International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
International Conference & workshop on Advanced Computing 2013 (ICWAC 2013) – www.ijais.org

WSD Tool for Ontology-based Text Document Classification

Nazia Ilyas Baig
Student Information Technology,VESIT
Chembur, Mumbai, India

Gresha Bhatia
Deputy HOD, Computer Engineering, VESIT
Chembur, Mumbai, India

ABSTRACT

The classification of document is required to extract relevant information from the huge set of documents. There are various traditional approaches which are being used satisfactorily, but even such approaches or techniques are not enough. These traditional approaches require training sets of pre-classified documents in order to train the classifier. These approaches mainly depend only on 'bag of words', this representation used is unsatisfactory as it ignores possible relations between terms. When training set is not available, ontology provides us with knowledge that can be efficiently used for classification without using training sets. Ontology expresses information in the document form of hierarchical structure. For classifying the documents using ontology we need to define the class or the concepts to categorize the document. Here we use WordNet to capture the relations between the words. Also it is seen that WordNet alone is not sufficient to remove Word Sense Disambiguation (WSD). So in our approach we use Lesk algorithm to deal with the WSD. In this paper, we implement the tool which disambiguates a keyword in the text file. This tool is actually a utility where the input will be a text file and the utility will process the input file to give the best sense for the most occurring keyword in the file. There are various modules for achieving this. This keyword is further used for mapping with concepts to create ontology. The ontology will have classes/concepts defined for all the files in the corpus. Our approach is leveraging the strengths of ontology, WordNet and Lesk Algorithm for improving text document classification.

General Terms

Ontology-based Text document classification, concepts, and keywords.

Keywords

Ontology, ontology-based text document classification, concepts, sense, WSD, WordNet, LESK algorithm.

1. INTRODUCTION

Text document classification plays an important role in providing intuitive navigation and browsing mechanisms by organizing large sets of documents into a small number of meaningful clusters. The bag of words representation used for these clustering methods is often unsatisfactory as it ignores relationships between important terms that do not occur literally [1]. When training set is not available, an alternative approach must be available. Another approach must use the available knowledge such as named entities, relationships between them. So, ontology provides us with such knowledge that can be efficiently used for classification without using training sets. Ontology expresses information in the document form of hierarchical structure. Ontology-based text document classification involves determining document features that represent the document most accurately and helps classify them into most appropriate categories. Ontology-based text document classification provide the conceptual instances within the ontology which have inherent semantics, and a close relationship with the class representatives of the classification scheme .Thus Ontology-based text document classification is a classification of documents using ontologies as category definition [2]. Ontology-based document classification provides [3]:

- Classification based on knowledge from ontology
- Classification that do not require training set
- Use of semantic information for categorization
- Explore role of semantic associations in text categorization
- Incorporation of user interest (context) into categorization

Ontology-based text document classification involves determining document features that represent the document most accurately and helps classify them into most appropriate categories. Ontology-based text document classification provides the conceptual instances within the ontology which have inherent semantics and a close relationship with the class representatives of the classification scheme [4]. In [5] they have proposed a general framework for text classification, which is applicable across different domains. But this approach does not deal with WSD. We are resolving this limitation in our approach.

Here in this paper we create our ontology for classification of the text documents in the corpus. For this we use documents describing the company's profile. We created the corpus from Yahoo! Finance.

As discussed earlier for making the classification more effective we implement the tool which disambiguates a keyword in the text file for our proposed system. This tool is actually a utility where the input will be a text file and the utility will gradually pre-process the input file to give the best sense for the most occurring keyword in the file. There are various modules for achieving this. The keyword is further used for mapping with concepts to create ontology. The ontology will have classes/concepts defined for all the files in the corpus. Figure 1 is our proposed system [6]. It has following modules:

1. *Pre-processing module*
   a. Filter
   b. Converting plurals to singulars
   c. Term/keyword formation
2. *Word sense disambiguation (WSD) module* and
3. *Ontology module*
   d. Mapping terms to concept
   e. Ontology formation

Each module mentioned above have a significant role as these step by step forms the keyword and then disambiguates the keyword.

