Spoken WordNet

Kishore Kashyap
Department of Information Technology
Gauhati University, India
kb.guwahati@gmail.com

Shikhar Kr Sarma
Department of Information Technology
Gauhati University, India
sks001@gmail.com

Kumari Sweta
Department of Information Technology
Gauhati University, India
swetagupta647@gmail.com

Abstract

WordNets have been used in a wide variety of applications, including in design and development of intelligent and human assisting systems. Although WordNet was initially developed as an online lexical database, (Miller, 1995 and Fellbaum, 1998) later developments have inspired using WordNet database as resources in NLP applications, Language Technology developments, and as sources of structured learned materials. This paper proposes, conceptualizes, designs, and develops a voice enabled information retrieval system, facilitating WordNet knowledge presentation in a spoken format, based on a spoken query. In practice, the work converts the WordNet resource into a structured voiced based knowledge extraction system, where a spoken query is processed in a pipeline, and then extracting the relevant WordNet resources, structuring through another process pipeline, and then presented in spoken format. Thus the system facilitates a speech interface to the existing WordNet and we named the system as "Spoken WordNet". The system interacts with two interfaces, one designed and developed for Web, and the other as an App interface for smartphone. This is also a kind of restructuring the WordNet as a friendly version for visually challenged users. User can input query string in the form of spoken English sentence or word. Jaccard Similarity is calculated between the input sentence and the synset definitions. The one with highest similarity score is taken as the synset of interest among multiple available synsets. User is also prompted to choose a contextual synset, in case of ambiguities.

1. Introduction

WordNets have become resources for many NLP applications, language technology developments, as well as a knowledge database (Morato, 2004). This is different from any other form of information storage as in WordNet words are stored in a way where different word forms (sysnsets) semantically linked. Many semantic relations are embedded in WordNet, and lexical units are defined and described with examples, concepts, and synonyms. Thus making the database a resourceful lexico-semantic knowledge base.

In this paper. we have developed a new voice based system integrating a new system pipeline for processing spoken query, and presenting retrieved knowledge also in spoken format. This has enabled usage of the textual lexico-semantic knowledge base as a spoken knowledge base, thus creating the new concept of Spoken WordNet. The intelligent spoken language interfacing technique is already been explored (Inagaki, 2013). The relevance of voiced enabled interface for information extraction has become more intense in recent years, and user base of voice interfaces is growing. Also, for visually challenged persons, voice based interface, and spoken knowledge presentation signifies a lot. For our current work, we used the Princeton English WordNet.

Query in the form of short sentence, or discrete word is inputted through the voice based interface, either the Web version, or the App version. For experimental and demonstration purpose, we limited the scope of work to Nouns only. Spoken query is processed for extracting/ identifying words. Then it undergoes a series of computational steps, ultimately defining the token of interest in the form of text. This token of

interest is then subjected to the retrieval process into the main WordNet database and required information is identified and extracted from WordNet. At this moment, we also mine to depth 1 of hypernym, extracting upper layer knowledge. This is now put in another process pipeline for structuring the presentable knowledge. Predefined format is used to embed the extracted segments, formatting new sentence level presentations. And ultimately such structured and formatted textual sentences are presented through the interface in spoken format.

2. The Algorithm and the Core Engine

Algorithm 1: Algorithm

Input: sentence **S**
Output: Word description and hypernym of depth 1

```
[S_t] = Tokenized S
[W] = POS tagged words of [S_t]
retrieve W_i from wordnet database where W_i ∈ [W], W_i = Noun
begin do ∀W_i ∈ [W]
    if N = 0, where N is number of synsets found for W_i
        speak "No synset found for W_i"
        goto step 4
    else
        if 1 < N
            get highest Jaccard Similarity from N W_i definition and S
            if highest Jaccard Similarity found
                speak W_i definition, W_i hypernym of depth 1
                goto step 4
            else
                list multiple synsets and prompt user to select one synset
                get selected synset and goto step 12
        else
            goto step 12
end do
```

FIGURE 1

In this section we will discuss the overall system model. The speech signal is taken as input from the user and send over the Internet in real time to convert it into text using open source Speech-To-Text. Upon receiving the text data, the core engine of Spoken WordNet works as the algorithm (figure 1) we have developed.

