IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

TEXT MINING USING NLP

Karen Nellimella, Pranav Thorat, Suraj Uikey, Rohan Godage

Student, Computer Engineering
Shree Ramchandra College of Engineering

1. Introduction

Text mining covers a wide range of theoretical approaches and methods that have one thing in common: text as input information. This extends from classic data mining extensions to texts to more sophisticated formulations such as "Use a large online collection of texts to discover new facts and trends about the world itself." Various definitions are possible (Hearst1999). In general, text mining is an interdisciplinary field of activity between data mining, linguistics, computational statistics, and computer science. Standard techniques are text classification, text clustering, ontology and classification construction, document summarization, and potential corpus analysis. In addition, many techniques from related fields such as information retrieval are widely used. The classic application for text mining (Weiss et al. 2004) comes from the data mining community such as document clustering (Zhao and Karypis 2005b, a; Boley 1998; Boley et al. 1999) and document classification (Sebastiani 2002) increase. In both cases, the idea is to convert the text to a structured format based on the frequency of terms and then apply the standard data mining technique. A typical application for document clustering is grouping documents from news articles or information services (Steinbachetal. 2000), methods of text classification, such as B. email filters, and automatic labeling of corporate library documents (Miller 2005). Certain distance measures such as cosine (Zhao and Karypis 2004; Strehl et al. 2000) play an important role, especially in clustering. In recent years, more innovative text mining techniques have been used in various disciplines of analysis.

According to Language Metrics (Gir´onetal. 2005; Nilo and Binongo 2003; Holmes and Kardos 2003), the probability that a particular author wrote a particular text can be determined by analyzing the author's writing style or in a search engine. It will be calculated. To learn document rankings calculated from search engine logs of user behavior (Radlinski and Joachims2007). With the latest developments in document exchange, they have created a valuable concept for automatic editing of text. The Semantic Web (Berners-Lee et al. 2001) propagates standardized format templates for document exchange, allowing agents to perform semantic operations on them. This is implemented by providing metadata and annotating the text with tags. An important format is RDF (Manola and Miller 2004), and attempts have already been made to process this format in R (R Development Core Team 2007) using the Bioconductor project (Gentleman et al. 2004, 2005). It has been. This development gives you great flexibility in exchanging documents. However, as XML-based formats (for example, RDF / XML as a common representation of RDF) become more popular, the tool needs to be able to process XML documents and metadata. The advantage of text mining lies in the large amount of valuable information hidden in the text. It is not available in traditional structured data formats for a variety of reasons.

Text has always been the standard way to store information for hundreds of years., due to time, staff, and cost constraints, text cannot be placed in a well-structured format (such as data-frames or tables). The amount of data is increasing exponentially every day. Almost all types of institutions, organizations and industries store their data electronically. A huge amount of text flows over the Internet in the form of digital libraries, repositories, and other textual information such as blogs, social networks, and email. Determining the right patterns and trends to extract valuable knowledge from this large amount of data can be a daunting task. Traditional data mining tools cannot process textual data because it takes hours and effort to extract the information. Text mining is the process of extracting interesting and important patterns for exploring knowledge from text data sources. Text mining is an interdisciplinary field based on information retrieval, data mining, machine learning, statistics, and computational linguistics.

You can apply several text mining techniques such as summarization, classification, and clustering to extract knowledge. Text mining processes natural language text stored in semi-structured and unstructured formats. Text mining technology is continuously applied in industry, science, web applications, the Internet, and other areas. Text mining for opinion mining, feature extraction, sentiment, prediction, and trend analysis in application areas such as search engines, customer relationship management systems, filter e-mail, product proposal analysis, fraud detection, and social media analysis. Use. Text mining is based on a variety of advanced techniques derived from statistics, machine learning, and linguistics. Text mining uses technology to find patterns and trends in "Unstructured data", which is more commonly due to, but is not limited to, textual information. The goal of text mining is to be able to process large amounts of text data to extract "high quality" information that will help you gain insights into the specific scenarios to which text mining applies. Text mining has many applications such as concept extraction, sentiment analysis, and summarization. Text mining is based on a variety of advanced techniques derived from statistics, machine learning, and linguistics.

