IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 4, April 2017

Domain Identification and Detection

S. S. Kulkarni[1], Sayali Deshmane[2], Rohan Chawla[3], Deep Vora[4], Pranjali Rautela[5]

Professor, Department of IT, SAE, Pune, India[1]

Student, Department of IT, SAE, Pune, India[2,3,4,5]

Abstract: In this internet era, vast amount of data is available and is generated on a continuous basis. For a variety of purposes, identifying the area to which a particular piece of text belongs is very crucial. This enables various data mining tools to better handle the text in terms of information extraction/mining. In this project we aim to provide that preliminary meta-information about a particular piece of text. In the virtual world, this automation is manifested through the evolution of efficient algorithms. Part of the process of automation in the virtual world is also dependent on enabling machines to do the tasks that humans naturally do. Domain identification is one such technique. In this paper, we plan to highlight the efficient use of "Natural Language Processing" (NLP) techniques to identify the domain of a given piece of text.

Keywords: URL, keyword matching, domain, database storage.

I. INTRODUCTION

We consider topic detection without any prior knowledge of category structure or possible categories. In this paper we consider the problem of identifying the topic of a particular document just by entering the URL. We have addressed three domains in this paper namely Sports, Politics and Agriculture. As far as Sports is considered we have decided to cover all the 50 sports of the Olympics as well as a few remaining ones. We treat the problem of identifying and characterizing a topic as an integral part of the task. As a consequence, we cannot rely on a training set or other forms of external knowledge, but have to get by with the information contained in the document itself.

The approach we will follow consists of three steps. First we work on our raw data by scraping it from the website and cleaning it. Secondly, we extract a list of the most informative keywords and create our own database. Thirdly, by using a certain proposed algorithm we track the domain or topic of the website entered.

The organization of this paper is as follows. In section 2 we discuss related work and the related problem of keyword extraction. In section 3 we highlight our proposed system. We end the section with a brief description of the expected results.

II. RELATED WORK

There has been fair amount of work done in the area of topic detection. But in most of the work done, the task of topic detection is achieved using probabilistic approach as compared to our non-probabilistic approach for the problem. Some of the existing models are as follows:

2.1. Emerging Topic Detection on Twitter

Twitter is a user-generated content system that allows its users to share short text messages, called tweets, for a variety of purposes, including daily conversations, URLs sharing and information news.[3] In this system, as information producers, people post tweets for a variety of purposes, including daily chatter, conversations, sharing information/URLs and reporting news, defining a continuous real-time status stream about every argument. Considering this aspect, one of the founders of Twitter.com, Evan Williams, defined the service as follows: What we have to do is deliver to people the best and freshest most relevant information possible. We think of Twitter as it's not a social network, but it's an information network. It tells people what they care about as it is happening in the world. Topic detection technique permits to retrieve in real-time the most emergent topics expressed by the community. First, the contents i.e. set of terms are extracted and the term life cycle is modelled according to a novel aging theory intended to mine the emerging ones. [3]A term can be defined as emerging if it is frequently occurring in the specified time interval and its occurrence was relatively rare in the past. Moreover, considering that the importance of a content also depends on its source, the social relationships in the network are analyzed with the well-known Page Rank algorithm to determine the authority of the users. Finally, a navigable topic graph is leveraged which connects the emerging terms with other semantically related keywords, allowing the detection of the emerging topics, under user-specified time constraints.

DOI10.17148/IJARCCE.2017.6489

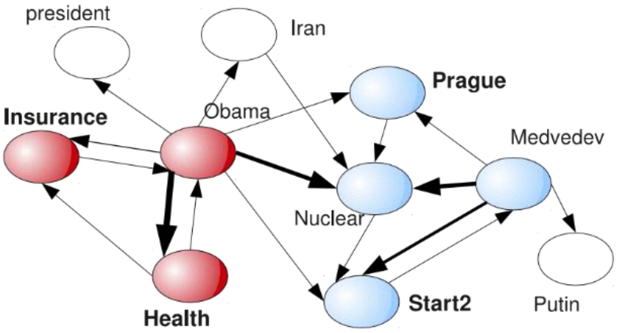

Figure 1: A Topic graph with two Strongly Connected Components (in red and yellow) representing two different emerging topics: labels in bold represent emerging keywords while the thickness of an edge represents the semantical relationship between the considered keywords.

