trees (Sohn and Kim, 2012).
Market Segmentation	Market level analysis, segmentation (Chiu *et al.*, 2009), (Punj and Stewart, 1983), (Wang, 2009), (Wedel and Kamakura, 2012).	k-means, particle swarm optimization (Chiu *et al.*, 2009)

Domain	Problem treated	Method used
		Cluster analysis (Punj and Stewart, 1983), kernel-based clustering (Wang, 2009), various clustering techniques (Wedel and Kamakura, 2012).
Marketing	Marketing, customer relationship management (Ngaia *et al.*, 2009), (Struhl, 2015). Customer loyalty analysis (Gans *et al.*, 2017). Finding trading rules, competition (Allen and Karjalainen, 1999).	Genetic algorithms, NLP (Allen and Karjalainen, 1999) NLP, social network mining (Gans *et al.*, 2017), (Struhl, 2015).
Firm level management	Trading strategies (Tan et *al.*, 2007). Corporate strategies (Sohn *et al.*, 2003), manufacturing policies (Akhbari *et al.*, 2014). Enterprise logistics (Stock *et al.*, 2000).	Classification, prediction.
Supply chain optimization	Supply chain optimization (Stock *et al.*, 2000).	k-means.
Transportation management	Traffic forecasting (Zhong and Ling, 2014), (Zhang *et al.*, 2013).	kNN Regression (Zhong and Ling, 2014).
Alliance-level decisions	Strategic merging decision, cross-border investments (Lau *et al.*, 2012). Finding synergies for merging and major competitors (Hoberg and Phillips, 2010).	Domain-specific sentiment analysis, business relation mining, statistical Learning, evolutionary learning, business intelligence (Lau et al.). NLP (Hoberg and Phillips, 2010).
Knowledge transfer, innovation, knowledge management	Knowledge transfer (Li *et al.*, 2014). Co-authorship networks of the US patent inventor (Balsmeier *et al.*, 2016). (Choi et al., 2008); knowledge management (Williams and Lee, 2009).	NLP, graph-based methods, clustering, classification, prediction, cluster analysis (Balsmeier et al., 2016), (Li et al., 2014).

Online Appendix

The following two examples are based on freely available data and the steps outlined here are intended to be used as practice for some of the concepts in the paper. The first example centers on prediction, and shows two methods using a freely available credit default dataset. The first example is implemented in RapidMiner and Matlab. The second example is implemented in RapidMiner. The scripts will be made available as part of the online appendix.

Example 1: Prediction of a binary default outcome using decision trees and kNN

The Taiwan credit default dataset is available in the UCI ML repository[7] and was first used in Yeh and Lien (2009). This dataset contains individual characteristics such as age, education, prior payments, and prior bill amounts, as well as a default binary outcome variable (a value of 1 corresponding to a customer defaulting on the loan). The goal is to produce a model that predicts default with good accuracy. The first implementation is that of a decision tree model in Rapidminer. We begin by setting a general framework to run cross-validation, as in Figure A.1. The inside of the Cross-Validation process in RapidMiner allows for choosing the model; in this example, the Decision Tree model is active. Naïve-Bayes and kNN are inactive but shown in Figure A.2 to illustrate running different models in this framework. The results are reported in the confusion matrix 2x2, format as in Figure 5.

[7] Default of credit card clients, UC Irvine Machine Learning Repository, https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients, Accessed July 5th 2017

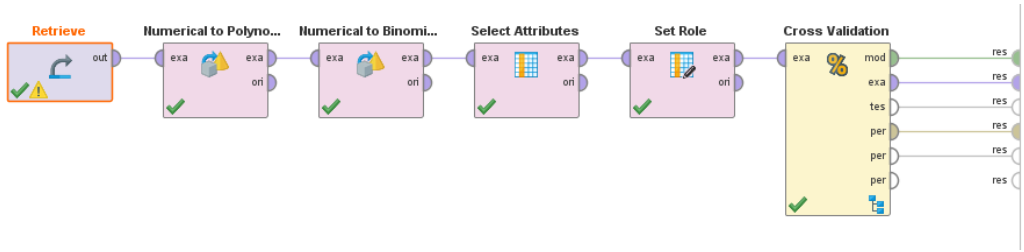

Figure A.1. Cross-validation implementation example; the Set Role operator designates the outcome variable (the variable containing the classes); the cross-validation operator contains the choice of model and the performance measurement.