Text mining uses technology to find patterns and trends in "Unstructured data", which is more commonly due to, but is not limited to, textual information. The goal of text mining is to be able to process large text data and extract "high quality" information. This helps to provide insight into the specific scenario to which text mining applies. Text mining has numerous applications such as extraction of concepts, sentiment analysis, and summarization.

2. Literature Survey

Research for Mobile On-Screen Keyboard

In recent years, the growth of technology has influenced the way people communicate. Writing, as well as speaking and listening, plays an important role in communication. In particular, the proliferation of mobile hardware has changed devices and changed applications. The purpose of this survey is to investigate student usage and preferences for mobile on-screen keyboards. In this regard, we looked at 20 popular keyboard programs offered for the IOS and Android platforms. A question pool containing various items about the functionality of these programs was created for your research. The research developed consists of 27 items, 26 of which are 3-point Likert type and 1 semi-structured type. In the survey, 238 participants participated in the survey. For of the collected data, frequency and percentage were calculated to determine participant preferences. The results conclude that students used the keyboard primarily for app applications. The most preferred features are local character support, commonly used words, sending voice messages, vertical or horizontal use, and key response time, while swipe keyboard input is for participants. It was the least desirable. Communication was once, so it has been done in many languages related to its origin. According to the Turkish Language Education Dictionary (2018), Communication is a communication of emotions that conveys information to the hearts of others as much as possible. Although the forms and channels of communication have changed, people have used Communication to express themselves according to their environment. Since they have spent most of their daily life on communication-based activities, they have tried to develop and effectively use these communication skills.

Natural Language Processing

Natural Language Processing (NLP) is a field of computer science and artificial intelligence that deals with natural language interactions between computers and humans. The ultimate goal of NLP is to enable computers to understand languages as we do. This is the power behind virtual assistants, speech recognition, sentiment analysis, automated text summarization, machine translation, and more. In this post, I'll explain the basics of natural language processing and elaborate on some of its techniques. You will also learn how NLP benefits from recent advances in deep learning. Natural Language Processing (NLP) is an interface between computer science, linguistics, and machine learning. This area focuses on communication between computers and humans in natural languages, and NLP aims to make computers understand. Natural Language Processing (NLP) is a field of computer science and artificial intelligence that deals with natural language interactions between computers and humans. The ultimate goal of NLP is to enable computers to understand languages as we do. This is the power behind virtual assistants, speech recognition, sentiment analysis, automated text summarization, machine translation, and more. In this post, I'll explain the basics of natural language processing and elaborate on some of its techniques. You will also learn how NLP benefits from recent advances in deep learning. Natural Language Processing (NLP) is an interface between computer science, linguistics, and machine learning. This area focuses on communication between computers and humans in natural languages, and NLP aims to make computers understand.

3. Conceptual process and framework

Text mining analysis involves several advanced process steps. These steps are primarily influenced by the fact that, from a computer perspective, the text is a collection of unstructured words. Text mining Analysts usually start with a series of very heterogeneous input texts. Therefore, the first step is to import these texts into the preferred computer environment (R in this case). At the same time, it is important to organize and structure the text so that it can be accessed consistently. Once the text is organized in the repository, the second step is to clean up the text. This includes pre-processing the text to get the appropriate representation for later analysis. This step may include reformatting the text (such as removing spaces). Remove stopwords or stemming procedures. Third, the analyst must be able to convert the preprocessed text to a structured format so that it can actually be calculated. For "classical" text mining tasks, this usually means creating a so-called term-document matrix. This is the most common form for representing text for calculations.

