International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-5 Issue-6, November 2015

Voice Driven Dynamic Generation of Webpages

Sahil Modak, Suril Shah, Sagar Vikmani, Lakshmi Kurup

Abstract— Designing dynamic webpages is a hectic task for experts and a complicated job for new users. Techniques imparting natural language processing provide a simple interface to the new users to handle the cumbrous applications. The proposed system incorporates natural language processing as a mean for human computer interaction to automate webpages generation. User will interact with our proposed system by communicating his requirements in natural spoken language. The proposed system will hence help the users to not worry about the coding aspect as the desired webpages will be automatically generated by first fetching the user input in the form of speech, understanding the natural language speech of the user, extracting the necessary data and then correspondingly generating the required webpage. This automated system can thus save a lot of time of practically any user to design webpages.

Keywords— Human Computer Interaction, Natural Language Processing, Automatic code generation, Webpage generation, Text understanding, Voice driven systems.

I. INTRODUCTION

The characteristics of web designing consists of various aspects such as the web-visuals, web-technology, web-stuffing and web-based money [1]. Web stuffing is associated to the definite data matters, information and statistics. Web technology delivers the authentic functions of the webpages of a website in the form of reports and dynamic web generation of web contents. Web visuals describe the shape of a website. Web economics offers the functionality to make online transactions.

With the rapidly growing impact of the Internet as a market place for E-commerce, web-page design is also becoming increasingly important as a means of advertising. In particular, for complex, difficult, or frequent routine design tasks support tools are desirable. In order to benefit from the practicality of natural language processing based interfaces we put forward a system wherein, the basic idea is to not burden the user with the coding aspect, all the user needs is an elementary idea of how he/she wants the website to look like and knowledge of relevant website components to convey the idea to the system, the code is to be generated automatically interpreting from the user's dialogues. The suggested system has been planned around an English speaking user, as the system would take simple voice commands in English and act according to the interpretation.

Manuscript published on 30 November 2015.
*Correspondence Author(s)
Sahil Modak, Department of Computer Engineering, Dwarkadas J. Sanghvi COE, Mumbai, India.
Suril Shah, Department of Computer Engineering, Dwarkadas J. Sanghvi COE, Mumbai, India.
Sagar Vikmani, Department of Computer Engineering, Dwarkadas J. Sanghvi COE, Mumbai, India.
Prof. Lakshmi Kurup, Department of Computer Engineering, Dwarkadas J. Sanghvi COE, Mumbai, India.

The major importance of the proposed system is to deliver an automatic system to the user so that he may himself communicate his requirements and may generate his desired output as webpages. The primary objective was to facilitate development of webpages to the novel users who do not have the technical expertise to do the same. A webpage is basically composition of various types of controls used for both displaying as well as fetching the necessary information [2]. Major html tags often used are table, anchor, lists, buttons, and combo and text boxes. The physical organization of these constituents and the written HTML code functioning behind them is quite a time consuming task. Several components are used for various purposes. These components can better serve their purpose, and webpages too can be more interactive and easy to use if they are all designed adhering to a user's needs.

II. PROBLEM STATEMENT

Websites have fast become the backbone of any business, education or a social organization. Web designing and development have thus seen a comprehensive rise in the technological arena. While, web development is on all-time high, highly skillful and expert personnel are required to design the webpages. And conventional styles of designing webpages have not only proven to be tedious and time consuming, but also have been known to be prone to human induced programming errors. Also, a website can be really unproductive and futile if the user requirements are not satisfactorily met.

III. PROBLEM SOLUTION

In order to cross the aforementioned barriers, the proposed solution aims to put forward a system that would dynamically build webpages driven by a user's linguistic commands. To generate the pages, a particular user only has to provide his specific necessities in simple English dialogues. The planned system interprets them as commands, converts them into text, comprehends the obtained text, digs out the required information and then eventually generates the webpages. User can use them with proper adjustments. Thus, the automated system eliminates the need of manual coding by field experts and related mistakes. Besides, the webpages can be more freely designed by any individual with a basic idea of the layout and structure of the webpage in accordance with his requirements.