In section 2 we elaborate the pre-processing module along with its sub-modules. In section 3 we discuss the WSD module followed by ontology module in section 4. In section 5 we propose a plan to evaluate the classification process. Conclusion and future scope being discussed in the next successive sections 6 and 7 respectively.

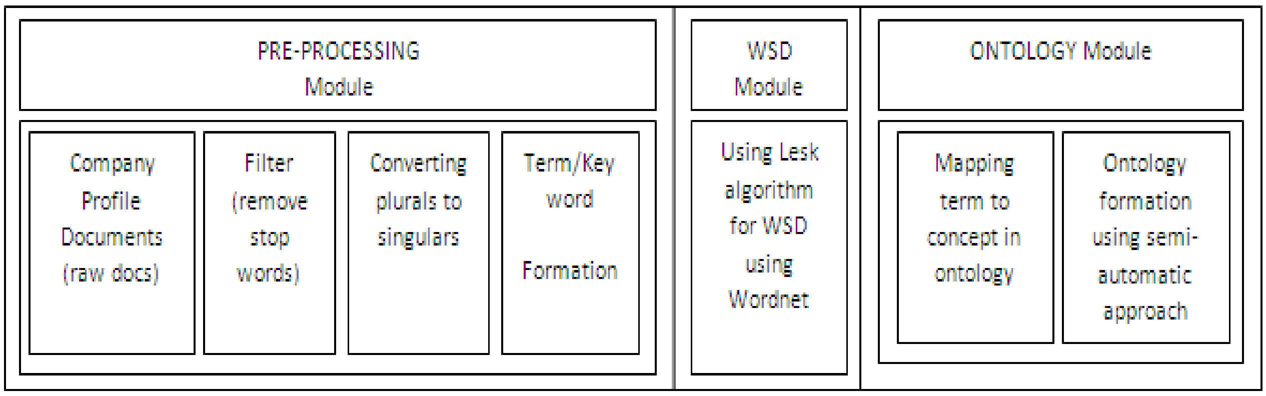

PRE-PROCESSING Module
Company Profile Documents (raw docs)
Filter (remove stop words)
Converting plurals to singulars
Term/Key word Formation
WSD Module
Using Lesk algorithm for WSD using Wordnet
ONTOLOGY Module
Mapping term to concept in ontology
Ontology formation using semi-automatic approach

Fig 1: The Proposed System

2. THE PRE-PROCESSING MODULE

This is the first module. This is a very important module as this narrows our choices for disambiguating a keyword. In this module we use the WordNet [7] for converting the plurals and for forming the keywords. The sub-modules in this module will help us in disambiguating the keyword used in classifying the document. As seen above first module has three sub-modules:

a) the *filter sub-module*; b) the *converting plurals to singulars. sub-module; and* c) the *term/keyword formation sub-module*

.

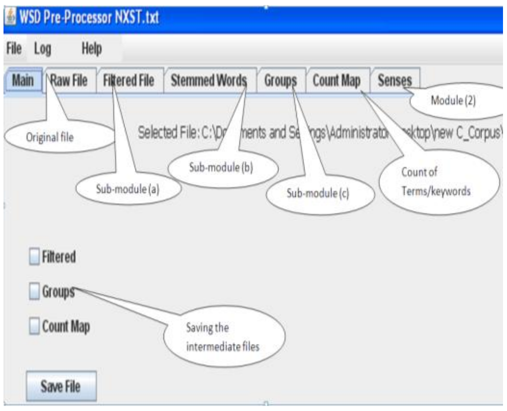


Figure 2: Main frame of the tool

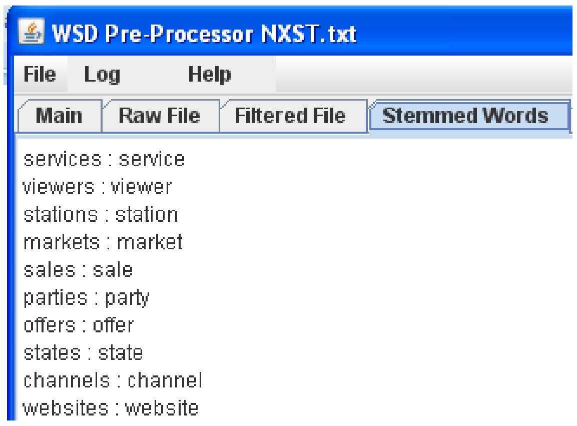
This is what we have automated as our tool for WSD. Figure 2 shows the main frame of our tool. We have taken the file *NXST.txt* as an input here. Various tabs have been provided to view the intermediate result of the process of WSD. It has a *save file* button to save these respective files for reference. We have discussed the three sub –modules in 2.1, 2.2 and 2.3 respectively.

