On-the-fly Context Based Definition Fetcher

[I]Shashidhara, [II]Prof, Madesha, [III]Alwyn Edison Mendonca
[I,II,III]Dept. of CSE, Sahyadri College of Engineering & Management, Adyar, Mangalore, India

Abstract

Internet is the main source of knowledge. There are many web-pages which provide the definition or details of almost all the things that exist on the planet. When the user browse the web page, all the sentence or words in the webpage may not be understandable. She/he may need to check for the meaning. If we associate the complex word with the corresponding meaning based on the context, then it would be great help for the user to understand the word without wasting time. The source of definition can be fetched from external source like Google or Word-Net.

Keywords

Application Programming Interface (API)

I. Introduction

When the user browse through the web pages, She/he may not know the definition of all the words which is there in the web pages. This makes the user to check dictionary book or open check definition from other website. This is the extra overhead for the user.

In the existing systems like ASKIt (A firefox add-on), Dictionary.com, it fetch the definition of the word from dedicated server. The definition fetched from these websites are not based on the context where the used. The definition will be greneral, which will be same in all cases. ASKIt is the add-on for firefox which fetches data from Google and Word-Net.

TAGME is the existing tool for accurate annotation of text with Wikipedia pages. This tool helps to get the context based definition. It provides APIs to get the job done. This system built using Java programming language. TAGME API return the JSON data. Giving the sentence as input to the TAGME, the return JSON data can be parsed and tagged with complex words.

The natural language processing must be used in order to understand the context and select the right article from the data source. Disambuation from the existing meaning set is an interesting concept, deciding the right definition makes concept more interesting. Also for this one should have the knowledge of data mining. There should be large dataset and response of the server should be quick. By combining the TAGME and the mentioned add-ons, context based definition can be fetched for any given word.

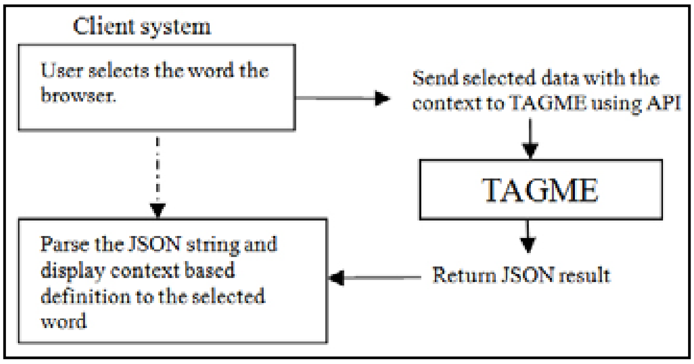
II. Literature Survay

The issue of cross-referencing content parts with Wikipedia pages, in a way that synonymy and polysemy issues are resolved accurately and gainfully. We take inspiration from a late stream of work , and amplify their circumstance from the clarification of long records to the clarification of short messages, for instance, bits of chase engine results, tweets, news, web diaries, etc.. These short and inadequately made compositions pose new challenges similarly as effectiveness and ampleness of the remark method, that we address by arranging and planning Tagme, the chief framework that performs an exact and on-the-fly remark of these short printed segments. An immense game plan of examinations shows that TAGME defeats front line computations when they are conformed to manage short messages and it comes about snappy and forceful on long messages.

Supporters of the well known news or website feeds(RSS/Atom) regularly confront the issue of data over-burden as these food sources ordinarily convey expansive number of things occasionally. One answer for this issue could bunch comparative things in the food peruser to make the data more reasonable for a client. Grouping things at the food peruser end is a testing errand as typically just a little part of the real article is gotten through the food. In this paper, we propose a technique for enhancing the precision of bunching short messages by advancing their representation with extra elements from Wikipedia. Exact results demonstrate this advanced representation of content things can considerably enhance the grouping exactness when contrasted with the customary sack of words representation.

III. Methodology

TAGME provides APIs to fetch context based definition from the wikipedia. We assume that wikipedia has almost all the existng words. Whenever the user need to get the definition, proposed system will send the request to the TAGME and response will be in JSON format. The JSON format then parsed and definition is attached to the word for which the user requested the definition. In the client side i.e. in the user browser, the javascript will send the request to the TAGME. This is done by using the TAGME APIs.


Client system
User selects the word the browser.
Send selected data with the context to TAGME using API
TAGME
Return JSON result
Parse the JSON string and display context based definition to the selected word

In the browser, definition of the word is displayed based on the context. User will not know that the entire context around the word is sent to the TAGME server. She/he will just select the word or double click on the word, meaning will be shown based on the context. Without the knowledge of the user, the paragraph or words around the word is sent to the TAGME server.

The result of the server is in JSON format, JSON string is then parsed and meaning of the selected word is shown to the user.

IV. Algorithm

Step 1: User select the word/double click on the webpage.
Step 2: Validate the text.
Step 3: Send the word along with the context to the TAGME

server.

Step 4: TAGME returns JSON string.

Step 5: Parse the JSON string, get the meaning of the selected word and display on the word.

V. Our Result

ASKIt	TAGME	Proposed System
Fetch result without context based	Fetch result for a paragraph and tag with meaning for all words	Fetch meaning for selected word based on context
Use: On the fly definition fetcher	Use: Separate tool. Not integrated with browser.	Use: Combination of these two techniques with help the user to understand the complex words.

VI. Conclusions

In this paper we have designed the system such for the user benefit of the user. This will combine the previous project ASKIt and the TAGME, fetch the definition from the TAGME and display in the web page. Along with this definition, it will be very useful if we are able to display detail of the context and its history. This we can take as the future work. Since Google provides

VII. Acknowledgement

This research was supported by Sahyadri College of Engineering & Management. We thank our colleagues from Srinivas institute of technology who provided insight and expertise that greatly assisted the research. We thank Paolo Ferragina for assistance with TAGME [1].

References

[1] *Fast and Accurate Annotation of Short Texts with Wikipedia Pages Paolo Ferragina ; Ugo Scaiella Publication Year: 2012, Page(s):70 – 75.*

[2] *S. Cucerzan. Large-scale named entity disambiguation based on Wikipedia data. Proc. of Empirical Methods in NLP, 2007.*

[3] *C. Carpineto, S. Osiński, G. Romano, and D. Weiss. A survey of web clustering engines. ACM Comput. Surv., 41(3):1–38, 2009.*

[4] *P. Ferragina and U. Scaiella. TAGME: On-the-fly annotation of short text fragments (by Wikipedia entities). In Proc. ACM CIKM, 2010 (poster).*

[5] *Gabrilovich. Feature Generation for Textual Information Retrieval Using World Knowledge. PhD Thesis, Department of Computer Science, Technion – Israel Institute of Technology, Haifa, Israel, 2006*

Author's Profile

Name: Shashidhara,
Qualification: Mtech in Computer science and Engineering.

Name: Prof, Madesha
Qualification: Mtech in Computer science and Engineering.

Name: Alwyn Edison Mendonca,
Qualification: Mtech in Computer science and Engineering.