International Journal of Multimedia and Ubiquitous Engineering
Vol. 4, No. 2, April, 2009

Knowledge Discovery using Text Mining: A Programmable Implementation on Information Extraction and Categorization

Atika Mustafa, Ali Akbar, and Ahmer Sultan
National University of Computer and Emerging Sciences-FAST, Karachi Pakistan
atika.mustafa@nu.edu.pk, aliakbaryousuf@gmail.com, cyphorous@gmail.com

Abstract

Textual data in electronic documents today around the world have no doubt brought forward all the information one could need and as data banks build up worldwide, and access gets easier through technology, it has become easier to overlook vital facts and figures that could bring about groundbreaking discoveries. This research paper discusses in detail an implementation of Information Extraction and Categorization in the text mining application that we have implemented. To extract terms from the document we have used modified version of Porter's Algorithm for inflectional stemming. For calculating term frequencies for categorization, we have used a domain dictionary for 'Computer Science' domain.

Keywords: *Knowledge management, Text mining, Categorization, Information Extraction, Unstructured data*

1. Introduction

Text mining — also called intelligent text analysis, text data mining, or knowledge discovery in text — uncovers previously invisible patterns in existing resources [1].

To perform analysis, decision-making, and knowledge management tasks, information systems use an increasing amount of unstructured information in the form of text. This data influx, in turn, has spawned a need to improve the text mining technologies required for information retrieval, filtering, and classification [1].

People who are involved in doing research can systematically analyze multiple research papers, e-books and other documents, and then swiftly determine what they contain. For example in an HR department, a CV which matches a particular job specification from amongst a million CV in a database may not be the simplest of tasks.

All this doesn't just make it easy to determine what to focus in a particular document, but also where to find it and how to important it is as compared to other similar documents. With its extensible knowledge base and generic algorithms, it can be brought to use for just about any field or industry.

Text Mining itself is not a function, it comprises of various functions which when combined can be called Text Mining functions. The main functions include Searching, Information Extraction (IE), Categorization, Summarization, Prioritization, Clustering, Information Monitor and Question and Answers [2].

This paper does not only describe and explain text mining and all its details, but it also provides a comprehensive discussion on an application which has been developed. All the algorithms which have been used have been described in full detail along with their performances.

2. Information Extraction

The general purpose of Knowledge Discovery is to "extract implicit, previously unknown, and potentially useful information from data" (Frawly 1991). Information Extraction IE mainly deals with identifying words or feature terms from within a textual file. Feature terms can be defined as those which are directly related to the domain.

Figure 1. A layered model of the Text Mining Application

These are the terms which can be recognized by the tool. In order to perform this function optimally, we had to look into few more aspects which are as follows:

2.1 Stemming

Stemming refers to identifying the root of a certain word. There are basically two types of stemming techniques, one is inflectional and other is derivational. Derivational stemming can create a new word from an existing word, sometimes by simply changing grammatical category (for example, changing a noun to a verb) [Wikipedia]. The type of stemming we were able to implement is called Inflectional Stemming. A commonly used algorithms is the 'Porter's Algorithm' for stemming. When the normalization is confined to regularizing grammatical variants such as singular/plural or past/present, it is referred to inflectional stemming [2].

To minimize the effects of inflection and morphological variations of Words (stemming), our approach has pre-processed each word using a provided version of the Porter stemming algorithm with a few changes towards the end in which we have omitted some cases.

e.g. apply – applied – applies
print – printing – prints – printed

In both the cases, all words of the first example will be treated as 'apply' and all words of the second example will be treated as 'print'.

2.2 Domain dictionary

In order to develop tools of this sort, it is essential to provide them with a knowledge base. A collective set of all the 'feature terms' is the Domain dictionary (our source was www.webopedia.com). The structure of the Domain dictionary which we implemented consisted of three levels in the hierarchy. Namely, Parent Category, Sub-category and word.

Parent categories define the main category under which any sub-category or word falls. A parent category will be unique on its level in the hierarchy. Sub-categories will belong to a

certain parent category and each sub-category will consist of all the words associated with it. As an example, consider the following

Table 1.Structure of the Domain Dictionary

Parent Category	Sub-Category	Words
Hardware	Data Storage	*Grabber*
	Input devices	*Light pen*
	Modems	*Joystick*
	Motherboards	*Contact image sensor*
	Networking	*Digital camera*

Table 1 is an example that shows how we identify words which belong to the Parent Category 'Hardware' and Sub-category 'Input Devices'.

2.3 Exclusion List

A lot of words in a text file can be treated as *unwanted noise*. To eliminate these, we devised a separate file which includes all such words. These include words such as the, a, an, if, off, on etc.