IV. RELATED WORK

Since the last few years researches have been initiated in the automatic web layout generation field. Visual interface layout has been the major area of interest [3]. The study in visual interface layout design came into presence with the dawn of new visual applications as web layout and graphical UI for software systems. ADDI, UIDE [4] are some of the few examples.

A number of methods and procedures have been defined to address automatic web-layout generation problems. To solve this problem, some applications also tend to use the visual knowledge base [5]. Almost all related work in this area has been done from the standpoint of assisting the experts. Not much work for the novices has come to light. A non-expert individual cannot really use these intimated software applications. While a few of the recent proposed solutions [6] have been centered around inexpert users, but the scope has been limited to only a specific component of a webpage and the methodology too has been more static than desired.

V. Natural Language Processing Techniques

A. Pattern Matching

The essence of the pattern matching [10] approach to natural language analysis is to infer the input expressions all together, rather than constructing their interpretation by merging the arrangement and connotation of words or other lower level components. The approach is thus holistic rather than constructive. The interpretations with this approach are attained by comparing patterns of words against the input utterance. Associated with each pattern is an interpretation, so that the derived interpretation is the one attached to the pattern that matched. In the simplest case, this arrangement is simply a list of correspondences between equivalence classes of utterances (the ones that match a given pattern) and interpretations (the ones associated with each pattern).

Unless a very shallow level of analysis is acceptable, the number of patterns required is too large, even for restricted domains. This problem can be overcome by hierarchical pattern matching in which the input is progressively canonicalized via matching pattern against sub phrases. Matching semantic primitives instead of words can also reduce the number of patterns. In more sophisticated variations of the approach, patterns may involve higher-level constituents or semantic elements, so that some aspects of the interpretation may become constructive.

B. Syntactically-Driven Parsing

Syntax deals with the way words fit together to form higher-level units such as phrases, clauses, and sentences. Syntactically-driven parsing [10] is, therefore, naturally constructive, i.e. the interpretations of larger groups of word s are built up out of the interpretations of their syntactic constituent words or phrases. The most natural way for syntactically-driven parsing to operate is to construct a complete syntactic analysis of the input utterance first, and only then to construct the internal representation or interpretation.

1) Parse trees and context-free grammars

The most common form of syntactic analysis is known as a parse tree. The below figure shows a parse tree for the sentence:

The rabbit nibbled the carrot.

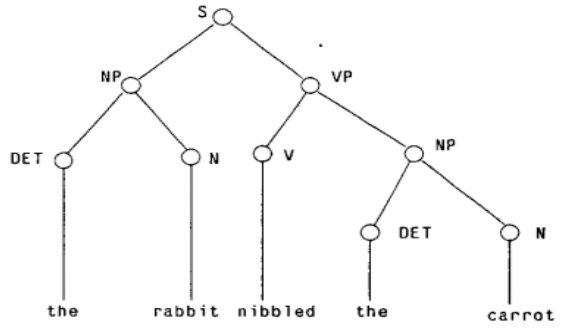

Fig. 1. Parse tree example

The tree shows that the sentence is composed of a noun phrase (subject) and a verb phrase (predicate). The noun phrase consists of a determiner (the) followed by a noun (rabbit), while the verb phrase consists of a verb (nibbled) followed by another noun phrase (the direct object), whose determiner is "the" and whose noun is "carrot". Syntactic evaluations are obtained by applying a grammar that defines what sentences are legitimate in the language being parsed. The method of applying the grammar to the input is called a parsing mechanism or parsing algorithm.

A very simple style of grammar is called a context-free grammar, which consists of rewrite rules of the following form:

S -> NP VP
NP -> DET N | DET ADJ N
VP -> V NP
DET -> the
ADJ -> big | green
N -> rabbit | rabbits | carrot
V-> nibbled | nibbled | nibble

As this example shows, context-free grammars have the advantage of being simple to define. They have been widely used for computer languages, and highly efficient parsing mechanisms have been developed to apply them to their input. However, they also suffer from some severe disadvantages such as:

The rabbits nibbles the carrot

The problem here is that the context-free nature of the grammar does not allow agreements such as the one required in English between subject and object. Moreover, a grammar which also allowed passive sentences such as:

The carrot was nibbled by the rabbit

would have to have another completely different set of rules, even though the passive and the active forms of the same sentence have a clear syntactic relation, not to mention semantic equivalence.