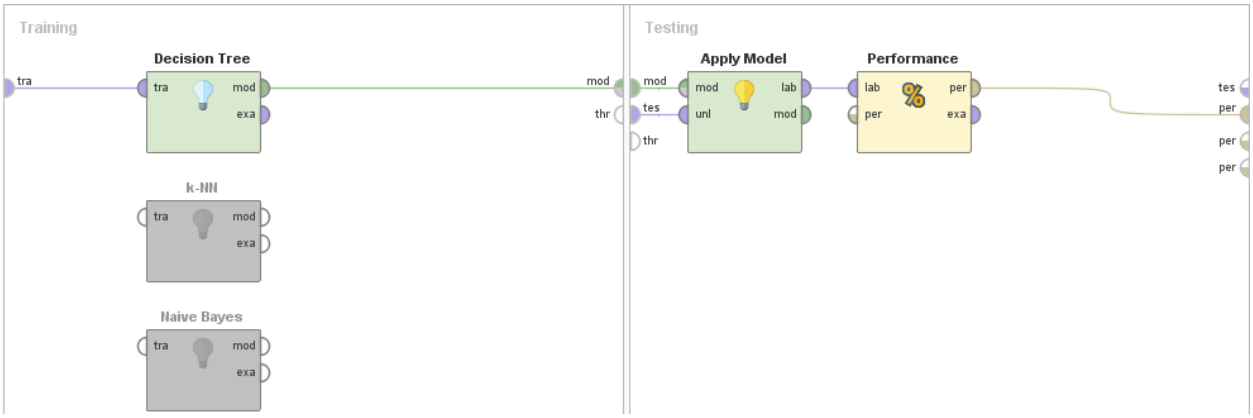

Figure A.2. Inside the cross-validation operator, the training component trains the chosen model (in this case a decision tree was used, while leaving as examples k-N-N and Naïve Bayes as inactive alternatives), whereas the testing component applies the model and determines the performance characteristics of the classifier, including the accuracy and the confusion matrix.

In this example, the decision tree approach yields a classification accuracy of roughly 80 percent, though the confusion matrix explained in Figure 5 paints a more complete picture, namely that the algorithm performs poorly on classifying true positives (default) but quite well

on classifying negatives (no default). This piece of information may be valuable in the decision making process.

accuracy: 81.95% +/- 0.80% (mikro: 81.95%)

	true false	true true	class precision
pred. false	22417	4467	83.38%
pred. true	947	2170	69.62%
class recall	95.95%	32.70%	

Figure A.3. Sample confusion matrix based on a simple decision tree model applied to a test dataset.

Next, consider the typical kNN classifier, applied to the same data (ran in Matlab). Depending on the value of k and using the Euclidean distance on the same input data, the classification errors vary as in Figure A.4. All predictors have been normalized to the interval [0,1]; no weights were used. The errors with k=4 are similar to the ones obtained with the decision trees.

