2008

Information extraction in text mining

Matt Mulins
Western Washington University

Information Extraction in Text Mining

Matt Mullins
Computer Science Department
Western Washington University
Bellingham, WA
mullinm2@cc.wwu.edu

Information Extraction in Text Mining

Matt Mullins
Computer Science Department
Western Washington University
Bellingham, WA
mullinm2@cc.wwu.edu

ABSTRACT

Text mining's goal, simply put, is to derive information from text. Using multitudes of technologies from overlapping fields like Data Mining and Natural Language Processing we can yield knowledge from our text and facilitate other processing. Information Extraction (IE) plays a large part in text mining when we need to extract this data. In this survey we concern ourselves with general methods borrowed from other fields, with lower-level NLP techniques, IE methods, text representation models, and categorization techniques, and with specific implementations of some of these methods. Finally, with our new understanding of the field we can discuss a proposal for a system that combines WordNet, Wikipedia, and extracted definitions and concepts from web pages into a user-friendly search engine designed for topic-specific knowledge.

Keywords

Text Mining, Data Mining, Information Extraction, Natural Language Processing, Knowledge Discovery, Semantics, Concept Mining, Wikipedia, WordNet

Table of Contents

1. INTRODUCTION

Text mining can be defined as "the discovery by computer of new, previously unknown information, by automatically extracting information from different written resources" [1]. Whereas Data Mining extracts patterns from structured databases or XML files, text mining revolves around extracting information from unstructured or semistructured sets of text data. This text can be anything from emails or web pages to medical abstracts, news articles, or business reports. With the abundance of text data now available via the Internet, it is clear the field of text mining has become increasingly useful. It has been applied in many domains including business, health care, government, education, manufacturing, history, sociology, research, and crime detection.

Text mining is really at the crossroads of other fields like Data Mining, Knowledge Discovery, Information Retrieval (IR), Machine Learning, and Natural Language Processing (NLP). It borrows many algorithms and techniques from these fields, as well. Another field that lends much to the discovery of information in text mining is that of Information Extraction (IE). IE and text mining differ in that IE involves the extraction of specific, structured information and predefined relations while text mining involves the discovery of general, unsuspected information and new relations. Yet, as we'll see later in the paper, IE can help us greatly in the knowledge extraction stage.

The rest of this paper is organized as follows. Section 2 describes various technologies and methods we can use in text mining, delving deeper into the areas of NLP techniques, IE techniques, representation models, and categorization techniques. In Section 3 a few of these methods are demonstrated in the form of five specific implementations grouped into three different types. We summarize the experimental results of these implementations in Section 4 and discuss the advantages and disadvantages of the general methods and technologies in Section 5.

2. TECHNOLOGIES & METHODS

The "key to text mining is creating a technology that combines a human's linguistic capabilities with the speed and accuracy of a computer" [2]. Though computers are still a long way from understanding and comprehending language completely, there have been many technological advancements that aim to approach human comprehension.

Summarization. Text summarization generates summaries for large text documents. These summaries hopefully reduce the length and detail while maintaining key points and overall meaning. This is challenging, though, because it is difficult to teach a computer to interpret semantics or meaning. Instead shortcuts are used. Sentence extraction extracts sentences deemed statistically important, using heuristics like positions in the document (opening sentences), significant phrases ("in conclusion"), or formatting (bolded terms or headings). Summarization techniques are especially useful in search engines to allow a user to assess the relevance of returned results.

Categorization. Sometimes we want to identify the major themes of a document and place it into one or more categories. This is where the classification methods of Machine Learning can help, whether it be Naïve Bayes or Support Vector Machines. Most of these methods use the same document representation: the Vector Space Model, or as it is typically called, the “bag-of-words” (BOW) representation. This means there is no attempt to process semantic meaning or context, each term is examined alone and a vector of the term frequencies is the resulting representation. Later on we’ll examine extensions and alternatives to this model. We can use thesauruses to define and relate terms found to create better categorizations.

Clustering. Clustering groups similar documents into sets. This may sound like categorization, but clustering doesn’t use predefined topics, it does its classifying based on similarity measures between documents. A document can end up in multiple clusters, and the clusters can even be hierarchical, expressing subtopic and supertopic relations.

