International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 – 5161

Available at http://inpressco.com/category/ijcet

Project Research Article

Cognitive Web

Amit Narote†, Darshan Sapaliga†, Aditya Patil†* and Shrutik Katchhi†

†Information Technology Department, Xavier Institute Of Engineering, Mahim (W), Mumbai, India

Accepted 16 Feb 2015, Available online 20 Feb 2015, **Vol.5, No.1 (Feb 2015)**

Abstract

Our final year project is titled 'Cognitive Web' and helps the user to discover interesting stuff quickly on the Internet. Using a combination of human opinions and machine learning, Cognitive Web presents registered users with only web sites that have been suggested by other like-minded users. With time, the database will recognize user's patterns of interest, and suggest the web pages accordingly. Cognitive web is an intelligent and helpful way to make the Internet a compact and more intimate place. This project helped us to master the complexities of database management and also provided further exposure to us in the field of Artificial Intelligence.

Keywords: *Cognitive web, Opinion Mining, Artificial Intelligence.*

1. Introduction

Whether you're a seasoned internet veteran or a newbie learner, you know how slow it can be to search the Web. You go to Google.com, you type in keyword phrases, and then you skim through hundreds of hits, hoping for something that interests you. By the time you're done, you've spent 45 minutes to find perhaps two or three useful web sites. There is a much smarter and faster way to search the Web. By joining a special network of Web users, you can cut your searching time in half. It's called Cognitive Web, and it's based on people sharing their destination recommendations electronically. Cognitive Web is not a search engine driven by keywords, but rather a "browsing engine" driven by user votes. Cognitive Web is based on a central database of many of users' personal suggestions on what they consider to be quality websites. Member suggestions are categorized by topic of interest, and are updated daily to reflect a number of members' votes. The more that members like a particular website, the more highly that website is ranked. As a member of Cognitive Web, you get access to thousands of hours of other people's search experiences, and can contribute your own suggestions. Cognitive Web is an amazing website. Cognitive Web discovers web sites based on your interests, learns what you like and brings you more content similar to your liking. Using a combination of human opinions and machine learning to immediately deliver relevant content, Cognitive Web presents only web sites that have been suggested by other like-minded users.

This document on Cognitive Web will highlight the purpose and scope of developing this Website. We shall predict how the system will be used in order to gain a better understanding of the project, outline concepts that may be developed later, and document ideas that are being considered, but may be discarded as the product develops. This document provides a detailed overview of our website, its parameters and goals. It also describes the project's target audience and its user interface, hardware and software requirements.

2. Concept Review

There are different types of web search engine based on the underlying mechanism. The term "search engine" is often used generically to describe crawler-based search engines, human-powered directories, and hybrid search engines. These types of search engines gather their listings in different ways, through

1. crawler-based searches
2. human-powered directories
3. hybrid searches

A. crawler-based searches

Crawler-based search engines, such as Google, create their listings automatically. They "crawl" or "spider" the web, then people search through what they have found. If web pages are changed, crawler-based search engines eventually find these changes, and that can affect how those pages are listed. Page titles, body copy and other elements all play a role.

The life span of a typical web query normally lasts less than half a second, yet involves a number of different steps that must be completed before results can be delivered to a person seeking information. The following graphic illustrates this life span.

*Corresponding author: **Aditya Patil**

1. The web server sends the query to the index servers. The content inside the index servers is similar to the index in the back of a book - it tells which pages contain the words that match the query.
2. The query travels to the doc servers, which actually retrieve the stored documents. Snippets are generated to describe each search result.
3. The search results are returned to the user in a fraction of a second.

B. Human powered directories

A human-powered directory, such as the Open Directory Project depends on humans for its listings. (Yahoo!, which used to be a directory, now gets its information from the use of crawlers.) A directory gets its information from submissions, which include a short description to the directory for the entire site, or from editors who write one for sites they review. A search looks for matches only in the descriptions submitted. Changing web pages, therefore, has no effect on how they are listed. Techniques that are useful for improving a listing with a search engine have nothing to do with improving a listing in a directory. The only exception is that a good site, with good content, might be more likely to get reviewed for free than a poor site.