3. Categorization

Categorization is a core function of text mining. Categorization helps to identify as to exactly which category of the domain in use, a certain text file relates to. The categorization that we implemented requires extensive tokenization. Tokenization refers to the extraction of feature terms in the document.

With categorization techniques, systems can assign previously unseen documents to the most suitable category available, based on a particular taxonomy or topic [1].

There are two hash tables that handle the categorization. One hash table handles the aggregated counts of the candidate categories found because of the matched tokens, the other hash tables manages the frequencies of the matched tokens. Matched tokens are the ones that are found in the document and are also in the domain dictionary. First is the hash table managing the token frequencies, it holds the frequency of the found tokens in the document. When that process is complete, the hash table for category aggregate is populated using the frequency hash table.

For a token in the frequency hash table, it finds the parent categories in the domain dictionary. For a single token there can be multiple parent categories. All of them are candidates to be the category of the document. This aggregate hash table holds all the candidate categories initialized by the frequency of its child token from the frequency hash table. If any other token in the frequency hash table a parent category is found which is already stored in the aggregate hash table, then it updates the frequency it had by adding the frequency of the new token.

This approach was used because of a certain number of situations, but the most important one was that a single word in the domain dictionary can fall under multiple categories.

Once we have the results the user can easily determine that the document mostly contains information related to the two main categories listed and a little bit of information related to the categories which come under the heading “Sub-categories” in the output screen.