Concept linkage. By identifying shared concepts between documents, concept linkage helps users find buried information. It is especially applicable in biomedicine, where there is so much information it is impossible for one user to read it all and make valuable connections. A good example of this is Don Swanson’s work [3] on hypothesizing causes of rare diseases by looking for indirect links between documents.

Question Answering. In order to efficiently retrieve information from a collection of documents we can design systems that can hand back answers to questions posed in natural language. Question Answering deals with many different question types and can be split into two tasks: the creation of a structure that can represent the knowledge, like an ontology, and the design of the system to convert natural language questions to queries that can retrieve this knowledge. Higashinaka and Isozaki [4] describe such a system.

Information Extraction. IE technologies are the starting point for the analysis of text. In order to identify relationships, match patterns, and extract structured information from unstructured text, we implement a slew of techniques borrowed from NLP.

2.1. NLP Techniques and Difficulties

The complexity and ambiguity inherent in human language is a giant hindrance to successful computer understanding. Even with large, complex grammars, there is always the issue of people using language freely, without adhering to rules. Misspellings, abbreviations, and slang are just a few problems. We summarize some issues we come across and methods we can use to deal with them.

Usually text documents contain many words that aren’t necessary to understand the general idea of the text. These high frequency words like ‘a’, ‘the’, and ‘of’ are called stop words and can be outright ignored in many situations. This is the approach fields like IR take to reduce the dimensionality of their term spaces and improve performance. It is less useful in the field of text mining, however, because these words can often lend information and clarify semantics. For example, the statement “He was promoted” contains two potential stop words: “he” and “was”. Without “was” we interpret an entirely different meaning: “He promoted”.

Stemming (or lemmatization) is also commonly used in IR. To group similar words into one, we reduce words to their stem, or root form. For example, “walking”, “walk”, “walked”, and “walker” will all be reduced to the root word “walk”. Although it can be argued that this effect is not as harsh as stop word lists, it can still be harmful to the semantics of the text.

To eliminate noisy data we can use spelling correctors and acronym and abbreviation expanders. These usually require a dictionary or thesaurus.

We also must attempt to deal with one of the larger issues in NLP as a whole, ambiguity. The word sense disambiguation (WSD) problem deals with finding the most probable sense of a polysemous word (a word with multiple meanings). We can approach this problem by considering the context in which the word occurs, to, for instance, determine if a word is a noun or verb.

Tagging involves the labeling of words in a corpus with part of speech (PoS) tags or XML markup. PoS tags label syntactic categories like nouns, verbs, and adjectives in order to identify syntactic structures like noun phrases or verb phrases.

A collocation is a sequence of words that are commonly used together but mean different things if separated. An example of this is “light rain”. Ideally we will treat this as a sole unit instead of splitting it into two words because we lose the meaning of the collocation if we only consider the words separately.

And finally, tokenization is a subject we’ll encounter at some point in the process. Do we want to split the text into units of paragraphs, sentences, phrases of a certain length, or single words? To aid our splitting we can take advantage of delimiters present in the text; spaces, tabs, punctuation marks, or even certain stop words. We can even use a method like n-grams to find frequent word phrases in the document.

2.2. IE Techniques/Problems/Methods

There are many approaches to constructing IE systems. We can manually create pattern-matching rules to identify the desired entities and relations, but this proves to be difficult and tedious and doesn’t result in very robust systems. Alternatively, we can use supervised machine-learning methods trained on human annotated corpora to develop our systems. We can automatically learn pattern-based extraction rules with the help of patterns that match the beginning and ending of phrases or inductive logic rules.

We can also turn to sequence labeling, which assigns labels to each token in the document to facilitate extraction. Even statistical sequence models like the Hidden Markov Model (HMM) or Conditional Random Field (CRF) can be applied, learning their parameters from supervised training corpora. Standard feature-based classifiers can help as well, predicting the label of each token based on the token and its surrounding context. By representing the context using a set of features that include the one or two tokens on either side of the target token and the previously extracted labels, we can generalize the sequence labeling problem to decision trees, boosting, memory-based learning (MBL), support-vector machines (SVMs), maximum entropy (MaxEnt), transformation-based learning (TBL), and many others. For more details on these techniques, see [5].