C. Hybrid searches

Today, it is extremely common for crawler-type and human-powered results to be combined when conducting a search. Usually, a hybrid search engine will favor one type of listings over another.

3. Proposed system

The proposed system will be human-powered directories based web search engine known as "Cognitive web". The key to COGNITIVE WEB is that its users can share their views on interesting Web content. Other users can then view that content, through topic-orientated browsing or purposeful searching, and reciprocate the favour by sharing their own views on interesting finds.

COGNITIVE WEB offers a personalized Web experience. At one time or another, every Internet user experiences difficulties getting the intended match from a query entered in a search engine. Narrowing search results can be a challenge, especially when you're not sure what you're looking for. Let's say you'd like to check out some cool photos taken around the world. Type "cool photos" into a regular search engine and you might get some worthwhile links mixed in with some questionable, unrelated or uninteresting content. Try the same thing with COGNITIVE WEB by selecting you interest as photos, and not only will you receive hits related to quality photography sites, the photos will crosscheck with your as well as others interests. A tool like COGNITIVE WEB helps filter Web content into a manageable, customized experience. And the more you indicate what you like, the better COGNITIVE WEB gets at offering content you will appreciate.

4. Opinion mining

The internet is one place where people can have their voice heard. As a result, billions share their views about a topic in one way or another. That's one aspect of the internet, sharing. But sharing has no value unless the shared resources or knowledge has a recipient. Keeping in mind these 2 main aspects, we are making a project named Cognitive Web, where people can share their resources which they find while surfing, furthermore they can use other's resources to surf internet in a smarter way.

Learning user's interest is the need of the day, if we know where the user's interest lie, we can make that user's surfing period memorable by suggesting selected data which lies within the realms of the user's interest.

Text classification or sentiment analysis is known to be one of the most complex and challenging tasks in natural language processing. For example when you say "This website is awesome", the sentiment analysis algorithm will recognize it as a positive statement because of the presence of the word "awesome". This was a simple example, but the human language is much more complex. People tend to express more than one view in a statement, for example "the UI of this website is mind-blowing, but the data is irrelevant"; or in this age, many user's use sarcasm like "Wow, I used this website for 30 seconds!" In such cases, the only option we have left is to teach computer our language patterns. We can pre-classify such statements into positive and negative statements and train the computer with them. Once the computer is trained with such pre-classified data, it can be used to learn patterns from actual data and classify them as positive or negative.

5. Algorithm

Step1: Create a text file that contains all the positive and negative data

Step2: Create a method that loops over the data, tokenizing the documents and storing counts of the terms for later use. Add the known data, it takes a text file, and a sentiment (either positive or negative), and if you want to limit the amount of sentiment data analyzed, enter the number of lines you want to sample as limit. Defaults to 0, meaning analyze everything.

Step3: Classify the data. Takes a string as a parameter.
The classify function should start by calculating the prior probability (the chance of it being one or the other before any tokens are looked at) based on the number of positive and negative examples - in this

that'll always be 0.5 as we have the same amount of data for each.

We then tokenise the incoming document, and for each class multiply together the likelihood of each word being seen in that class. We sort the final result, and return the highest scoring class.

Step4: Tokenize
Find matches in either positive or negative sentiment. The tokenism function takes a string as a parameter.

Step5: Put sentences of string into array.

Sep6: Create array for positive and negative sentiment.

Step7: Calculate assurance
Step7.1: To find how much bias there is (assurance), divide the positive sentiment score by the negative.
Step7.2: If the assurance is not a decimal divide the negative by positive.
Step7.3: Remove the least likey alternative

Step8: If you are not sure (the assurance is less than. 47), it's inconclusive. Otherwise, it's most likely good.

Step9: Display the sentiment.