2.1 The Filter sub-module (a)

It filters the file from the unwanted stuff, i.e., stop words. Stop words are the words which do not change their meaning, for example "*a, and, but, how, or,* and *what*". Here the stop words are removed from the documents. The stop-words does not contribute in finding sense of the keyword so all stop words which are present in file are discarded from the file using stop words look up list. The output of this module is the filtered file and can be seen in the *Filtered File* tab.

2.2 The Converting Plurals to Singulars sub-module (b)

In this sub-module we partially stem the filtered file .i.e.; we convert the plurals to singulars. The plural word is reduced to its base form. The module is useful as it helps us to maintain the count of the keyword which this is required in the other module. For example in figure 3 there keywords converted to singular form. In this module we use WordNet to check if the singular form of the reduced word is available in the dictionary.



```
services : service
viewers : viewer
stations : station
markets : market
sales : sale
parties : party
offers : offer
states : state
channels : channel
websites : website
```

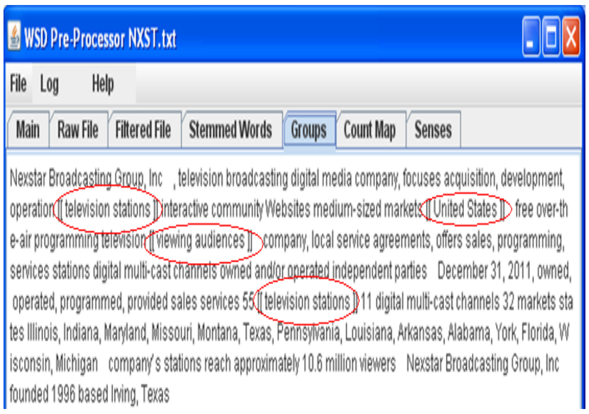
Figure 3: keywords converted to singular form

2.3 The Term/Keyword Formation sub-module (c)

A keyword or a term can be single word or multiple words [8]. The term/keyword formation module plays a very significant role in disambiguation. For example 'station' has four sense/meaning in the WordNet and 'television' has three sense/meaning but when we combine these two words to form a single keyword 'television station' it will have only one sense 'station for the production and transmission of television broadcasts'. As said again in this module we use WordNet to form the keywords by grouping the words. We form the keywords by making pair with the neighbors and check if there exist a sense of the pair in the WordNet. The pseudo code is as below:

```
formCombinationsLR()//left to right

Create new ArrayList
Initialise len to itms.length
For cnt is 0, cnt is less than len,
cnt increments by 1
      Create new StringBuffer
      For i is 0, i is less than cnt,
i increments by 1
            Call method sb.append
with position i in itms plus "      "
      EndFor
      Call method list.add with
sb.toString
EndFor
Return list


formCombinationsRL()//right to left

Create new ArrayList
Initialise len to itms.length
For cnt is 0, cnt is less than len,
cnt increments by 1
      Create new StringBuffer
      For i is cnt, i is less than
len, i increments by 1
            Call method sb.append
with position i in itms plus "      "
      EndFor
      Call method list.add with
sb.toString
EndFor
Return list
```

Figure 4 has the *groups* tab where the terms/keywords formed are displayed. For this input file the keywords formed are *television stations, United States* and *viewing audiences*. To differ these keywords we have surrounded them by "[[ ]]" braces.