2.3. Representation Models

The most widely used representation of text is the Vector Space Model (VSM). The text is described by a vector whose dimension is the number of text features and whose content is a function of the frequencies with which these features appear. Things like the order and relations between words are totally ignored, which is why this model is also called the "bag-of-words" (BOW) model. Most other representations are extensions of the BOW model. Some focus on phrases instead of single words, some give importance to the semantics and relations between words, and others take advantage of the hierarchical nature of the text. We can take into consideration the location of words; the first paragraph of a document is often an introduction while the last is usually a conclusion. The context of a word is also a very useful piece of semantic information. We can represent the context as a vector that contains the co-occurrence frequencies between a term and set of features. We can also just use context patterns; regular expressions with PoS tags and ontology information. N-grams, which are subsequences of words, are also used to represent and categorize the context of text. Instead of focusing on context we can turn towards sense-based representations, which use meanings of words as features instead of the words themselves. Matrix space models have been one of the more recent proposals for text representation. Term-by-section, term-by-paragraph, and term-by-sentence matrices can be created. The techniques are similar to Latent Semantic Indexing (LSI), which also uses matrices to describe the occurrence of terms in documents. For more details on these models, see [6].

2.4. Categorization Techniques

In order to organize unstructured collections of documents we use text categorization. In so doing, we facilitate storage, search, and browsing. By categorizing, we may even discover links and patterns between documents that weren't easily noticeable. We can use supervised or unsupervised categorization.

For supervised classification, we first define the documents that will be used. We have a training set with annotated documents, the development set used to test the classifier before it is completed, and the test set that comprises the documents which will evaluate the performance of the classifier. Since the cost of text annotation is high and obtaining the training data is difficult, we can use smaller sets of labeled data with alternative semi-supervised techniques.

Unsupervised classification is called clustering. There are no labeled documents. We define a similarity measure and the documents are compared with each other and divided into different clusters.

In text mining we can use categorization in many ways. For thematic categorization we focus on noun terms that can characterize a topic. Automated learning algorithms like SVM, k-nearest-neighbor, and Linear Least Squares Fit (LLSF) have been shown to perform better than others like neural networks, decision trees, and Naïve Bayes. For sentiment classification we turn our attention towards the subjective opinion of the author. The semantic orientation of a word, like whether it's positive or negative, is the main focus. We can look at adjectives and connecting conjunction words, modeling where appropriate and taking into consideration context to try to catch metaphors and irony.

3. METHOD IMPLEMENTATIONS

3.1. Concept/Definition Finding

Instead of regular “find one page” searches, we can strive for “learn about a topic” searches. We can look for topic specific knowledge which includes concepts and definitions rather than just links to articles. We want to help the user to perform systematic learning of a topic on the Web instead of browsing through page after page of documents rated highly because of their popularity, but not necessarily because of their informative potential. Liu et al. [7] outline a prototype system that implements discovering salient concepts and identifying informative pages containing definitions. Their goal is to “help people learn in-depth knowledge of a topic systematically on the Web.” They use what they call their WebLearn algorithm to parse pages and an Apriori algorithm for handling transactions.

WebLearn, in a nutshell, submits a topic to a search engine, which returns a set of relevant pages. The system then mines the sub-topics or salient concepts using a set of top ranking pages from the search engine. The informative pages are then discovered (they are the pages containing the definitions of the topic and subtopics (salient concepts). The user is then presented with the concepts and informative pages and has the opportunity to perform the process again on one or more subtopics.

Nested inside WebLearn is the Apriori association rule mining algorithm. They mine frequent occurring phrases to find frequent itemsets as possible subtopics or salient concepts. Definitions are identified using linguistic rules and infallible heuristics, mostly with the aid of definition identification patterns that are suitable even for the noisy environment of Web pages.

In the end the proposed system yields, on average, a higher precision than both Google and AskJeeves.