Analysts can now manipulate and calculate text using standard statistical and data mining techniques such as clustering and classification methods. This fairly typical process model highlights the key steps that require support from the text mining infrastructure. The text mining framework should provide the ability to manage text documents, abstract the process of manipulating documents, and facilitate text formatting by using heterogeneous. Therefore, you need a conceptual entity similar to a Database that maintains and manages text documents in the usual way. This entity is called a collection of text documents or corpus. Text documents exist in a variety of file formats and locations, including compressed files on the Internet and locally stored text files with additional annotations, providing a standardized interface for accessing document data. An encapsulation mechanism is required. This feature is summarized in the so-called source. In addition to the actual text data, many modern file formats provide the ability to annotate text documents (e.g., XML with special tags). H. Metadata is available that can further explain and enhance the textual content and provide valuable insights into the structure or additional concepts of the document. Also, additional metadata may be created during the analysis. Therefore, the framework must be able to support the use of metadata at both the document level (e.g.). Example: A brief summary or description of the selected document) and collection level (Example: Classification tag for the entire collection).

In addition to the text document data infrastructure, the framework must provide tools and algorithms to process the document efficiently. This means that the framework needs features to perform common tasks such as removing spaces, stemming, and removing stopwords. Functions that affect a collection of text documents are called transforms. Another important concept is filtering. This basically involves applying the predicate function to the collection to extract the pattern of interest. A surprisingly difficult operation is merging a collection of text documents. Merging a set of documents is easy, but intelligently merging metadata requires more advanced processing. This is because storing metadata from different sources in sequential steps inevitably results in a hierarchical tree-like structure. The challenge is to efficiently maintain these links and subsequent searches to search a large collection of documents. A realistic scenario of text mining uses hundreds of thousands of documents, from at least hundreds of text documents up to. That is, the compact storage of documents in the document collection is related to reasonable RAM usage. A simple approach is to keep all documents in memory once read and even shut down a fully RAM-equipped system with a document collection of thousands of texts. Documents soon. However, a simple database-oriented mechanism can already avoid this situation.

4. Application

Cloud-Base Evaluation

One of the easiest methods of analysis in text mining is based on the numerical evaluation. This means that the most frequently occurring terms in the text are classified as important. Although this approach is simple, it is widely used in text mining because it is easy to interpret and requires little computational effort (Davi et al. 2005). First, create a term-document matrix for the raw dataset. Rows correspond to document IDs and columns correspond to terms. Matrix elements contain specific frequencies.

R> crudeTDM <- TermDocMatrix(crude, control = list(stopwords = TRUE))

Then use the term document analytic function, which turns a term that appears at least times. For example, you can select these terms from the rough terms that appear at least 10 times-Document Matrix.

R> (crudeTDMHighFreq <- findFreqTerms(crudeTDM, 10, Inf))

Conceptually, we interpret a term as important according to a simple counting of frequencies. As we see the results can be interpreted directly and seem to be reasonable in the context of texts on crude oil (like OPEC or Kuwait). We can also apply this function to see an excerpt (here the first 10 rows) of the whole (sparse compressed) term-document matrix.

Internally we compute the correlations between all terms in the term-document matrix and filter those out higher than the correlation threshold.

Simple text clustering

In this section we will discuss classical clustering algorithms applied to text documents. For this we combine our known acq and crude data sets to a single working set ws in order to use it as input for several simple clustering methods.

R> ws <- c(acq, crude)
R> summary(ws)

Hierarchical clustering

Here is a hierarchical clustering (Johnson 1967; Hartigan 1975; Anderberg 1973; Hartigan 1972) and a text document. Of course, the choice of distance measure has a significant effect on the results of the hierarchical clustering algorithm. Common similarities in text mining are metric distance, chord measure, Pearson correlation, and extended Jaccard similarity (Strehl et al. 2000). Use the similarity provided by dist from package Proxy (Meyer and Buchta 2007) in the tm package, and use the general user-defined distance function dissimilarity () for term-document matrices. Therefore, you can easily use cosine as a rough term-document matrix distance measure.