6. Working

Cognitive Web discovers web sites based on your interests, learns what you like and brings you more. Cognitive Web terms itself as a Discovery Engine. That may sound like a scientific term, but it is in fact a website for you to discover new interesting sites. There is no toolbar to download or any software to install. You just need to go to the site, sign up and click on the Cog button (or any of your favorite tags) to start discovering new sites. If you are not happy with the result, simply click on the Cog again to go to the next site. The key to COGNITIVE WEB is that its users can share their views on interesting Web content. Other users can then view that content, through topic-orientated browsing or purposeful searching, and reciprocate the favour by sharing their own views on interesting finds. COGNITIVE WEB offers a personalized Web experience. At one time or another, every Internet user experiences difficulties getting the intended match from a query entered in a search engine. Narrowing search results can be a challenge, especially when you're not sure what you're looking for. Let's say you'd like to check out some cool photos taken around the world. Type "cool photos" into a regular search engine and you might get some worthwhile links mixed in with some questionable, unrelated or uninteresting content. Try the same thing with COGNITIVE WEB by selecting you interest as photos, and not only will you receive hits related to quality photography sites, the photos will crosscheck with your as well as others interests. A tool like COGNITIVE WEB helps filter Web content into a manageable, customized experience. And the more you indicate what you like, the better COGNITIVE WEB gets at offering content you'll appreciate.

Recommendation engine

Recommendation engine will provide web sites to user either randomly or by the preference given. If the user is new to proposed website and user hasn't given areas of interest then web pages are shown randomly. If user has given preferences then web sites are shown according to the give preference.

COGNITIVE WEB is all about site discovery. User clicks on the "Cog!" button and the recommendation engine would return some random site based on the categories user said he/she was interested in. But then the more user uses it, better sites would be sent on his way. This is because it's not actually random, but rather sites are served up based on a series of processes that go on within the COGNITIVE WEB

A. Recommendation Engine.

As you can see in the chart above, there are two key parts to the Recommendation Engine. There are pages from the topics you marked that interest you and peer endorsed pages. Peer endorsed pages are ones from users who have similar voting habits (giving a site the thumbs up or thumbs down) as you.

When a site is first visited, it is put through both the Classification Engine and the Clustering Engine as shown above. The Classification Engine filters the page by topic and tags. While the Clustering Engine suggests the pages based on users past experience. The Clustering Engine sorts out the votes a site is getting so it can determine which sites are the quality ones that should be served. This engine is a key component which holds the system together.

This all makes for a system of "quality plus relevance". And as with any of the recommendation engines, the more data you have, the better it'll perform.

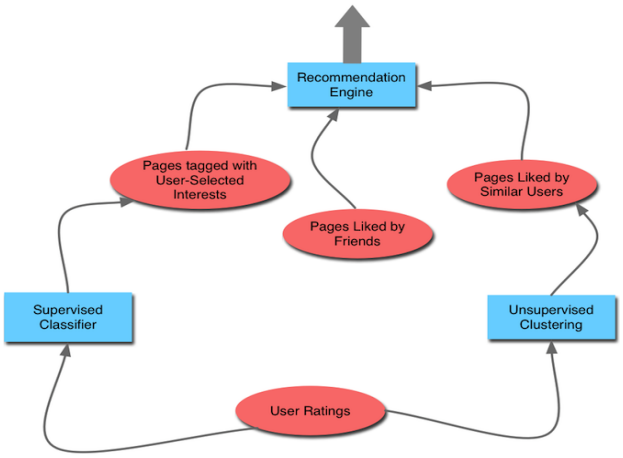

Figure 3.1 Recommendation Engine

B. Pseudo code

1. User Signs Up
    1.1 User Logs In
2. Generate Queue: Priority: High Rated pages to low Rated
    2.1: If User likes a Link
        2.1.1: G=Get(Genre)
        2.1.2: Add User's rating to the database
        2.1.3: Prioritize Genre G in that queue
        2.1.4: Display Webpages on click.
    2.2: If User dislikes a Link
        2.2.1: H=Get(Genre)
        2.2.2: Add User's rating to the database
        2.2.3: Push Genre H down to the end of the queue
        2.2.4: Display Webpages on click.
    2.3: If user comments on a link
        2.3.1: J=Get(Genre)
        2.3.2: Run Opinion mining algorithm
        2.3.3: make changes in the algorithm
        2.3.4: Display Webpages on click.
    2.4: Else Display Random Webpages.
3. Update and Make copy of the database.