2) Transformational grammar

The problems mentioned above as specific to context-free grammars were tackled by linguists, in particular Chomsky through Transformational Grammar [7]. Their answer was to add another type of rule to a context-free grammar. The basic idea was to use the context free grammar to generate a parse tree just as before, but add onto it certain tags, such as one for a plural sentence.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The set of obligatory transformations on the parse tree would then rearrange things so that the 'pluralness' was transmitted to all parts of the tree concerned and the required agreements could be enforced.

While transformational grammar did a much better job of accounting for the regularities of natural language than context-free grammar, from the point of view of computational effectiveness it was much worse. Consequently, parsers based on transformational grammar have not played a major role in natural language processing.

3) *Augmented transition networks*

Largely in response to the problems of transformational grammar, Bobrow and Fraser [8] proposed and Woods [9] subsequently developed a method of expressing a syntactic grammar that was computationally tractable and yet still could capture linguistic generalizations in a concise way, in many cases more concisely than transformational grammar itself. The formalism Woods developed was known as an augmented transition network or ATN. It consisted of a recursive transition network (formally equivalent in expressive power to a context-free grammar), augmented by a set of tests to be satisfied before an arc was traversed and a set of registers that could be used to save intermediate results or global state.

VI. PROPOSED METHODOLOGY

Researching all the available tools and techniques to carry out the various stages of the project, we have proposed the methodology we wish to employ in our solution.

A. Speech Acquisition: The user speech is to be considered as commands for the system. These commands would be captured via a microphone enabled computer machine. As the user starts speaking, the speech would be simultaneously acquired by the system.

B. Text conversion: The speech of the user captured by the system would be concurrently converted to text. Hidden Markov Model is intended to be used for the conversion. Also, the converted text phrase would be displayed to the user to ensure accurate conversion, in case of anomalies the user has the liberty to pause, stop or resume the process.

C. Text processing: The text converted from speech would then be passed through a series of processing steps:

1) ***Parsing***, to parse words in a sentence and gives the tag based on the rules of the grammar.

2) ***Part-Of-Speech (POS) Tagging*** to tag every word that has been parsed in a sentence based on Brown tag set or Penn Treebank tag set in order to obtain the relationship between words. The tags that are given are more complex and detailed than the tag by the parser, so that POS Tagger can distinguish variations in verb forms, variations of singular and plural nouns, and conjunctions variations.

3) ***Named Entity Recognizing (NER)***, is a technique that is used to identify a noun word and able to acknowledge whether it is a name or a number, etc.

Post these steps, dependency parsing is intended to be employed which would help analyze the grammatical structure of a sentence and establish relationships between "head" words and words which modify those heads to better interpret the natural language texts.

D. Knowledge Extraction: This is the basic understanding module where the system would interpret the requirements of the user from the processed text. Once the dependencies are obtained the corresponding keywords found according to the parts of speech would be searched in a maintained dictionary, wherein, if the keyword is present, a mapping for the keyword to its corresponding HTML tags would be obtained; and if not present a suitable error would be displayed.

E. Code generation: This step uses the extracted information as an input. The mapping information from the above step would then be used to generate HTML/CSS codes. Once the user instructs the system, the generated code would be then saved with an '.html' extension with the styling interpreted from the texts, implemented as inline CSS. The user has the freedom to see the code files and make manual changes in the code.

F. ***Template creation:*** All the user's previous works are saved and constantly referenced to observe any kind of similarity to previous works. Based on the similarity measures, the system suggests predictions to the users to use these previously saved files as a template to edit and work upon, saving time consumptions.

After successfully saving, the file would be run in a browser window for the user to view the results, based on which the user could suggest further editing or command the system to save or delete the current work and exit.