The Jaccard similarity is calculated as:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

(If A and B are both empty, we define $J(A,B) = 1$.)

$$0 \leq J(A,B) \leq 1.$$

where, A is the set of words in the input sentence and B is the set of words in the synset definition. Both A and B are free of English stop words.

Here, Jaccard Similarity is used for word sense disambiguation (WSD).

The core engine is entirely written in Python3 and NLTK (Bird, 2004) is used for WordNet information retrieval. For POS tagging TextBlob (https://textblob.readthedocs.io/en/dev/) package is used. English stop words are removed using NLTK.

3. User Interface Design and Implementation

It is essentially very important in today's technological use case scenario that a great deal of usage of any successful software product should include a good and user friendly UI. Now, users are more Internet centric as compared to users a decade ago. So, the authors decided to develop two important areas of Human Computer Interaction UI. Namely Smartphone App and Web Interface.

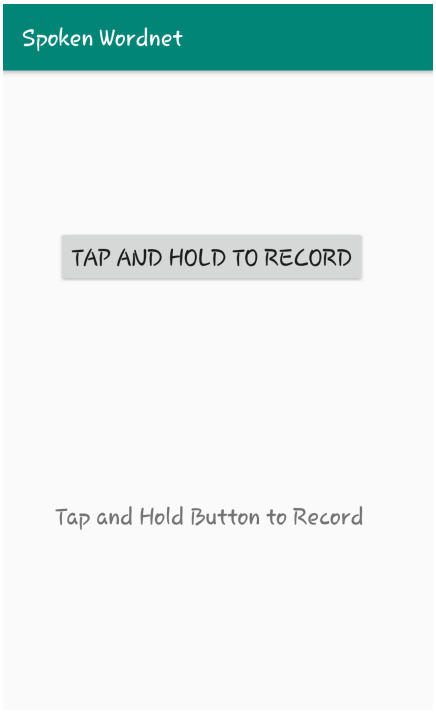

FIGURE 2

3.1. Smartphone App

People are becoming more interested in App based information retrieval, because of user friendliness and readymade linking to the host servers, and cloud based services. But spoken interfaces are very limited and there are no substantial evidence of such an existing system. Although access to WordNets are free, and many APIs including APIs for smartphones have been developed, but we have not seen any smartphone App for voiced based connection to WordNet. As WordNet contain knowledge, and also the WordNets are scalable, an App based Voice Interface with linking and structured retrieving of knowledge will facilitate a wide range of users. This is more user friendly, as query is in the form of voice prompts, and also significantly important to visually challenged persons. App is designed and developed using Android Studio and Firebase. Speech signal from user is taped using the Smartphone microphone, and then the App module temporarily stores it and then sends to the Firebase cloud server. The voice segment is stored as a .wav file in the Firebase server. This file is used for processing in our core system pipeline. The structured and formatted output from the core processing is then send back to the server, and the App accesses the server, takes the file, and reproduces the file into spoken format. Thus, for user, it's a voice-input-voice-output system, creating the App for Spoken WordNet.

3.2. Web Interface Design and Implementation

We have developed a Web Interface for the system which currently runs in the Google chrome browser. User can click on the main interface and then start speaking. The Speech To Text conversion and Text To Speech Conversion is done using the JavaScript Web Speech API.

Flask is used for web server creation. Flask is a micro framework for python web development