Nexstar Broadcasting Group, Inc , television broadcasting digital media company, focuses acquisition, development, operation [[television stations]] interactive community Websites medium-sized markets [[United States]] free over-the-air programming television [[viewing audiences]] company, local service agreements, offers sales, programming, services stations digital multi-cast channels owned and/or operated independent parties December 31, 2011, owned, operated, programmed, provided sales services 55 [[television stations]] 11 digital multi-cast channels 32 markets states Illinois, Indiana, Maryland, Missouri, Montana, Texas, Pennsylvania, Louisiana, Arkansas, Alabama, York, Florida, Wisconsin, Michigan company's stations reach approximately 10.6 million viewers Nexstar Broadcasting Group, Inc founded 1996 based Irving, Texas

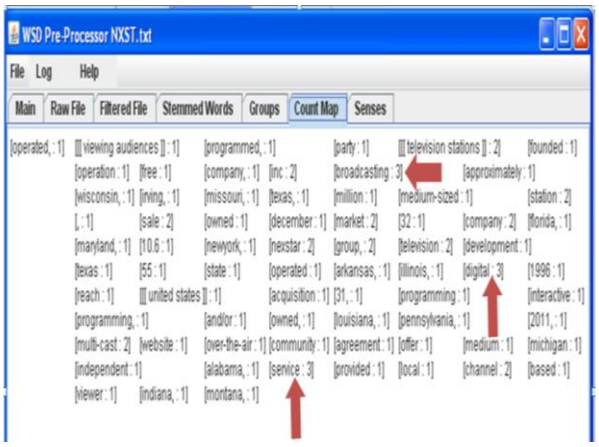
Figure 4: *Groups* tab displaying the terms/keywords formed

3. THE WSD MODULE

Once the keywords are formed the next step is to find the most occurring keyword in the file. If the file has a particular keyword occurring for maximum number of times it means that the file is related to that keyword. So before actually disambiguating the keyword we find the count of each keyword occurrences in the file. Our tool explicitly disambiguates the keyword with maximum count also a provision is made so that the user can ask to disambiguate another keyword with next highest count. This provision is made because sometimes the company's name itself may have the maximum count and this company name is not in the stop word list and is not filtered. So for our simplicity we can go through the count map and check the next keyword with maximum count. In this module again we have used the WordNet dictionary to find the best sense for our keyword using modified LESK algorithm [8]. The brief idea of modified LESK algorithm is given in [6]. The pseudo code is as below:

```
getBestSense()

Create new HashMap
If word is equal to null
      Return store
EndIf
TODO Implement the Code logic
Set of Synset toset synN to
getSynsets with neighbour
If synW is equal to null
      Return store
```

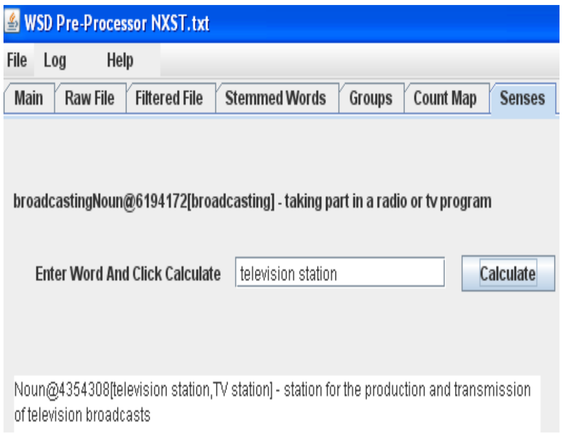
```
EndIf
For (Synset syn1 : synW)
      Initialise i to 0
      For (Synset syn2 : synN)
            I += getScore(syn1,
syn2);
            Display the msg;
      EndFor
      Call method store.put with
syn1.getDefinition
EndFor
Return store

getScore()// for syn1 and syn2

Initialise score to 0
Initialise synDef to
syn2.getDefinition
Create new StringTokenizer
While st.hasMoreTokens
      Initialise tmp to st.nextToken
      If RoughDict.getInstance
            continue;
      EndIf
      If synDef.contains with
tmp.toLowerCase
            Score += 1;
      EndIf
EndWhile
For (String tmp :
syn1.getWordForms())
            If synDef.contains with
tmp.toLowerCase
            Score += 1;
      EndIf
EndFor
Return score
```