6. Test Results of Opinion Mining

Here we test out few sentences, the results obtained while testing out our opinion mining tools are encouraging. We had a success percentage of
We started with simple statements to test our opinion mining algorithm.
Sentence1: I'll go ahead and say this is the worst site for searching
Result: Negative
Sentence 2: UI is amazing
Result: positive
Then we tried ambiguous but conclusive statements.
Sentence 3: exceeds expectations
Result: positive
Sentence 4: the simple structure of the website is comforting yet pretty useless
Result: negative
Then we added a negation to the statement.
Sentence 5: Not very user friendly
Result: Negative

Lastly, we tried an actual review which was too complicated for the algorithm to decide. We added a special functionality in the code wherein if the net positive and negative meanings do not have much difference, it will be referred as inconclusive and will be save for a human reading.

Sentence 6: In short, uneducated redneck sees 9/11 unfold on his TV, uneducated redneck joins the Military and shoots brown people. No questions asked. 'Merica I understand why this movie was made. Dumb people that aren't capable of thinking outside of the box STILL view Iraqis as bad people despite the fact that nobody in Iraq had anything to do with 9/11, and Americans INVADED THEIR country, for literally no reason. If anything, that time period in American history should be buried, not glorified. The guy this movie is based on is a racist pig that made millions of dollars off people that are somehow dumber than he is.

Back in High School we would watch old Nazi Propaganda films, which depicted the Germans as superior, and everyone else as not being worthy of breathing the oxygen in the air. 200 years from now this piece of garbage will be viewed in some history class, and I only hope the teacher informs the class that not ALL Americans are this stupid.
Result: inconclusive

Conclusion

The Internet is just a world passing around notes in a classroom. What are trying is to impart a bit of knowledge in those notes.

The proposed idea is comparatively unique and its scope is wide. Instead of selecting one generic topic, we tried to select a topic which includes many generic ideas.

In this age of media and Internet access, we are much more talkative than ever before. We are planning to use this expressiveness for a good cause, to make people see what they are missing out on.

Future Scope

Our project is still at the prototyping stage. In future, we wish to flawlessly implement all the features from our proposed system with maximum efficiency.

Database expansion: We plan on expanding out database to a minimum of 1500 websites before the final presentation of the project. Such a huge number of websites will really give the users a wide variety of websites.

User reach: We will plan to reach out to as many users as we can. Because, in the end, user interaction plays a key role in our algorithm. We want users to take the front row and carry the show further.
User's resources: We plan to add a functionality where the user can add a website.

References

Magdalini Eirinaki, Shamita Pisal, Japinder Singh (2010), Feature-based opinion mining and ranking, *Computer and System Sciences*, 1-3

P.D. Turney, M.L. Littman (2003), Measuring praise and criticism: Inference of semantic orientation from association, *ACM Trans. Inf. Syst. 21*, 315–346.

A. Esuli, F. Sebastiani (2005), Determining the semantic orientation of terms through gloss classification, *14th ACM International Conference on Information and Knowledge Management, CIKM'05,* 617–624.

Q. Miao, Q. Li, R. Dai (2008), An integration strategy for mining product features and opinions, *17th ACM Conference on Information and Knowledge Management, CIKM'08*, 1369–1370

Khairullah Khan, Baharum Baharudin, Aurnagzeb Khan, Ashraf Ullah (2014), Mining opinion components from unstructured reviews, *King Saud University – Computer and Information Sciences,* 1-5

B. Liu, M. Hu, J. Cheng (2005), Opinion observer: Analyzing and comparing opinions on the web, *14th International Conference on World Wide Web, WWW'05,* 342–351

S.M. Kim, E. Hovy (2004), Determining the sentiment of opinions, *20th International Conference on Computational Linguistics, COLING'04,*

Turney Peter (2002), Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. *40th Annual Meeting on Association for Computational Linguistics, Philadelphia, Pennsylvania, USA*