Sophisticated Lexical Databases - Simplified Usage: Mobile Applications and Browser Plugins For Wordnets

Abstract

India is a country with 22 officially recognized languages and 17 of these have WordNets, a crucial resource. Web browser based interfaces are available for these WordNets, but are not suited for mobile devices which deters people from effectively using this resource. We present our initial work on developing mobile applications and browser extensions to access WordNets for Indian Languages.

Our contribution is two fold: (1) We develop mobile applications for the Android, iOS and Windows Phone OS platforms for Hindi, Marathi and Sanskrit WordNets which allow users to search for words and obtain more information along with their translations in English and other Indian languages. (2) We also develop browser extensions for English, Hindi, Marathi, and Sanskrit WordNets, for both Mozilla Firefox, and Google Chrome. We believe that such applications can be quite helpful in a classroom scenario, where students would be able to access the WordNets as dictionaries as well as lexical knowledge bases. This can help in overcoming the language barrier along with furthering language understanding.

1 Introduction

India is among the topmost countries in the world with massive language diversity. According to a recent census in 2001, there are 1,365 rationalized mother tongues, 234 identifiable mother-tongues and 122 major languages. Of these, 29 languages have more than a million native speakers, 60 have more than 100,000 and 122 have more than 10,000 native speakers. With this in mind, the construction of the Indian WordNets, the IndoWordNet (Bhattacharyya, 2010) project was initiated which was an effort undertaken by over 12 educational and research institutes headed by IIT Bombay. Indian WordNets were inspired by the pioneering work of Princeton WordNet (Fellbaum, 1998) and currently, there exist WordNets for 17 Indian languages with the smallest one having around 14,900 synsets and the largest one being Hindi with 39,034 synsets and 100,705 unique words. Each WordNet is accessible by web interfaces amongst which Hindi WordNet(Dipak et al., 2002), Marathi WordNet and Sanskrit WordNet(Kulkarni et al., 2010) were developed at IIT Bombay. The WordNets are updated daily which are reflected on the websites the next day. We have developed mobile applications for the Hindi, Marathi and Sanskrit WordNets, which are the first of their kind to the best of our knowledge.

This paper is organized as follows: Section 2 gives the motivations for the work. Section 3 contains the descriptions of the application with screen-shots and the nitty gritties. We describe the browser extensions in Section 4, and we conclude the paper with conclusions, and future work in Section 5. At the very end, some screen-shots of the applications and browser extensions are provided.

1http://en.wikipedia.org/wiki/Languages_of_India
2http://www.cfilt.iitb.ac.in/
2 Motivation

According to recent statistics, about 117 million Indians\(^3\), are connected to the Internet through mobile devices. It is common knowledge that websites like Facebook, Twitter, Linkedin, Gmail and so on can be accessed using their web browser based interfaces but the mobile applications developed for them are much more popular. This is a clear indicator that browser based interfaces are inconvenient which was the main motivation behind our work. We studied the existing interfaces and the WordNet databases and developed applications for Android, iOS and Windows Phone platforms, which we have extensively tested and plan to release them to the public as soon as possible.

Our applications and plugins are applicable in the following use cases:

1. Consider an educational classroom scenario, where students, often belonging to different cultural and linguistic background wish to learn languages. They would be able to access the WordNets as dictionaries for multiple Indian languages. This would help overcome the language barrier which often hinders communication, and thus, understandability. The cost effective and readily available “Aakash” tablet device\(^4\) will be one of the means by which our application will be accessed by educational institutes over India.

2. Tourists traveling to India can use the WordNet mobile apps for basic survival communication, because Indian language WordNets contain a lot of culture and language specific concepts, meanings for which may not even be available on internet search.

3. People who read articles on the internet may come across words they do not understand and can benefit from our plugins which can help translate words and give detailed information about them at the click of a button.

4. Linguists who happen to be experts at lexical knowledge can use the WordNet apps as well as plugins to acquire said knowledge irrespective of whether they have mobile phones or PCs.

Apart from the cases mentioned above, there are many other cases where our apps and plugins can be used effectively.

3 Mobile WordNet Applications

In the subsections below we describe the features of the applications accompanied by screen-shots.