WSD Pre-Processor NXST.txt
File Log Help
Main Raw File Filtered File Stemmed Words Groups Count Map Senses

```
[operated, : 1] [[[viewing audiences]] : 1] [programmed, : 1]  [party : 1] [[[television stations]] : 2] [founded : 1]
   [operation : 1] [free : 1] [company, : 1] [inc : 2] [broadcasting : 3] [approximately : 1]
   [wisconsin, : 1] [irving, : 1] [missouri, : 1] [texas, : 1] [million : 1] [medium-sized : 1] [station : 2]
   [, : 1] [sale : 2] [owned : 1] [december : 1] [market : 2] [32 : 1] [company : 2] [florida, : 1]
   [maryland, : 1] [10.6 : 1] [newyork, : 1] [nexstar : 2] [group, : 2] [television : 2] [development : 1]
   [texas : 1] [55 : 1] [state : 1] [operated : 1] [arkansas, : 1] [illinois, : 1] [digital : 3] [1996 : 1]
   [reach : 1] [[[united states]] : 1] [acquisition : 1] [31, : 1] [programming : 1] [interactive : 1]
   [programming, : 1] [and/or : 1] [owned, : 1] [louisiana, : 1] [pennsylvania, : 1] [2011, : 1]
   [multi-cast : 2] [website : 1] [over-the-air : 1] [community : 1] [agreement : 1] [offer : 1] [medium : 1] [michigan : 1]
   [independent : 1] [alabama, : 1] [service : 3] [provided : 1] [local : 1] [channel : 2] [based : 1]
   [viewer : 1] [indiana, : 1] [montana, : 1]
```

Figure 5: *Count map* of the keywords

In figure 5 we have the count map which displays the count of keywords. For the file *NXST.txt* we have *broadcasting*, *digital* and *services* as keywords with max count as *3*. The tool takes the keyword with max count value and disambiguates it.


WSD Pre-Processor NXST.txt
File Log Help
Main Raw File Filtered File Stemmed Words Groups Count Map Senses
Calculate

broadcastingNoun@6194172[broadcasting] - taking part in a radio or tv program

Enter Word And Click Calculate: television station

Noun@4354308[television station,TV station] - station for the production and transmission of television broadcasts

Figure 6: Disambiguation of keywords.

The file has three keywords with max count and out of these three the tool has selected the *broadcasting* keyword as it the first of all. It is shown in figure 6.Also we have a provision to calculate the sense of the other keywords.

4. THE ONTOLOGY FORMATION MODULE

This is the actual module where we define the classes for our text documents (the keywords are also called as concepts just to avoid confusion between the keywords from our tool and keywords from OntoGen we call them as concepts). From the previous *term/keyword formation module* we take reference here.

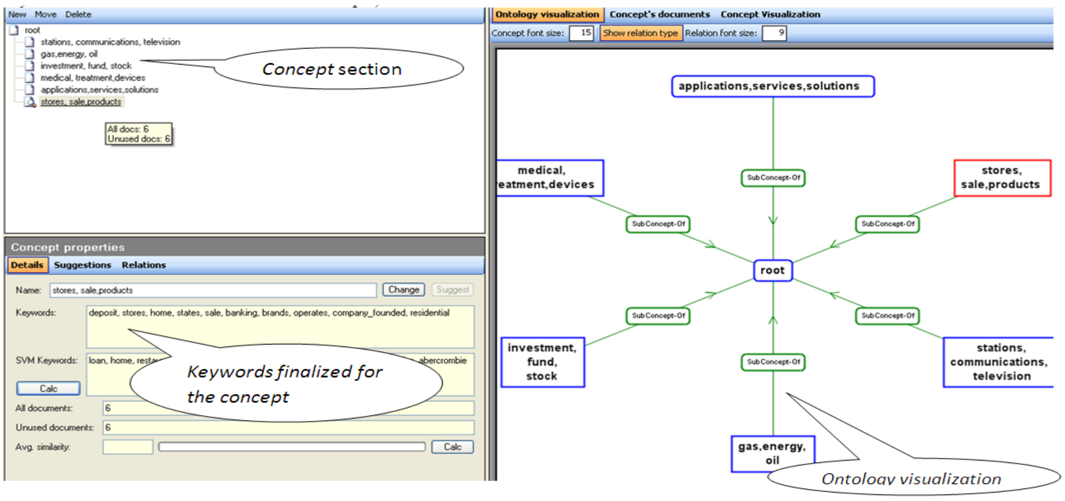
4.1 The Mapping Terms to Concept sub-module (d)

In sub-module *mapping terms to concept* we implicitly map the terms/keywords from (c) with the concepts suggested by the ontology editor 'OntoGen'. OntoGen is a semi automatic ontology editor where we can give our inputs when creating ontology and this is called semi-automated creation of ontology in OntoGen [9]. It is where we actually map the keywords generated from our tool to the concepts suggested by OntoGen.