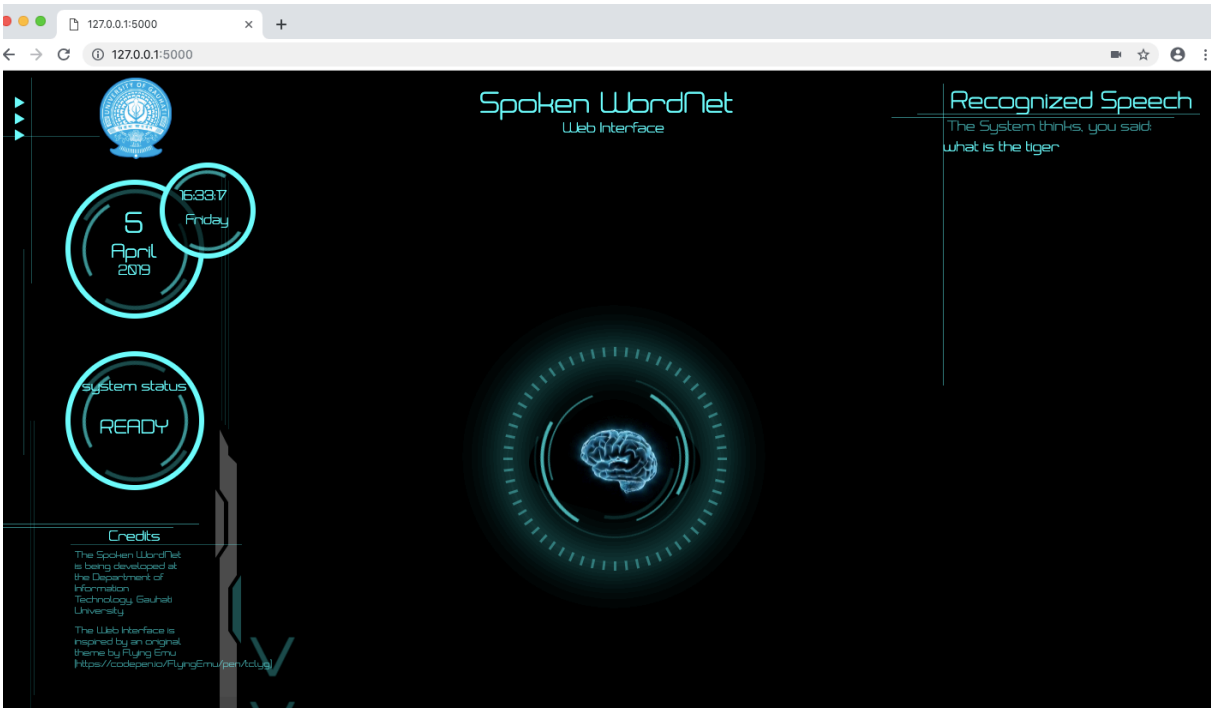

FIGURE 3

(Ronacher http://mitsuhiko.pocoo.org/flask-pycon-2011.pdf).
For message passing to and from the web interface to the core engine is done through SocketIO. Socket.IO enables real-time, bidirectional and event-based communication.

4. Result

The system is working well for nouns as we have limited our concentration to the nouns POS. We have also experiments with the Lesk WSD algorithm available in NLTK. But, Jaccard Similarity is found to be working well for nouns and having more than one words in the query sentence, e.g., knowing what words are to be given as input for extracting the correct sense of the word.

5. Future Work

We have presented a system of voiced based interface for the WordNet in both Smartphone App and Web environment. In this prototyping work, we have considered only nouns for their semantic extraction from WordNet. In future, this can be extended for other parts of speech. Furthermore, it has scope for experimenting with other sentence similarity measures for WSD. Exploring the word expansion for simplifying spoken text may be another future direction of research through this system.

References

Armin Ronacher, Opening the Falsk, URL:http://mitsuhiko.pocoo.org/flask-pycon-2011.pdf

Christiane Fellbaum 1998, ed. WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press.

George A. Miller 1995, WordNet: A Lexical Database for English. Communications of the ACM Vol. 38, No. 11: 39-41.

Hirohito Inagaki, Takaaki Hasegawa, Satoshi Takahashi, and Yoshihiro Matsuo 2013, Toward Intelligent Spoken Language Interface Technology, NTT Technical Review, Vol. 11 No. 7

Jorge Morato, Miguel Angel Marzal, Juan Lloréns, and José Moreiro, 2004, WordNet Applications, Proceedings of GWC2004, pp. 270–278.

Steven Bird and Edward Loper. 2004. NLTK: the natural language toolkit. In Proceedings of the ACL 2004 on Interactive poster and demonstration sessions (ACLdemo '04). Association for Computational Linguistics, Stroudsburg, PA, USA, , Article 31 . DOI=http://dx.doi.org/10.3115/1219044.1219075