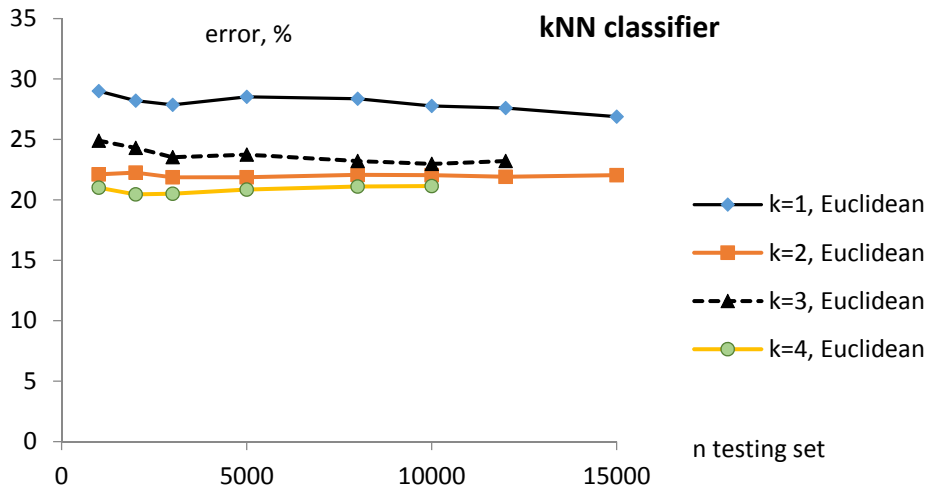

Figure A.4. kNN classifier errors.

The choice of the weights may degrade or improve the error of the classifier. In Figure A.5., the effect of the weight is slightly detrimental.

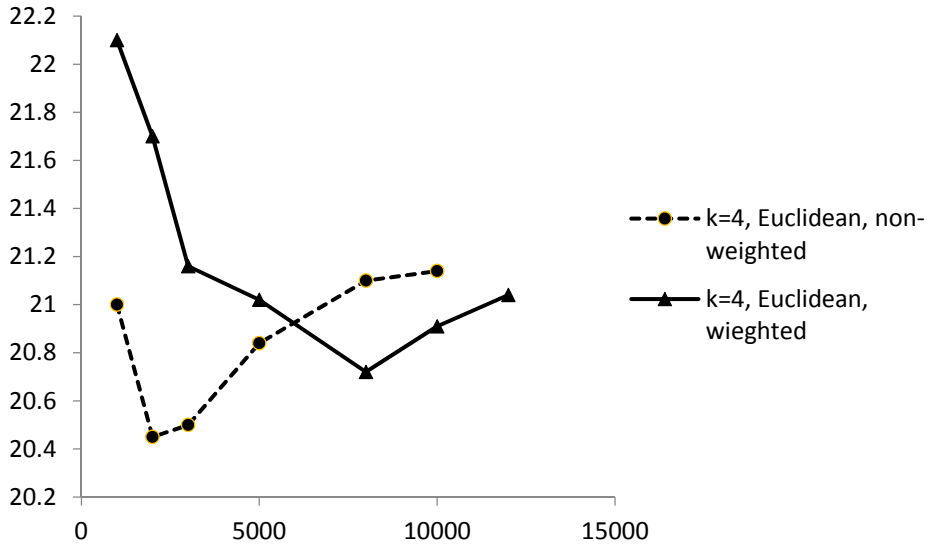

Figure A.5. kNN classifier, same k =4, Euclidean distance, unfavorable weighting and unweighted (comparable to Figure A.4).

The methods outlined above are standard algorithms for classification. However, in practice, there are issues to be considered by both the strategist and the data analyst. Recall that features with no variance bring no information in the classification process. One of the features having large variance in the example above is the gender of the borrower. Thus, one can expect that gender will play a significant role in the classification and ultimately in the decision. However, that may point out a bias in the form of discrimination due to gender, which may require further analysis prior to inclusion in a prediction model. There are costs to include certain features in prediction, such as penalties the firm may pay in terms of reputation and customer loyalty. In the case of Target and the prediction of customer pregnancy, while highly accurate at prediction, the fact that the company was running such a model had a high reputational cost including unwanted attention on the lack of privacy Target customers had

while shopping; for more on this and other examples see (Siegel, 2013). Other features in the data, like age, may rise similar questions. In practice, there is a tradeoff between these costs and accuracy of prediction.

