SMART USE OF ENTITY EXTRACTION IN E-SHOPPING USING NATURAL LANGUAGE PROCESSING

[1]ANIRUDDH CHATURVEDI, [2]PRATIK CHAVAN, [3]YOGEN CHAUDHARI, [4]PROF. SMITA R. CHUNAMARI

A.C Patil College of Engineering, Kharghar, Navi Mumbai
Email: aniruddhrchaturvedi@gmail.com, chavan633@gmail.com, yogen.chaudhari@gmail.com, srchunamari@acpce.ac.in

Abstract— Natural language processing (NLP) is a field of computer science, artificial intelligence, and linguistics concerned with the interactions between computers and human (natural) languages. Entity Extraction being the major sub-domain in NLP can have varied applications. We exploit this to make e-shopping more efficient and less cumbersome.

Keywords—Named Entity Recognition, Natural Language Processing, e-shopping, e-commerce.

I. INTRODUCTION

The purpose of this paper is to propose a smarter semantic based search web-application which is much faster, easy-to-use and involves precision in finding products the customer needs and can have a edge above the key-word based search technique. Starting with a speech-to-text converter to get the target sentence which is further put through the entity extraction module which recognizes the items required by the customer and further returns the items for the customer to select. In the next chapters there will be a detail explanation the architecture, implementation and the integration.

II. EXISTING TECHNOLOGIES IN E-SHOPPING FOR SEARCHING ENTITIES

Keyword-Based Search and filtering

The most common search engines are “Keyword based” which means all text query and retrieval will operate under keyword rules of stemming. Increasing complications further, many text indexing systems generally pick up every word in the text except commonly occurring stop words such as "a," "an," "the," "is," "and," "or," and "www”. This means these search engines are completely devoid of offering "meaning" [5]. Users want answers instantly by way of asking naturally not by guessing using a single or group of words that a search engine must logically pair with potential words it "might" be related too. Conventional Search Engines are very helpful in finding information on the internet and getting smarter with the passage of time, but they suffer from the fact that they do not know the meaning of the terms and expression used in the web pages and the relationship between them. Current keyword based search technologies have reached a plateau. Surveys indicate that almost 25% of Web searchers do not find adequate results in the first set of URLs returned, in part due to the daily sixty-terabyte increase in the size of the Web [6]. For keyword based search engines, the “amount of Web content outpaces technological progress”. In addition to their inability to keep pace with the growth of the Web, search engines rely too heavily on keyword-based string matching and word frequency and proximity techniques. As a result, queries are often overly sensitive to certain vocabulary used in the initial query string [7]. Search words often have multiple meanings or appear in multiple contexts, many of which are irrelevant to the Web searcher. Further, semantically similar pages that are desirable are often not retrieved, resulting in a set of results that is far from comprehensive. Some of the limitations of traditional search engines are:

- Problem due to Polysemy words (one word having several meaning).e.g. word “Bank” it can be financial institution or river shore.
- Problem due to synonymy (several words having same meaning) e.g. For example, “baby” and “infant” are treated as synonyms in many thesauri, but “Santa Baby” has nothing to do with “infant”. “Santa Baby” is a song title, and the meaning of “baby” in this international Conference on Cloud, Big Data and Trust 2013, Nov 13-15, RGPV 131 entity is different than the usual meaning of “infant” [8].
- Traditional Information Retrieval (IR) technology is based almost purely on the occurrence of words in documents. The availability of large amounts of structured, machine understandable information about a wide range of objects on the Semantic Web offers some opportunities for improving on traditional search.
- Low Precision and Low Recall Problem. Precision is the fraction of the documents retrieved that are relevant to the user's information need. While Recall is the fraction of the documents that are relevant to

the query that are successfully retrieved. Both precision and recall are therefore based on an understanding and measure of relevance [9].

III. PROPOSED ARCHITECTURE

As stated above this paper is to introduce an architecture for an e-shopping website which involves a different searching methodology using the concepts of artificial intelligence, writing set of rules and knowing the users mind. The presentations of Prof. Diana played a major role in the construction of this architecture [3]. It served as a stepping stone to the construction of the architecture shown below :

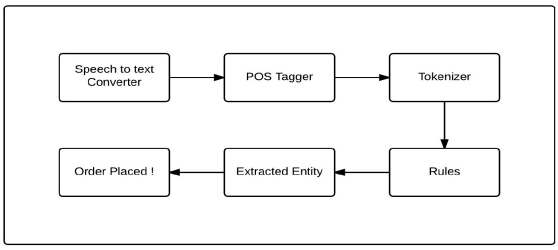

Fig : 1 Architecture involving the basic steps.

The architecture in stages can be explained as follows :

1. Speech-to-Text converter

This module involves flexibility of shopping anywhere with just speaking the order with some trigger word and also serves as the input to the next step in the architecture which is the actual statement which is going to be processed further.