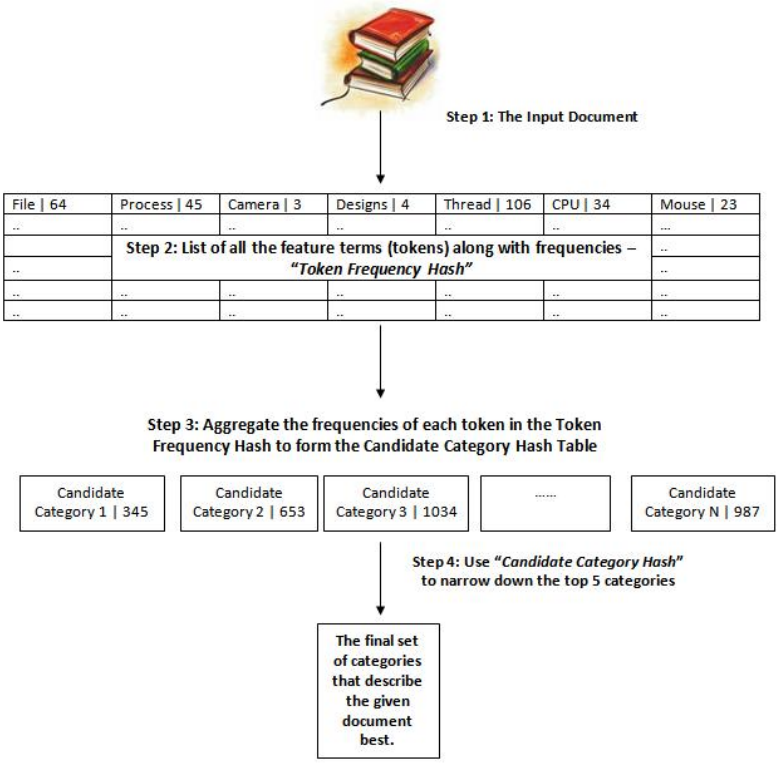

Figure 2. Extraction of feature terms and calculation of frequencies

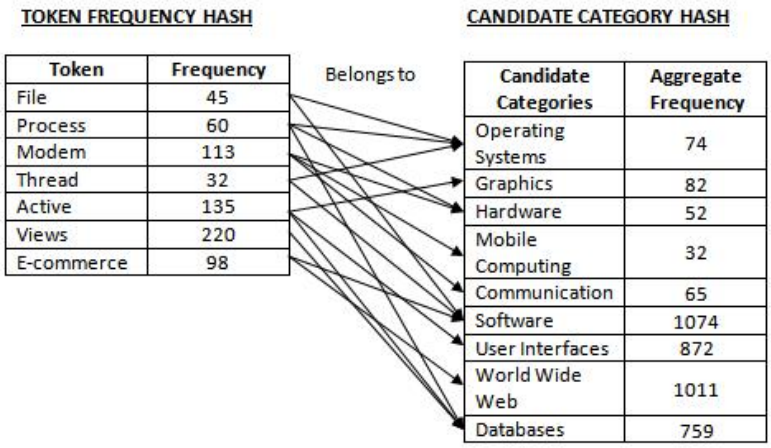

Figure 3. Frequency calculation in categorization

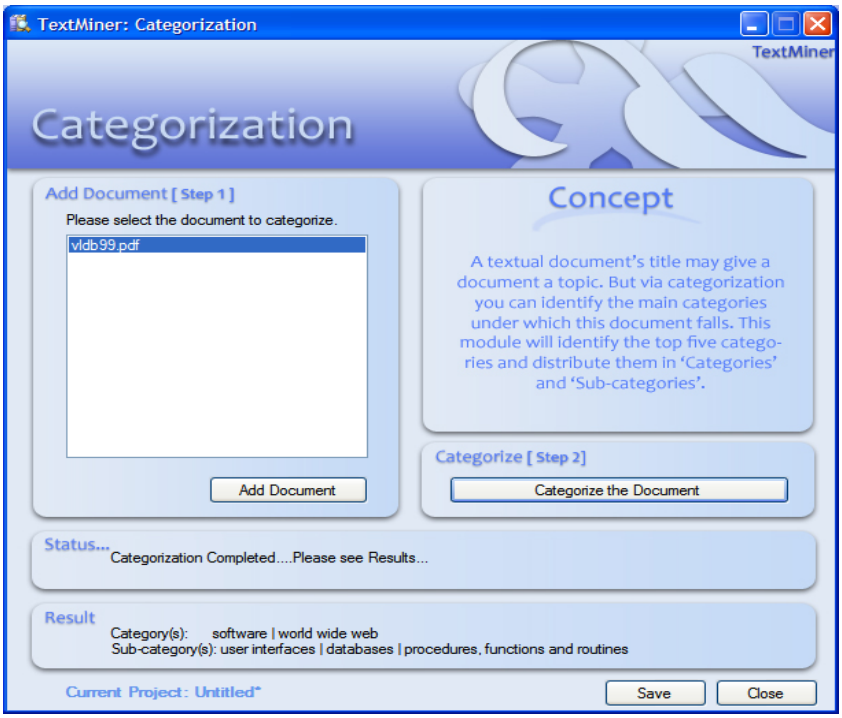

Figure 4. Categorization results in our text mining tool

Figure 4 shows a result we obtained by categorizing a research paper "Active Views for Electronic Commerce [6]." Figure 2 can be referred to get an idea how the algorithm worked. The algorithm deduced the Electronic Commerce directly related to the World Wide Web and since the paper discusses details on a specific application, it related it to Software. Furthermore, it discusses some functions and routines on how the active can be implemented. The paper took about 35 to 40 seconds depending upon the available resources of the PC on which our tool was running.

4. Limitations

The application currently provides accurate results only documents which are related to the Computer Science field.

The domain dictionary is populated with very little words as compared to the amount words it should actually have. Currently it has about 8000 terms; we are planning to expand it with another 40,000 terms very soon. This will result in excellent performance of the application in terms of accuracy and precision.

5. Future Work

'Self Learning' functionality should be introduced. This will help the application to learn on its own. One approach to this is that if it gets frequent words again and again, it could take some input from the user and add that particular word under the correct category.

Information extraction remains a challenging problem with many potential avenues for progress. One approach is to use active learning methods to decrease the amount of training

data that must be annotated by selecting only the most informative sentences or passages to give to human annotators [5].

Clustering is another really important text mining functionality that should be incorporated into the tool.

Even though the tool does answer factoid questions currently, there is vast room for improvement in its results. Answering complex questions is the extreme of text mining, yet achievable.

In the end, we would like to add that giving this text mining tool a web interface will be a very big step forward.

6. Conclusion

In this paper we have discussed what text mining is and its categorization process. We have further discussed the challenges faced in the implementation of these functions. Text mining works on unstructured data (textual files). The domain dictionary which defines the set of terms (of a certain field, in our case Computer Science) consists of all feature terms is the essence of such mining tools. A lot of work can be done to further improve and extend this implementation.

References

[1] Juan José García Adeva and Rafael Calvo, "Mining Text with Pimiento", *University of Sydney*

[2] Text Mining Application Programming by Manu Konchadi. Published by Charles River Media. ISBN: 1584504609

[3] Automated Concept Extraction From Plain Text. Boris, GARAGe Michigan State University, East Lansing MI 48824.

[4] Dr. Antoine Spinakis; Asanoula Chatzimakri, "Comparison Study of Text Mining Tools".

[5] Raymond J. Mooney and Razvan Bunescu, "Mining Knowledge from Text Using Information Extraction", University of Texas at Austin

[6] Tova, Milo, "Active views for Electronic commerce". Paper number: EUROPE64.

[7] Martin Rajman, "Text Mining – Knowledge extraction from unstructured textual data". CS Dept, Swiss Federal Institute of Technology