3.1 Home Screen

When the user starts the application, the home screen (Figure: 1) is shown with a brief description of how to use it, the link which takes the user to search interface.

3.2 Search Interface

We have provided the user with two types of input mechanisms, Phonetic Translitera-
tion using Google Transliteration API\(^5\), and a JavaScript based online keyboard (Figure: 2) for input of Hindi Unicode characters. Transliteration for a native user is very convenient. In case, the user does not know the right combination of keys then the keyboard for Devanagari is provided. These two methods ensure that all words can be easily entered for searching. Thereafter, by touching / clicking on “Search”, the synsets with all relevant information are retrieved.

### 3.3 Search Process

Indian languages are fairly new to the web, and despite standard UTF encoding of characters, there remain a few steps to be taken to sanitize the input for WordNet search. The steps taken by us are given below:

#### 3.3.1 Nukta Normalization for Hindi

Hindi Characters such as क (ka), ख (kha), ग (ga), ज (ja), ड (ḍa), ढ (ḍha), फ (pha), झ (jha), take up nukta symbol to form क़ (qa), ख़ (kḫa), ग़ (ġa), ज़ (za), ड़ (ṛha), ढ़ (ṛha), फ़ (fa), झ़ (zha), respectively. These characters occur twice in the Unicode chart, both with nukta as a separate unicode character, and adjoining the parent character. We normalize the input for standard unicode encoding with nukta as a separate character before search.

#### 3.3.2 Morphological Analysis

Before searching in the databases the word is first passed to a morphological analyzer to obtain its root form. We use Hindi Morph Analyzer (Bahuguna et al., 2014) to return the root form of the input word for Hindi language, since by principle, WordNet only contains root forms of the words.

Due to non-availability of other language Morphological Analyzers, we may not be able to include them in the search process. Though, in the future, we can use a fully automated version of the “Human mediated TRIE base generic stemmer” (Bhattacharyya et al., 2014) for obtaining root forms for other languages later.

#### 3.3.3 Handling Multiple Root forms

Our interface also requests the user to select the preferred root, if more than one root forms are returned post morphological analysis. The user can then just select one and then the synset retrieval process is initiated on the server. It gives the user more control, and choice over results. We assume that while searching the WordNet, a user may not be familiar with all the senses of the words, or all the morphology of the word. It may be possible that the user came across the word over the internet, and is using our plugin to search the WordNet. This feature enables the user to select the appropriate root, or check all the possibilities for the correct answer.

### 3.4 Application Design

We have used the WebView class, and URL loading from the Android SDK\(^6\), and Windows Phone SDK\(^7\) to display a responsive layout of the WordNets. WebView renders the application pages seamlessly onto the mobile / handheld devices, thus making the application usable for mobile, tablet, and other handheld devices.

---

\(^5\)https://developers.google.com/transliterate/
\(^6\)https://developer.android.com/
\(^7\)https://dev.windows.com/en-us/develop/download-phone-sdk
Similarly, for iOS, we have used the UIWebView class with some scaling measures to render the pages with a responsive layout onto the device screen. Our application is compatible with all iOS devices. It will be deployed to Apple App Store soon.

A preliminary check on internet connection is done before connecting to the web interface, and retry button is provided on the front, in case an internet connection is not detected.

3.5 Search Results

The results returned by the server are interpreted by the application pages and displayed in a very simplistic manner. We display all synsets for each part of speech and all senses of that word and initially showing the synset words, gloss and example. These senses are categorized by their part of speech categories. We have conformed to the principles of good User Interface design and provided for an incremental information display.

3.5.1 Additional Information

If the user wishes to see the synset relations and the translations of that word in other synsets the link “Relations and Languages” should be clicked to give a list of all additional information that can be displayed. Relations like Hypernymy and Hyponymy and the relevant synset in the other 16 languages can be displayed. Please refer to figure 3 for an example.

3.5.2 Current Drawbacks

Current version of Android OS (Lollipop 5.0) deployed on most of the smartphones, does not support rendering of Gujarati, Punjabi, and Nepali languages, on all devices. The language support also depends on the device manufacturer. Hence, they are currently disabled from the interface.