2. POS Tagger

A Part-Of-Speech Tagger (POS Tagger) is a piece of software that reads text in some language and assigns parts of speech to each word (and other token), such as noun, verb, adjective, etc., although generally computational applications use more fine-grained POS tags like 'noun-plural'. This software is a Java implementation of the log-linear part-of-speech tagger. Its prefered using Stanford log-linear POS tagger. Many people think the tagger is slow but it is not the case. People who think that the tagger is slow have made the mistake of running it with the model wsj-0-18-bidirectional-distsim.tagger. In applications however there is a need to use the english-left3words-distsim.tagger. It's nearly as accurate (96.97% accuracy vs. 97.32% on the standard WSJ22-24 test set) and is an order of magnitude faster [3]. Comparing apples-to-apples, the Stanford POS tagger isn't slow.

Example:
Plain text: i want to buy a refrigerator
Tagged: i_LS want_VBP to_TO buy_VB a_DT refrigerator_DT

3. Tokenizer

In general Tokenizer is a module which divides a statement into various tokens/ chunks.
The tagged sentence is given as input to this module.
The output of tokenizer is an array of tokens and an array of their corresponding tags.
Example:
Tagged Input: i_LS want_VBP to_TO buy_VB a_DT refrigerator_DT
Output of tokenizer:
Token: i want to buy a refrigerator
Tag : LS VBP TO VB DT NN

TOKEN	TAG
i	LS
want	VBP
to	TO
buy	VB
a	DT
refrigerator	NN
Array of tokens	Array of corresponding tags

Table 1 : Explaining the array for tokenization.

4. Rules

Now after the tagging, the various parts of speech in the array at a corresponding location according to the sentence structure and now the main part of writing rules comes into picture.

The Table 1 tags can be interpreted as follows :

TAGS	THE CORRESPONDING MEANING ACCORDING TO THE STANFORD POS TAGGER
LS	List item marker
VBP	Verb, non-3rd person singular present
TO	To
VB	Verb, base form
DT	Determiner
NN	Noun, singular or mass

Table 2 : Tags and their corresponding standard Stanford POS Tagger meanings.

Now as it can be seen in the above example it can be observed that the refrigerator is "NN" and thus there is a need to select this as a target and write rules related

to it using if else looping which further enables in determining the correct object restricting the unnecessary matter in the sentence entered and guiding the search for a particular object which the customer wants.
The above sentence will return the various refrigerators irrespective of which company or what features (eg: single door or double door) is needed as the input doesn't state so. But now if the additional feature is mentioned the search becomes more precise. The more detailed the input the similar detailed the output will be:
Example :
i want to buy a single door refrigerator of lg having 230 liters capacity and a 4 star rating

i_LS want_VBP to_TO buy_VB a_DT single_JJ door_NN refrigerator_NN having_VBG 180_CD liters_NNS capacity_NN and_CC 4_CD star_NN rating_NN

This above example is large involving various attributes and thus the rules are more complex and involve proper usage of the loops. Thus there were various rules prepared for some of the items. Now as the input was precise the customer would be returned with a refined option, thus saving time.

5. Extracted Entity
It is the output and the result of the above stages which yields us a result of various options for the customer to choose from. It involves matching the item in the database and if the item is present then showing the item else notifying the user to refine his/her search.

6. Order is Placed !
This involves showing the user with the desired product which was retrieved from the database, availability and other basic cart details if the desired item is present in the database else it shows a refine search or invalid search option.

IV. ADVANTAGES OF OUR SEARCH OVER THE KEYWORD-BASED SEARCH

Firstly, it can be said that one doesn't need to look at all the available options if he/she knows the item and the specs one needs and thus will search for the particular item and will get that item only.
Secondly, instead of manually thinking and typing the keyword one can naturally say what one needs.

CONCLUSION

Thus, considering the above working and theories that this smart use of entity extraction is and efficient and a comfortable option for e-shopping with the speech-to-text converter also being appended to the module. It overcomes the drawbacks of the keyword-based search and thus proves powerful and effective.

REFERENCES

[1] Guntur Bharadwaja Kumar and Kavi Narayana Murthy. Ucsg shallow parser. In Computational Linguistics and Intelligent Text Processing, pages 156–167. Springer, 2006.
[2] Stanford POS tagger. The Stanford natural language processing group.
[3] NLP-NER presentation by Prof. Diana which served as an inspiration.
[4] Bob Carpenter and Breck Baldwin. Natural language processing with lingpipe 4, 2011.
[5] The comparison between keyword-based search and filtering and the natural language search given on www.inbenta.com.
[6] W. Roush, “Search beyond Google,” Technology Review, 2004.
[7] G. Antoniou and V. Harmelen, “A Semantic Web Primer,” MIT Press Cambridge, Massachusetts, 2004.
[8] X. Wei, F. Peng, H. Tseng, Y. Lu and X. Wang, “Search with Synonyms: Problems and Solutions,” 2008.

★★★