4.2 The Ontology Formation sub-module (e)

In *ontology formation sub-module* we then finally create our ontology using the concepts in the previous module. The figure 7 shows our ontology for the company corpus. The corpus has 31 text documents collected from YAHOO! FINANCE [10]. The left side of the figure shows the hierarchical structure of the ontology and the left down part *concept properties* shows the details of the class or the concept mapped for ontology formation also in this part we can ask query to add the file in that concept required for mapping. The right part shows the *ontology visualization* where root is the root class by default and the actual classes are connected to it by an incoming arrow also showing their relation type to the root class as *Sub Concept of.* Another important part is *concept's document* this is shown in figure 8.

It has the *sim graph i.e.* the similarity graph. This graph indicates how closely the file is related to the class/concept. The red dots indicate the files of the concept which is selected in the *concept* section and the blue dots are the other files. Here we selected the 'gas, energy, oil' concept and all the files under this concept are checked in the *document* section. In this case the files 'qre.txt','cjes.txt','sempra.txt' and 'nee.txt' are checked. The *sim graph* is generated at the bottom of *document* section. The *suggestions* under *concept's properties* allow us to ask query and then add the file to the most related concept. This is the semi-automated creation of the ontology. We also plan to create another fully automated ontology for our corpus and where we skip WSD process. The ontology will be fully automated because we will not ask the query to do any mapping rather will take the ontology generated as it. This ontology will be just for comparison purpose.


New Move Delete
root
stations, communications, television
gas,energy, oil
investment, fund, stock
medical, treatment,devices
applications,services,solutions
stores, sale,products
Concept section
All docs: 6
Unused docs: 6
Concept properties
Details Suggestions Relations
Change Suggest
Keywords finalized for the concept
Calc
Ontology visualization Concept's documents Concept Visualization
Concept font size: 15 Show relation type Relation font size: 9
applications,services,solutions
medical, eatment,devices
stores, sale,products
root
SubConcept-Of
investment, fund, stock
stations, communications, television
gas,energy, oil
Ontology visualization

Name: stores, sale,products

Keywords: deposit, stores, home, states, sale, banking, brands, operates, company_founded, residential

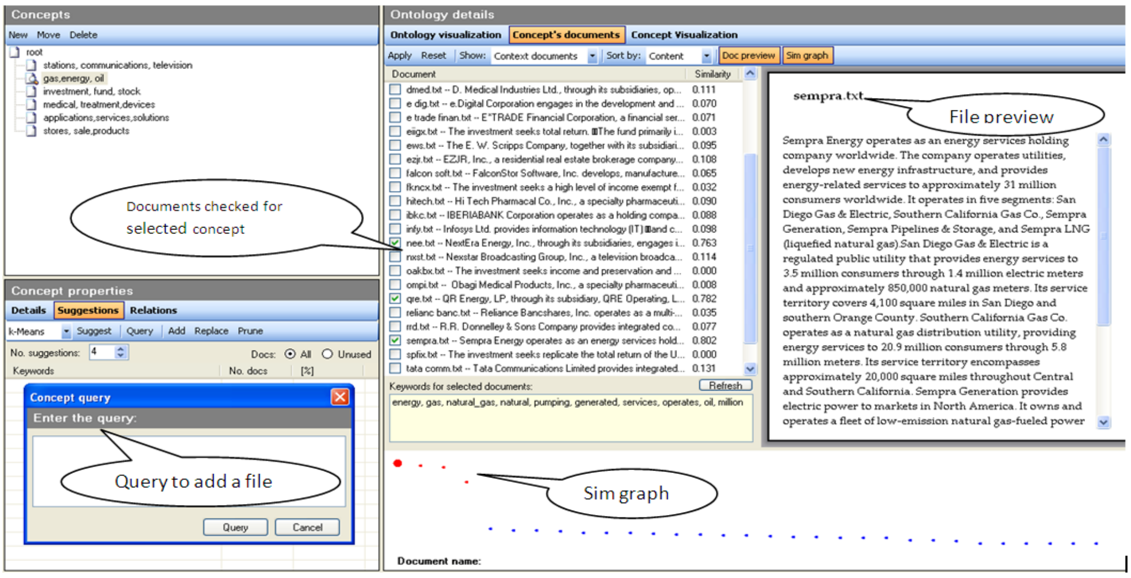
SVM Keywords: loan, home, resta… abercrombie