2.2. Google News

2.2.1. To Identify Articles

Not every page on the web is an article. Techniques are applied to distinguish an article page from a non-article page, from a page that's trying to sell products, from a listing of other articles, and so forth.

2.2.2. To Identify Text in an Article

Articles have text embedded inside along with a lot of unwanted boilerplate, ads, copyright messages etc. The article is segmented to just get the article text and throw away the rest. This is an information extraction problem. Most sites use HTML DOM/SAX parsing along with a lot of other heuristics. [7]

2.2.3. To Identify Keywords

Articles are a lot of text,that include all kinds of conjunctions, connectives, pronouns, nouns, numbers, etc. Techniques like TF-IDF exist to get to a good distance. Some types of features are more important than others - especially when the objective is to group related articles together. When you are considering news, you are more interested in putting articles from an incident (or event) together. For example, you want articles from an armed robbery in Albania to come together, rather than all articles on robberies from around the world. It so happens that "named entities" (proper nouns) are best suited to characterize an incident. So they are given more weightage. Sometimes considering phrases (like New York) also help improve the quality of clustering. [7]

2.2.4. To Identify Similar Documents

Till now, an article is brought to its vector form. A vector of keywords and weights - that depict importance to the article. If we have two such vectors, how we do figure the similarity between them? Measures like cosine similarity exist here. There are several similarity measures, and each one has pros and cons. For example, cosine similarity also gives importance to terms found in one article, and not found in the other. So if one document is a superset of another, the cosine similarity may still be low.

2.2.5. To Group Similar Documents

Document clustering is an extremely well researched topic. To start with, we get algorithms off the books - like hierarchical agglomerative clustering, k-means clustering, and top down clustering. K-means clustering may not be best suited when we don't know how many clusters we have to group articles into. So, we have to inspect HAC and the top down methods. The biggest hurdles to conquer in a production clustering system are distributed clustering, scalability and incremental approach of articles.

Google checks to see what individual stories are being published on each news source and then determines the grouping based on the following.

1. Unique identifying keywords such as names, places, things etc.
2. The timing of the story (do other sources include similar articles around the same time?)
3. Quotations from persons interviewed.

This grouping and ranking are algorithmically determined and the results are not always accurate by any means. At the same time Google only allows for a maximum of 10 story clusters at any given time and for vague search terms will not produce accurate results, due to a limit of 10 different "stories" per search term.

III.PROPOSED SYSTEM

In our approach, we work with a single article instead of an entire corpus of documents. We are dealing with three main domains namely Sports, Politics and Agriculture. Once a URL is entered in the textbox, our main aim is to categorize it into one of the three domains. For this purpose, we have three main procedures going on in the backend of our project. They are data pre-processing, database creation and algorithm implementation. Each of these procedures is elaborated in this section.

3.1. Data Pre-Processing

Once a URL is entered, our main task is scraping the data from this link. Scraping data is one of the important aspects because a link may contain several advertisements and irrelevant information. Scraping the correct data, therefore, is an important task. We aim to scrape only the textual data. Every website has a different structure. Designing a generalized scraper for all the websites together is not possible. We aim to take 10 websites from each domain into consideration i.e. 10 pertaining to sports, politics and agriculture each. For this we plan to make use of Stanford NLP and JSOUP Jar file.

3.2. Creation of a Database

Once, the data is scraped from the website, our next step is the collection of keywords from the data. In our approach we maintain our own database with keywords related to the selected domains.We have tried to automate the entire process of data gathering and database creation. An attempt has been made to create a web crawler for the selected websites to gather data about the domains. The crawler finds the appropriate articles and scrapes only the textual data. Using Stanford NLP we reduce the text to basic Parts of Speech such as nouns, proper nouns, verbs and adjectives. We use the term frequency to find the keywords.