Also, Our applications are currently online, and can only be used if the user is connected to the internet. We plan to implement an offline version of our applications.

4 Browser Extensions

Major WordNets of the world are available via web interfaces, enabling a user to search for the senses using a web browser on a computer or mobile. The process commonly involves a user navigating to a web page, and searching the required ‘word’ for its senses. In a world where getting things done in one click is important, we feel that the process of searching needs to be simplified. We develop browser extensions to ease this process. Google Chrome and Mozilla Firefox are the most popular web browsers among the web users\(^8\). Our approach makes the search quite simple and is summarized in the following 3 steps:

- User highlights the word of interest and right-clicks the page or clicks on the plugin shortcut.
- They click the context menu option for ‘Search <relevant> WordNet for . . .’
- A new tab opens up showing the information from the relevant WordNet.

We present the sample context menu screenshots, post installation in Figures 6 and 7, respectively.

5 Conclusions and Future Work

In this era of handheld mobile devices, there is a great need to make available traditional web services as mobile applications which are extremely popular. Our success in developing mobile applications for Hindi, Marathi and Sanskrit WordNets along with browser plugins for English, Hindi, Marathi and Sanskrit to simplify word look-up is the first step in providing people with easy access to such important knowledge bases. We have described a variety of use cases for our apps and plugins which are quite realistic, especially in India where language and cultural diversity is quite varied. These can have a huge impact on language education, especially in the rural areas, along with enabling people to understand a multitude of languages.

We plan to make available offline search in our apps. Also, we plan to make efforts towards improving this application to enable searching for words belonging to all languages which have a common interface via language detection. We also plan to inculcate Word Suggestions as they are being typed so that the

\(^8\)http://gs.statcounter.com/#all-browser-ww-monthly-201506-201506-bar
user is presented with better lexical choices. Plugins like PeraPera\(^9\) for Japanese and Chinese are quite popular since they simply provide lexical information when the user hovers the mouse over words. Implementing such a feature is something we plan to do in the immediate future. Also, We would publish our application, and browser plugin source codes publicly for research purposes.

6 Acknowledgment

We gratefully acknowledge the support of the Department of Electronics and Information Technology, Ministry of Communications and IT, Government of India. We also thank Ravi Nambudripad, for replicating the application in other languages and the entire computational linguistics group at Centre For Indian Language Technology, IIT Bombay, which has provided its valuable input and critique, helping us refine our work.

References

Dipak Narayan, Debasri Chakrabarti, Prabhakar Pande and Pushpak Bhattacharyya. 2002. An Experience in Building the Indo WordNet - a WordNet for Hindi. In First International Conference on Global WordNet, (GWC 2002), Mysore, India.

Christiane D. Fellbaum. 1998. WordNet: An Electronic Lexical Database. Published by Bradford Books

Pushpak Bhattacharyya, 2010. IndoWordNet. In Proceedings of Lexical Resources Engineering Conference, May, 2010, Malta.

Ankit Bahuguna, Lavita Talukdar, Pushpak Bhattacharyya and Smriti Singh. 2014. HinMA: Distributed Morphology based Hindi Morphological Analyzer. In Proceedings of the 11th International Conference on Natural Language Processing (ICON 2014), December, 2014.

Pushpak Bhattacharyya, Ankit Bahuguna, Lavita Talukdar, and Borauli Phukon. 2014. Facilitated Multi-Lingual Sense Annotation: Human Mediated Lemmatizer. In Proceedings of the Global Wordnet Conference 2014 (GWC 2014), January, 2014.

Malhar Kulkarni, Chaitali Dangarikar, Irawati Kulkarni, Abhishek Nanda, and Pushpak Bhattacharyya. 2010. Introducing Sanskrit Wordnet. In Proceedings of the Global Wordnet Conference 2010 (GWC 2010), January, 2010.

\(^9\)http://www.perapera.org/
Figure 4: Search Results with Malayalam, Tamil, and Telugu Synsets

Figure 5: Search Results with English, Bengali, and Marathi Synsets

Figure 6: Browser Extensions Context Menu for word ‘specific’

Figure 7: Browser Extensions Context Menu for word ‘हिसा’ (hiMsaa) translated to ‘violence’