Next, the strategist and data analyst must decide what distance should be used. From the point of view of the analyst, the best distance to be used and the best value of k are those which produce the lowest prediction errors. If the strategists are conservative and wish to minimize the number of defaults, they wish a good precision for predicting only the non-default cases. In this case, a larger value of k may be needed (similarity with more neighbors); in addition, the standard criterion in the kNN algorithms (simple majority vote) may have to be changed, for example to similarity with at least two thirds of the neighbors. Depending on the application, the maximum distance (Chebyshev) may be better instead of the Euclidean distance – the distance is another parameter in the model. All these are conscious choices one must make when designing the prediction model.

Raw prediction accuracy is not necessarily sufficient to judge whether the model is good or not for a particular application. For example, there may be a higher cost associated with false positives than false negatives; these costs may be quantified financially. In the example above, the strategist may ask the data analysis team to modify the classifying tool such that to maximize the profit, taking into account that potential borrowers wrongly predicted as likely to default mean lost, irreplaceable customers, while borrowers predicted trust-worthy but who defaulted produce damages – the two costs may however be substantially different and vary from business to business. A suitable criterion for the accuracy of the classifier may thus be the maximization of:

$$G(p_+ - p_+^-) - Lp_-^+$$

where G is the average gain produced by a reliable borrower, p_+ is the probability of correctly predicting reliable borrowers, p_+^- is the probability of losing a reliable borrower because of a false positive, L is the loss due to a defaulting customer, and p_-^+ is the probability that a loan was given to a customer predicted as reliable, but who defaulted. Such a choice of criterion for the classifier may be quite different from the textbook examples but is more informed by the specifics of the setting, and as such would involve the business strategist. In practice, the power of using machine learning techniques largely reside in the co-operation between the strategist (familiar with the setting and business constraints) and the data analyst (familiar with a wide variety of prediction models).

Example 2: Sentiment Analysis Applied to Competitors

The corporate motto fulfills essential roles for the firm: attracting customers, distinguishing the firm from competitors, signaling the core of the firm's culture, and motivating employees. They are essential in commerce and have been in use for hundreds of years and can be trademarked as part of the firm's brand. Firm mottos are designed to evoke emotion – in the customer, the employee, and the competitor – and as such linguistic properties related to sentiment and degree of subjectivity are useful in classifying them. Firms which evoke similar sentiments as found in the text analysis may be competing for a particular type of customer and may be closer in competition than others with very different corporate mottos; this may aid the researcher with a layer of classification beyond that of industry and location. For this example, I chose again a freely available dataset and a set of intuitive tools. The dataset is a listing of

corporate mottos used by banks and is collected by The Financial Brand and available online[8]. The version of the data current as of the writing of this appendix and used for this analysis contains 888 financial firms. The analysis was done in RapidMiner with two different NLP sentiment analysis packages, AYLIEN and the Meaning Cloud Sentiment Analyzer. The misclassification error was lower for this data using the latter package; I present the steps and output from the Meaning Cloud package. Both packages run on the cloud (send the input data to a server and process it on the server before sending the results back to RapidMiner) and as such require registration for a free account (the free account was more than sufficient for a dataset this size and both allow tens of thousands of lines of text analyzed per day for free. Implementing this in Python NLTK is of course possible, though will require significant implementation effort. The schematic for the RapidMiner process is in Figure A.4:

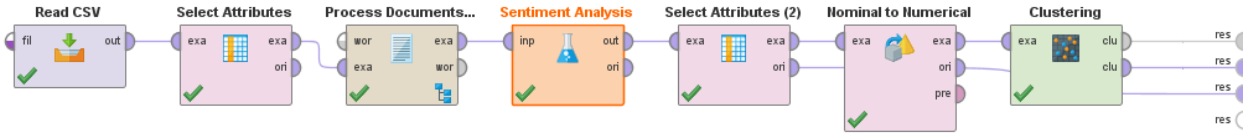

Figure A.4. Implementation of a sentiment analyzer and clustering based on sentiment characteristics of text; the clustering step is optional – the output of the Meaning Cloud sentiment analysis operator is a vector of characteristics (polarity, on a five point scale ranging from very negative to very positive; objectivity taking values of objective or subjective; irony taking values of ironic or not ironic; agreement taking a binary value as well; and finally the confidence of the classification on a scale of 0 to 100).