The keywords have a greater count in the article then the other words. Every domain may have certain fuzzy words which fail to point to one particular domain or can be a part of any domain in general. We identify these junk words and remove them from our keyword collection. An attempt has been made to automate the process of removing junk words from our collection. Once this process has been accomplished, our next step is storing these keywords into a database. A separate database is maintained to store the keywords belonging to each domain along with its proper nouns. Within our database we maintain three lists called definitive, probabilistic and inter-domain for each domain. Using the frequency aspect of our algorithm, we will put the words with highest frequency in our definitive list (eg: Proper Nouns). The terms with kisser frequency or generic terms will be put under the probabilistic list. The common terms between two domains will be under the inter-domain list. This strategy provides us with higher accuracy. The keywords stored in the database are an important asset in the application of our algorithm. The main benefit of this approach is that, we use strong keywords related to domains which excludes assumptions, thus making our proposed system error-free and more accurate.

Algorithm for Database Creation

1. Scrape article from multiple sites using site-specific scraper to create a corpus of documents related to that domain.
2. Parse the corpus through Stanford NLP to get the nouns, verbs and adjectives from the corpus.
3. Get the frequency of words. Keywords will have a frequency higher than that of the normal words.
4. Analyze the data and decide the threshold value for a keyword.
5. The words above the threshold value will be put in the definitive list, the words between a certain value and threshold value will be put in the probable list.
6. The proper nouns obtained from the corpus are to be entered into the definitive list excluding the location. This constitutes our definitive list.
7. The words common between two domains will be put in inter-domain list.

3.3. Algorithm Implementation

In Haribhakta's paper for "Unsupervised model for Topic Detection", a simple yet effective method is proposed to uniquely identify a topic. [1] We work on the same terms with a little modification. Firstly, we consider just a single document and solely identify the topic for that document. The algorithm is as follows:

Algorithm:
Step 1: Scrape data from the URL provided by the user, using the scrapers designed for the different websites.
Step2: Parse through **SNLP** for tokenization and getting a list of nouns, verbs etc.
Step 3: Remove junk words
Step 4: Store the first 20 words from the header in the data structure **DH**. [1]
Step 5: Get the top 20 frequent words from the article and compare them with words stored in the header(**DH**).

DOI10.17148/IJARCCE.2017.6489

Step 6: The list of words that will match will be the most confident words of the article.
Step 7: In the database there are three lists: **probabilistic list**, **definitivelist** and **inter-domain list**. The keywords or the most confident words are compared with the words present in these lists. The number of hits for each domain will be recorded. Selection of domain name will take place based on the maximum number of hits.
Case 1: If the margin of difference **> 3** then the domainselection will take place based on the probabilistic list comparison.
Case 2: If the margin of difference **< 3**then the comparison is done with the definitive list also.
Step 8: The domain identified after the comparisons is the output.

3.4. Cosine Algorithm Approach:

In the last part we compare the keywords collected from our URL with the keywords in our database. We have maintained 3 array lists of keywords belonging to each domain and one array list of the keywords of the current document.
We start comparing the list for the current URL with the first array that belongs to the agriculture domain. We calculate the cosine relation between the two arrays.
Let |c| be the number of terms in current document.
Let |a| be the number of terms in the agriculture document.
Let $|a_c|$ be the intersection of the two arrays then cosine value is given as:

$$\frac{|a_c|}{\sqrt{|a|^2.|c|^2}}$$

Similarly, for the domain Sports we have:
Let |c| be the number of terms in current document.
Let |s| be the number of terms in the sports document.
Let $|s_c|$ be the intersection of the two arrays then cosine value is given as:

$$\frac{|s_c|}{\sqrt{|s|^2.|c|^2}}$$

Similarly, for the domain Politics we have:
Let |c| be the number of terms in current document.
Let |p| be the number of terms in the sports document.
Let $|p_c|$ be the intersection of the two arrays then cosine value is given as:

$$\frac{|p_c|}{\sqrt{|p|^2.|c|^2}}$$

Using the above formulas, we get the value of cosine for each domain. The one with the highest value is the domain of the URL.