3.2. Leveraging Wikipedia/WordNet/PhraseNet

By mining synonym, hypernym, polyseme, and associative relations from Wikipedia, Hu et al. [8] build a concept thesaurus for semantic relations. By analyzing structure and links they are able to extract semantics. They then build a framework to integrate the relations into traditional text similarity measures for the purpose of document clustering. Much of their techniques take advantage of Wikipedia’s structure: their general titles for concepts, their redirect links for synonyms, their disambiguation pages for polysemy, their hierarchical category structure for hypernymy, and the abundance of hyperlinks on each article’s page for relatedness. They then use this cleaned version of Wikipedia as a thesaurus to enhance the traditional text similarity measures, replacing or appending to original document vectors.

Many other techniques use things like WordNet as a thesaurus, which is man-made for the purpose of acting as a semantic lexicon and consequently much more structured. But there is something to be said about the large, constantly updated, diverse structure of Wikipedia. Suchanek et al. [9] unify WordNet and Wikipedia to create the largest publicly available formal ontology with near human accuracy of around 95%. By using Wikipedia category pages to generate candidates for concepts and relations, they take advantage of the immenseness of Wikipedia. By using WordNet they tie down their Wikipedia data with a clean and carefully assembled hierarchy. There are multiple techniques used to link the two which yields great overall accuracy. A data model of entities, facts, binary relations between facts, and properties of relations is established using some of the same information extraction techniques talked about earlier. Relations like “subClassOf”, “means”, “bornInYear”, “locatedIn”, “politicianOf”, etc. are mined for in Wikipedia and written into their ontology. This ontology is decidable, extensible, and compatible with Resource Description Framework Schema (RDFS).

And still others try to build more features on to WordNet while maintaining its strengths. Li et al. [10] complain that since WordNet is organized at the word level, it suffers from ambiguities. They propose PhraseNet, a system based on the assumption that semantic ambiguity in English mostly disappears when a word’s local context is taken into account. Their system is context-sensitive and, while an extension of WordNet, stands as an independent lexical semantic knowledge base system. They define context hierarchically, with abstract syntactic skeletons, consets (words in their context, together with all relations associated with them) and Hyper and Equal relations, forming a graph of contexts.

3.3. Boosting and pLSA

As we’ve discussed previously, term-based representations of documents largely disregard lexical semantics and, as a result, aren’t sensitive to linguistic variations such as vocabulary and word choice. To tackle this we can use concept-based document representations to supplement word or phrase-based features. Cai et al. [11] use probabilistic Latent Semantic Analysis (pLSA) to extract concepts and then the AdaBoost machine-learning algorithm to combine multiple weak classifiers into a strong classifier. This gives us a framework for combining semantic features with term-based features.

Latent Semantic Analysis (LSA) is a technique for reducing dimensions in co-occurrence and count data. It uses a singular value decomposition (SVD) to map documents from the standard vector representation to a lower dimensional latent space. The term co-occurrence statistics can capture semantic relationships among terms and documents. For example, synonym dimensions in the vector space will be reduced to a single dimension in the latent space. Probabilistic Latent Semantic Analysis (pLSA) builds a statistical foundation on to LSA and is used in the authors’ system for its potential for automatically identifying factors that may correspond to relevant concepts or topics.

Boosting is a powerful machine learning approach that combines many weak, moderately accurate classifiers into one highly accurate ensemble. The authors use AdaBoost, the most popular boosting algorithm. After the pLSA stage has augmented the document representation with concepts to identify synonyms and polysemes, AdaBoost is the framework that combines the semantic features with the term-based features. This approach substantially increases accuracy gains.

4. EXPERIMENTAL RESULTS

Though the implementations discussed in Section 3 all have their own respective results, due to the variance in methods, corpora, and goals, we can't directly compare their results. Instead we highlight their findings in Table 1.

Table 1. Results of different method instances.

	Results
Concept/Definition Finding	• Average precision of WebLearn much better than Google and AskJeeves (61.23 compared to Google's 18.44 and AskJeeves' 16.88) • Execution time takes on average 2.5 minutes, but this didn't concern the authors and can be easily improved
Leveraging Wikipedia/WordNet/PhraseNet	• Clustering performance improved compared to previous methods • With optimized parameters based on a few user-provided labeled data, performance even higher (16.8% and 18.8% over baselines) • YAGO ontology 95% accurate • PhraseNet reduces error rate of classifier from baseline to 12% compared to WordNet's 5.7%
Boosting and pLSA	• Consistent and substantial accuracy gains (in the range of 5%)

5. DISCUSSION

In Table 2 we compare and contrast the key features of some of the technologies, methods, and representation models we've discussed.