R> dissimilarity(crudeTDM, method = "cosine")

Our dissimilarity function for text documents takes as input two text documents. Internally this is done by a reduction to two rows in a term-document matrix and applying our custom distance function. For example, we could compute the Cosine dissimilarity between the first and the second document from our crude collection.

R> dissimilarity(crude[[1]], crude[[2]], "cosine")

K-means clustering

Using the term-document-matrix representation of news articles, the classic k-means algorithm (Hartigan and Wong 1979; MacQueen 1967) is used to provide valid input for to existing methods in R. Here's how. Perform classical linear k-mean clustering at k = 2 (since we concatenated the two topic sets acq and the raw working set, we can see that only two clusters are valid values).

R> wsKMeans <- kmeans(wsTDM, 2)

and present the results in form of a confusion matrix. We use as input both the clustering result and the original clustering according to the Reuters topics. As we know the working set consists of 50 acq and 20 crude documents

R> wsReutersCluster <- c(rep("acq", 50), rep("crude", 20))

Using the function cl_agreement() from package clue (Hornik 2005, 2007a), we can compute the maximal co-classification rate, i.e., the maximal rate of objects with the same class ids in both clusterings—the 2-means clustering and the topic clustering with the Reuters acq and crude topics after arbitrarily permuting the ids:

R> cl_agreement(wsKMeans, as.cl_partition(wsReutersCluster), "diag")

This means that the results of k-means clustering can recover about 70% of human clustering. See Karatzoglou and Feinerer (2007) for a practical example of text clustering in a tm package containing hundreds of documents. This shows that text clustering works very well with the right set of documents.

5. Conclusion

Introduced a new framework for R text mining applications via the tm package. It provides capabilities for managing text documents, abstracts the document editing process, and facilitates the use of heterogeneous text formats in R. The package includes database back-end support to minimize the memory footprint. Enhanced metadata management is implemented in a collection of text documents to facilitate the use of large document sets enhanced with metadata. This package includes Reuters-21578 dataset, Reuters Corpus Volume 1 dataset, Gmane RSS feeds, email, and some traditional file formats (plain text, CSV text, PDF, etc.) native to processing. Support is included. The package is modular and can easily integrate new file formats, readers, conversions, and filtering operations to extend the data structure and algorithms to meet custom requirements. Tm provides easy access to pre-processing and manipulation mechanisms such as, such as whitespace removal, stemming, and conversion between file formats (eg. from Reuters to plain text). In addition, you can use a common filtering architecture to filter documents according to specific criteria and perform full-text searches. This package supports the export of document collections in the term document matrix commonly used in text mining literature. This makes it easy to integrate existing methods of classification and clustering. The tm not only uses the technology available in R, but also has a wide range of interfaces with other open source toolkits such as Weka and openNLP that provide additional methods such as tokenization, stemming, and lexical recognition. We already support and cover text mining methods. They provide Speech tagging However, there are still many areas that can be further improved. B. A method that is more common in linguistics such as latent semantic analysis. I am thinking of a better integration of the tm and lst packages. Another important technique addressed in in the future is the efficient processing of very large conceptual document matrices. In particular, we are working on a memory-efficient clustering technique in R to handle high-dimensional sparse matrices, as seen in a large text mining case study. With continued research efforts at to analyze large datasets and the use of sparse data structures, tm will be one of the first s to take advantage of new technologies. Finally, we will continue to add read capabilities to procure classes in a common data format.

6. References

- Adeva JJG, Calvo R (2006). "Mining Text with Pimiento." IEEE Internet Computing, 10(4), 27–35. ISSN 1089-7801. doi:10.1109/MIC.2006.85.
- Bates D, Maechler M (2007). Matrix: A Matrix Package for R. R package version 0.999375-2, URL http://CRAN.R-project.org/package=Matrix.
- Bierner G, Baldridge J, Morton T (2007). "OpenNLP: A Collection of Natural Language Processing Tools." URL http://opennlp.sourceforge.net/.
- Cavnar W, Trenkle J (1994). "N-Gram-based Text Categorization." In "Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval," pp. 161–175. Las Vegas.