[8] "The Biggest List of Financial Slogans Ever", The Financial Brand, https://thefinancialbrand.com/1779/financial-slogans/, Accessed September 30, 2017.

A sample of the output is in Figure A.5 and the summary view of the clusters in Figure A.6.

Row No.	FirmName	cluster	text	polarity(text) = NONE	polarity(text) = P+	polarity(text) = P	polarity(text) = N+
1	121 Financial Credit Union	cluster_9	banking focused on you	1	0	0	0
2	1st Advantage Bank	cluster_9	expertise you need service you deserve	0	1	0	0
3	1st Financial Federal Credit Union	cluster_0	the better way to bank	0	0	1	0
4	1st Mariner Bank	cluster_0	we built this bank for you	0	0	1	0
5	1st National Bank	cluster_9	grow with us at home with first	1	0	0	0
6	1st National Bank of So. Florida	cluster_9	your first choice	1	0	0	0
7	1st Source Bank	cluster_9	your partners from the first	1	0	0	0
8	A+ Federal Credit Union	cluster_9	with you every step	1	0	0	0
9	ABN AMRO Bank	cluster_9	making more possible	1	0	0	0
10	ABNB Federal Credit Union	cluster_0	open honest hardworking	0	0	1	0
11	AIG	cluster_0	the strength to be there we know money	0	0	1	0
12	ANZ	cluster_5	the better we know you the more we can do	0	1	0	0
13	ASB Bank	cluster_0	creating futures	0	0	1	0
14	Abbey National Bank	cluster_6	get the abbey habit turning banking on its head...	0	0	0	1
15	Abington Bank	cluster_0	banking for people with better things to do	0	0	1	0
16	Absa Bank	cluster_9	today tomorrow together	1	0	0	0
17	Access National Bank	cluster_9	the difference is access	1	0	0	0
18	Achieva Credit Union	cluster_0	dream it achieve it	0	0	1	0
19	Acru	cluster_9	money life	1	0	0	0
20	Addison Avenue FCU	cluster_0	we listen you prosper	0	0	1	0
21	Advantage Plus FCU	cluster_4	not for profit for people	0	0	0	1
22	Afena Credit Union	cluster_9	we are already there	1	0	0	0
23	Affinity Group Credit Union	cluster_9	changing lives one member at a time	1	0	0	0
24	Affinity Plus FCU	cluster_4	not for profit for people	0	0	0	1

Figure A.5. RapidMiner output (partial) showing some of the text sentiment categories and the cluster assignments for a sample of the firms.

Cluster Model

```
Cluster 0: 318 items
Cluster 1: 1 items
Cluster 2: 3 items
Cluster 3: 4 items
Cluster 4: 4 items
Cluster 5: 14 items
Cluster 6: 1 items
Cluster 7: 5 items
Cluster 8: 1 items
Cluster 9: 537 items
Total number of items: 888
```

Figure A.6. Output of ten clusters based on the sentiment analysis vectors. Cluster 9 is dominated by firms with no sentiment polarity; cluster 0 by firms with positive sentiment found in the mottos.

The confidence in the classification is at above 99%. Interestingly only 15 firms had a negative message, whereas 376 firms had positive messages. Of the 888 texts, 177 were classified as subjective; no firms had irony in their mottos.

This is a very simple example of how one may use sentiment analysis to determine firms with similar strategies; of course, the text analysis portion would need to be supplanted by other data and other methods. However, the steps outlined here can easily be extended to other data and other questions. This appendix serves as a simple set of examples to guide the reader to implementations of the concepts and tools found in the paper. To serve that purpose, both examples are available with free and tools and datasets. The UCI ML repository is an excellent source of additional examples, papers, code and data, and several of the machine learning books outlined in the references point to additional examples and online sources for the interested reader.