Table 2: Key features of selected technologies, methods, and representation models.

	Advantages	Disadvantages
Word-based	• Good statistics • Synonyms • Use of WordNet possible	• No context information • Open to ambiguity • Problems with collocations
Phrase-based	• Context information • Semantic quality • Collocations can be captured	• Statistical quality isn't excellent
VSM (BOW)	• Simple measures for document similarity	• Can only capture frequencies of words
Stop Words Lists and Stemming	• Improve performance by reducing dimensionality	• Loss of information that might clarify semantics
Supervised Classification	• Labels are predefined	• Computationally costly and difficult to obtain learning data
Clustering	• No training data needed	• Not as explicit as supervised classification
Concepts/Relations	• Richer semantics and knowledge potential	• More difficult to work with and generate easy models for

6. INSPIRATION AND FUTURE RESEARCH

As we can see from this brief survey, while the fields of text mining and information extraction are rich with proven techniques and promising results, they also offer new directions and hopes of leaps in the areas of efficiency, accuracy, and usability. This trend will only strengthen as more and more knowledge floods the Internet and users all over the globe strive for efficient extraction and interpretation of information. With the insight provided by this survey I propose a system that combines some of the techniques and goals discussed in [7-9]. By combining the extraction of knowledge from Wikipedia with the clean verification of WordNet and using this thesaurus to pinpoint concepts and definitions in regular pages, we can accomplish the goal set forth by Liu et al. [7] and create a search engine that would teach us about a topic instead of just listing pages where we might find answers to a very specific question. With my understanding of the inner workings of the respective algorithms there should be minimal interface problems and a very user-friendly, user-centric, useful system could come out of it.

7. REFERENCES

[1] M. Hearst, “What is Text Mining?”; http://people.ischool.berkeley.edu/~hearst/text-mining.html.

[2] W. Fan et al., “Tapping the power of text mining,” *Commun. ACM*, vol. 49, 2006, pp. 76-82.

[3] D.R. Swanson and N.R. Smalheiser, “Assessing a gap in the biomedical literature: magnesium deficiency and neurologic disease,” *Neuroscience Research Communications*, vol. 15, 1994, pp. 1-9.

[4] R. Higashinaka and H. Isozaki, “Automatically Acquiring Causal Expression Patterns from Relation-annotated Corpora to Improve Question Answering for why-Questions,” *ACM Transactions on Asian Language Information Processing (TALIP)*, vol. 7, 2008, pp. 1-29.

[5] R.J. Mooney and R. Bunescu, “Mining knowledge from text using information extraction,” *SIGKDD Explor. Newsl.*, vol. 7, 2005, pp. 3-10.

[6] A. Stavrianou, P. Andritsos, and N. Nicoloyannis, “Overview and semantic issues of text mining,” *SIGMOD Rec.*, vol. 36, 2007, pp. 23-34.

[7] B. Liu, C.W. Chin, and H.T. Ng, “Mining topic-specific concepts and definitions on the web,” *Proceedings of the 12th international conference on World Wide Web*, Budapest, Hungary: ACM, 2003, pp. 251-260.

[8] J. Hu et al., “Enhancing text clustering by leveraging Wikipedia semantics,” *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, Singapore, Singapore: ACM, 2008, pp. 179-186.

[9] F.M. Suchanek, G. Kasneci, and G. Weikum, “Yago: a core of semantic knowledge,” *Proceedings of the 16th international conference on World Wide Web*, Banff, Alberta, Canada: ACM, 2007, pp. 697-706.

[10] X. Li, D. Roth, and Y. Tu, “PhraseNet: towards context sensitive lexical semantics,” *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, Edmonton, Canada: Association for Computational Linguistics, 2003, pp. 87-94.

[11] L. Cai and T. Hofmann, “Text categorization by boosting automatically extracted concepts,” *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, Toronto, Canada: ACM, 2003, pp. 182-189.