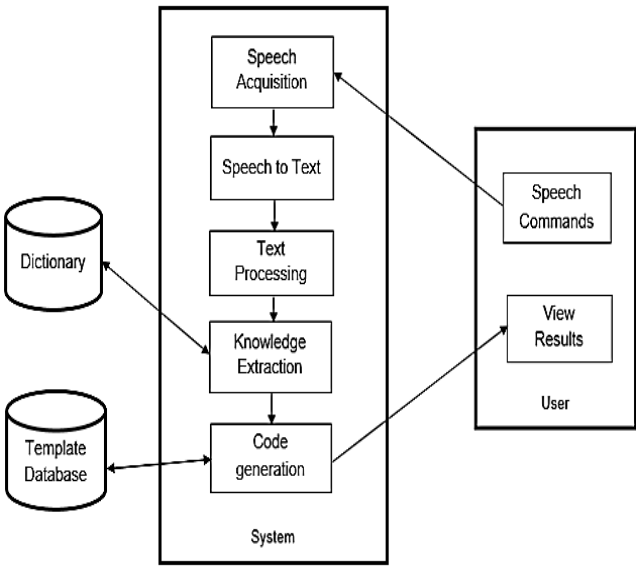

Fig. 2. System architecture design

VII. CONCLUSION

A natural language processing based system for dynamically building websites is proposed in this document. The system would be driven by users' speech, which will be treated as commands to the system. The speech after having converted to text, will be pre-processed and then processed to extract information from the speech.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

On the basis of the knowledge interpreted, code will be generated to build static websites centered on HTML and inline CSS. Providing predictive templates would help reduce time and effort consumption. Our proposed system, currently in the implementation phase, aims to achieve low computational complexity and very high accuracy for a limited and essential set of HTML tags at the beginning. The system will be able to provide a smooth interface for any user to interact with the computer to build functional webpages.

VIII. FUTURE SCOPE

With an enormous future scope, the proposed scheme provides vast researching avenues not only in the field of Natural Language Processing but also in the fields of Human Computer Interaction, Artificial Intelligence and Automated Code Generation. Future scope in this domain would be to develop a software that can predict and suggest templates more accurately using machine learning and neural networking. Also, a language independent system could be a further development prospect. More efficient natural language processing can be introduced with the help of neural networks develop a more intelligent and reliable system. Bringing in more dynamicity with more languages like PHP, JavaScript, etc., different database models and various frameworks like ASP.Net. This software can be used in institutions like schools where deaf or handicapped users can learn to build websites. Besides, currently the system is being developed as a desktop application, further versions can be developed to make the system web based and completely mobile.

REFERENCES

1. Imran S. Bajwa, M. Asif Naeem, Riaz-Ul-Amin, M A. Choudhary, Speech Language Processing Interface for Object-Oriented Application Design using a Rulebased Framework, 4th International Conference on Computer Applications 2006 Rangoon, Myanmar
2. Malaisé Véronique, Zweigenbaum Pierre, Bachimont Bruno, Mining Defining Contexts to Help Structuring Differential Ontologies Terminology, 11:1, 2005
3. A. R. Ahmad, O.Basir, K.Hassanein, “Fuzzy Inferencing in the Web Page Layout Design”, Proc. of the 1st Workshop on Web Services: Modeling, Architec. & Infrastructure, France, pp. 33-41, April 2003.
4. J. Foley, W. Kim, S. Kovacevic, and K. Murray, “UIDE-An Intelligent User Interface Design Environment”, In J W Sullivan and S.W. Taylor (Eds.), Intelligent User Interface, ACM, NY, 1991
5. A.R. Ahmad, O. Basir, K. Hassanein, “Efficient Placement Heuristics for Genetic Algorithm based Layout Optimization”, Working Paper, Systems Design Engineering, University of Waterloo, 2003m.
6. Bajwa, Imran Sarwar, Waqar Aslam, and Syed Irfan Hyder. "Speech Language Engineering System for Automatic Generation of Web based User Forms."
7. Chomsky, N. Aspects of the Theory of Syntax. MIT Press, 1965.
8. Bobrow, D. G. and Fraser, J. B. An Augmented State Transition Network Analysis Procedure. Proc. Int. Jt. Conf. on Artificial Intelligence, Washington, D. C, 1969, pp. 557-567.
9. Woods, W. A. "Transition Networ k Grammar s for Natural Languag e Analysis." Comm. ACM 73, 10 (Oct. 1970), 591-606.
10. Hayes, Philip J., and Jaime Guillermo Carbonell. "A tutorial on techniques and applications for natural language processing." (1983).W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)