All documents: 6

Unused documents: 6

Avg. similarity:

Figure 7: Ontology of the proposed system created in OntoGen


Concepts
New Move Delete
root
stations, communications, television
gas,energy, oil
investment, fund, stock
medical, treatment,devices
applications,services,solutions
stores, sale,products
Documents checked for selected concept
Concept properties
Details Suggestions Relations
k-Means Suggest Query Add Replace Prune
Concept query
Enter the query:
Query to add a file
Query Cancel
Ontology details
Ontology visualization Concept's documents Concept Visualization
Apply Reset Doc preview Sim graph
File preview
Refresh
Sim graph

No. suggestions: 4 Docs: (•) All ( ) Unused

Keywords No. docs [%]

Show: Context documents Sort by: Content

Document	Similarity
☐ dmed.txt -- D. Medical Industries Ltd., through its subsidiaries, op...	0.111
☐ e dig.txt -- e.Digital Corporation engages in the development and ...	0.070
☐ e trade finan.txt -- E*TRADE Financial Corporation, a financial ser...	0.071
☐ eiigx.txt -- The investment seeks total return. The fund primarily i...	0.003
☐ ews.txt -- The E. W. Scripps Company, together with its subsidiari...	0.095
☐ ezjr.txt -- EZJR, Inc., a residential real estate brokerage company...	0.108
☐ falcon soft.txt -- FalconStor Software, Inc. develops, manufacture...	0.065
☐ fkncx.txt -- The investment seeks a high level of income exempt f...	0.032
☐ hitech.txt -- Hi Tech Pharmacal Co., Inc., a specialty pharmaceuti...	0.090
☐ ibkc.txt -- IBERIABANK Corporation operates as a holding compa...	0.088
☐ infy.txt -- Infosys Ltd. provides information technology (IT) and c...	0.098
✓ nee.txt -- NextEra Energy, Inc., through its subsidiaries, engages i...	0.763
☐ nxst.txt -- Nexstar Broadcasting Group, Inc., a television broadca...	0.114
☐ oakbx.txt -- The investment seeks income and preservation and ...	0.000
☐ ompi.txt -- Obagi Medical Products, Inc., a specialty pharmaceuti...	0.008
✓ qre.txt -- QR Energy, LP, through its subsidiary, QRE Operating, L...	0.782
☐ relianc banc.txt -- Reliance Bancshares, Inc. operates as a multi-...	0.035
☐ rrd.txt -- R.R. Donnelley & Sons Company provides integrated co...	0.077
✓ sempra.txt -- Sempra Energy operates as an energy services hold...	0.802
☐ splix.txt -- The investment seeks replicate the total return of the U...	0.000
☐ tata comm.txt -- Tata Communications Limited provides integrated...	0.131

Keywords for selected documents: energy, gas, natural_gas, natural, pumping, generated, services, operates, oil, million

sempra.txt

Sempra Energy operates as an energy services holding company worldwide. The company operates utilities, develops new energy infrastructure, and provides energy-related services to approximately 31 million consumers worldwide. It operates in five segments: San Diego Gas & Electric, Southern California Gas Co., Sempra Generation, Sempra Pipelines & Storage, and Sempra LNG (liquefied natural gas).San Diego Gas & Electric is a regulated public utility that provides energy services to 3.5 million consumers through 1.4 million electric meters and approximately 850,000 natural gas meters. Its service territory covers 4,100 square miles in San Diego and southern Orange County. Southern California Gas Co. operates as a natural gas distribution utility, providing energy services to 20.9 million consumers through 5.8 million meters. Its service territory encompasses approximately 20,000 square miles throughout Central and Southern California. Sempra Generation provides electric power to markets in North America. It owns and operates a fleet of low-emission natural gas-fueled power

Document name:

Figure 8: Query and Concept's document of the ontology along with the sim graph

5. EVALUATION OF ONTOLOGY

As discussed in section 4 we propose to have two ontologies with us one with WSD (shown in figure 7) and the other will be without WSD. For evaluating these ontologies for classification we plan to use such a tool where we can get the results in terms of Precision, Recall, F-Measure. These are the metrics for accuracy. We propose to create the *arff files* for this classification [11]. These *arff* files will be treated as the main source for calculating the accuracy of the classification process.