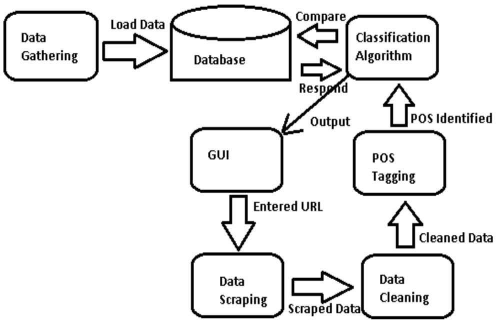

Figure 2: System Flow Diagram

IV. CONCLUSION

The proposed system to be developed was analysed and compared with the existing systems, keeping advancements in mind. Requirement analysis was done and the design was framed accordingly wherein the system is divided into multiple modules. The system thus developed would be beneficial to all the Data Scientists.

In this work, we have addressed the problem of automation and analysing a huge amount of data. Thus, we conclude that the success rate of any NLP system depends on the quality of data gathered, the transformation of data that is carried out and the technologies that are used.

NLP in the field of Automated Domain Analysis can bring a revolution if implemented with proper care.

Our system makes use of the data gathered, in the form of a data dictionary, to analyse a URL for delivering a probable domain. An automated Domain identification and detection system can ease out the tedious task of finding the domain to which a particular piece of information belongs.

ACKNOWLEDGMENT

This project is supported by STES' Sinhgad Academy of Engineering. We thank our principal, **Prof. K. P. Patil** for providing all the resources we are and will be using during the course of this project. We would like to thank our HOD **Prof. Abhay Adapanwar** for his comments that greatly improved our manuscript.

We thank our internal guide **Dr. S. S. Kulkarni** for providing useful insights and expertise that greatly assisted our project. We would also like to thank our external guide **Mr. Akshat Joshi** and our sponsoring company CDAC for providing the essential help from time to time. Any errors of our own should not tarnish the reputation of these esteemed persons.

REFERENCES

[1] "Unsupervised Topic Detection Model and Its Application in Text Categorization". Yashodhara Haribhakta, Arti Malgaonkar, Dr. Parag Kulkarni[University of Pune].
[2] "Topic Detection, A New Application for Lexical Chaining" Paula Hatch, Nicola Stokes, Joe Carthy, Department of Computer Science, University College Dublin,
[3] Ireland.
[4] "A Topic Detection and Tracking method combining NLP with Suffix Tree Clustering", Yaohong JIN**,**Institute of Chinese Information Processing**,**Beijing Normal University Beijing, P. R. CHINA
[5] Lloret, E., Topic Detection and Segmentation in Automatic Text Summarization, (Dec.2009).
[6] "Using Lexical Chains for Text Summarization", Regina Barzilay and Michael Elhadad Mathematics and Computer Science Dept. Ben Gurion University in the Negev Beer-Sheva, 84105 Israel.
[7] "Construction of Topics and Clusters in Topic Detection and Tracking Tasks" , Masnizah Mohd, Fabio Crestani, Ian Ruthven Scotland, United Kingdom
[8] www.quora.com/How-Google-News-works

BIOGRAPHIES

Sayali Deshmane Pursuing Bachelor of Engineering (B.E) in Information Technology, STES' Sinhgad Academy of Engineering, Savitribai Phule Pune University (S.P.P.U)

Rohan Chawla Pursuing Bachelor of Engineering (B.E) in Information Technology, STES' Sinhgad Academy of Engineering, Savitribai Phule Pune University (S.P.P.U)

Deep Vora Pursuing Bachelor of Engineering (B.E) in Information Technology, STES' Sinhgad Academy of Engineering, Savitribai Phule Pune University (S.P.P.U)

Pranjali Rautela Pursuing Bachelor of Engineering (B.E) in Information Technology, STES' Sinhgad Academy of Engineering, Savitribai Phule Pune University (S.P.P.U)

DOI10.17148/IJARCCE.2017.6489