6. CONCLUSION

In our proposed system we have used all the non-traditional tools and technologies to achieve the classification of text documents. And expect it to be a good approach to do the classification. The heart of the system is the WSD module which has helped to overcome the disambibuation. This is done by using modified LESK algorithm and WordNet. WordNet has made possible the keyword formation as well as the WSD module. To incorporate WordNet in our system we used JAWS [12] which is a very efficient interface for accessing WordNet in java applications. The ontology editor OntoGen proved to be very helpful as it provides semi-automated creation of ontology and due to which we could apply dismbiguation at this level. About six concepts were created for the ontology by this strategy for our corpus. We expect more accuracy for this classification.

7. FUTURE SCOPE

The main aim was to find the effect of having created the semi-automated ontology by eleminating disambiguation. In this case we have considered only one class for each document. The document can be classified in other concept as well. The ontology can be created by taking care of this and the results can be seen for accuracy. Also the ontology is single level hierarchy, sometimes there can exist more level of concepts giving rise to sub-concepts of the concept. Another thing can be done for evaluation process. Evaluation of the classification is comsuming more time as we need the *arff* file, to reduce time for evaluation creating *arff* files can be automated.

REFERENCES

[1] Ontologies Improve Text Document Clustering Andreas Hotho, Steffen Staab, Gerd Stumme {hotho,staab,stumme}@aifb.uni-karlsruhe.de Institute AIFB, University of Karlsruhe, 76128 Karlsruhe, Germany

[2] http://books.google.co.in/books?id=Z0ExaN_ZssC&pg=PA693&dq=classification+using+ontology&hl=en&sa=X&ei=fDAiT5r1KIOrrAfrq4G0CA&ved=0CFYQ6AEwBg#v=onepage&q=classification%20using%20ontology&f=false

[3] Training-less Ontology-based Text Categorization: lsdis.cs.uga.edu/~mjanik/presentation/20080701-PHD_Defense.ppt, Dr. Krzysztof J. Kochut, LSDIS lab, Computer Science, University of Georgia

[4] Using WordNet for Text Categorization- Zakaria Elberrichi1, Abdelattif Rahmoun2, and Mohamed Amine Bentaalah1 1EEDIS Laboratory, Department of Computer Science, University Djilali Liabès, Algeri 2King Faisal University, Saudi Arabia

[5] General Framework for Text Classification based on Domain Ontology, Xi-quan Yang1, Na Sun1, Ye Zhang1, 2, De-ran Kong1. 978-0-7695-3444-2/08 $25.00 © 2008 IEEE DOI 10.1109/SMAP.2008.17

[6] An overview to Ontology, WordNet and WSD and a proposed system for Ontology-based text document classification, Nazia Ilyas Baig, Gresha Bhatia. International Conference on Computer Science & Information Technology (ICCSIT- 2012),Goa, India. ISBN:978-93-82208-03-7

[7] http://en.wikipedia.org/wiki/WordNet

[8] An ontology Clarification Tool for Word Sense Disambiguation,Manasi Kulkarni1 Suneeta Sane

[9] OntoGen: Semi-automatic Ontology Editor, Blaz Fortuna, Marko Grobelnik, and Dunja Mladenic, M.J. Smith, G. Salvendy (Eds.): Human Interface, Part II, HCII 2007, LNCS 4558, pp. 309–318, 2007. © Springer-Verlag Berlin Heidelberg 2007

[10] http://biz.yahoo.com/

[11] Probabilistic Reasoning with Naïve Bayes and Bayesian Networks Zdravko Markov1, Ingrid Russell Department of Computer Science, Central Connecticut State University, 1615 Stanley Street, New Britain, CT 06050.

[12] http://lyle.smu.edu/~tspell/jaws/index.html