

A Study on Development of Multilingual Dictionary Software

Dr.J.VijiPriya ¹, Dr.S.Suppiah ², Ms. Adeela Ashraf ³

¹ Assistant Professor, College of Computer Science and Engineering, University of Hail, Hail, Saudi Arabia

² Dean, VelTech University, TamilNadu, India

³ Lecturer, College of Computer Science and Engineering, University of Hail, Hail, Saudi Arabia

vijipriyajkv@gmail.com ¹, apravi31@gmail.com ², a.raja@uoh.edu.sa ³

Abstract:- Machine Translation is global. Recently there is no global business without translation. Clients buy and sell everything online. Although there are various researchers to build translators, several research papers deal with design and implementation of Online Website Electronics Dictionaries, not Multilingual Dictionary software in any one of programming languages. As a result, there are many challenges to developers and researchers such as selection of programming language to develop machine translation software. This study presents the design and implementation of Multilingual Dictionary Software in Python. It motivates developers to build many more machine translators. This Multilingual Dictionary is tremendously useful to people who want a fast translation.

Keywords:- Machine translation, Python, Multilingual, Dictionary, Text to Speech.

1. Introduction

Language is a system for communicating human ideas, desires, and emotions using signals, such as voice sounds, gestures, and symbol. Linguistics is a study of language. Evolutionists conclude that man started language in Africa. On the other hand, linguists conclude that languages have Indo-European roots. The center of the linguistic region is Anatolian very near the Biblical land of Shinar ⁸. Currently, there are 9000 living languages around the world.

Machine translation (MT) is intelligent automated software capable of translating one source language and multiple languages (multilingual). Machine Translation systems may be proposed for either two languages (bilingual) or more than one language (multilingual). Translation companies have developed machine translation for web publishers to place it on their sites. It is easy for machine visitor to switch between human languages ⁷.

Different electronic translation tools can be divided into three categories ¹¹:

1. MT systems- for personal, Internet/web and professional use.
2. Translation support tools- for Electronic dictionaries (mono or bilingual, technical or encyclopedic)
3. Online systems- for MT systems (e.g., translation service via the Internet/ mobile phone, with or without human post-editing) and MT portal (e.g., services on the Internet providing access to some MT services and information about MT systems).

Most industries around the world need language services. Law enforcement, medical notes and records, Court proceedings, court records, petitions, Media, insurance companies, financial companies, and companies with web presence all require translation in one form or another. Moreover, Individuals, mostly those who travel

to foreign where they do not have their native language, they need a transformation of their documents such as passports, birth certificates, marriage licenses, and divorce papers ⁷.

A Dictionary is a resource dealing with the individual words of a language along with its orthography, usage, synonyms, derivation, pronunciation, etc. ordered for ease of referencing the phrase. Common types of dictionaries are Encyclopedia, Thesaurus, Etymological Dictionary, Dialect Dictionary, Specialized Dictionary, Bilingual or Multilingual Dictionary, Learner's Dictionary, Reverse Dictionary, Phonetic Dictionary, and Visual Dictionary. The proposed work relies on Thesaurus and Multilingual Dictionary ¹⁰.

On demand of application of MT, English, and Multilingual Dictionary (EM Dict) has been developed. This research paper describes EM Dict, its design, and implementation to inspire MT developers. The rest of the paper is organized as follows: Section 2 describes methodologies used in EM Dict. Section 3 details components of EM Dict, followed by its design and layout. Section 4 gives the result of the dictionary. Section 5 presents conclusion, scope, and future enhancements.

2. Methodology

This section depicts methodology to plan, and execution of EM Dict incorporates some of the python packages Tkinter, ttk, go slate, tty, PyDictionary, and pyttsx. Tkinter and ttk offer various controls, such as buttons, labels and text boxes used in Layout of EM Dict. The goslate is a single file module Goslate().translate(sentence, target language id) and provides rapid google translation service without fee by querying google translation website ². PyDictionary is a Dictionary Module to obtain meanings, synonyms, Antonyms and Translations of words. WordNet was used for getting meanings, Google for translations, and thesaurus.com for getting synonyms and antonyms ¹. pyttsx supporting text-to-speech engines on Windows, Mac OS X, and Linux ³. The other python packages are distutils.Core, setup, py2exe, sys, and os used in translate from python code emdict.py to emdict.exe of EM Dict. Finally, Inno Setup compiler was used to convert English and Multilingual Dictionary as software to be installed on all windows.

3. Modules of EM Dict

The proposed English to Multilingual Dictionary provides Synonyms, Antonyms, Verbs, Nouns, Pronunciation and Multilingual Translation. Users can enter an English word and select target language; as a result, they observe it is all Synonyms, Antonyms, Verbs, Nouns in English and other selected target language. It supports more than 105 languages and pronunciation. It can convert text to speech. It runs on windows. When using this Multilingual Dictionary, Users need Internet connection ^{5,6}.

Python is a popular scientific language for machine learning. A newly written Python program customizes English to Multilingual Dictionary components are depicted in Figure 1.

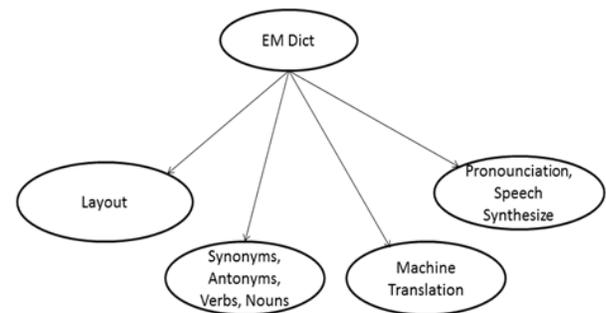


Figure 1. Components of EM Dict

These components are considered as a module of this study. As a result, the EM Dict was divided into chapters: Synonyms, Antonyms, Verbs and Nouns, Multilingual Translation, Pronunciation, and Speech Synthesizes. The following chapters ornately describe the implementation of these modules.

3.1 EM Dict Layout

This section describes the design of EM Dict from Standard Python GUI Libraries. The modules Tkinter and ttk provide a fast and powerful object-oriented interface to the Tk GUI toolkit ⁴. Figure 2 shows Python code for EM Dict Layout.

The following steps are executed to create GUI applications using Tkinter and ttk:

1. Import the Tkinter and ttk modules.
2. Call Tkinter.Tk() to create the GUI application main window.
3. Add one or more widgets: Frame, Label, Entry box, Button, Combobox and Textbox to the main window.
4. Add properties to all widgets.

5. Call `widget.grid(row, column)` to arrange all widgets on the main window.
6. Enter the main event loop `window.mainloop()` to take action against each event triggered by the user.

3.2 Synonyms, Antonyms, Verbs and Nouns

This chapter describes the implementation of Dictionary modules includes Synonyms, Antonyms, Verbs, and Nouns is shown in Figure 3. The following essential steps have been in employ to the development of EM Dict modules ¹:

1. Installation of PyDictionary module
2. Import PyDictionary module
3. Creation of dictionary instance from PyDictionary module
4. To return Synonyms of a word, Call `dictionary.synonym(word)`
5. To return Antonyms of a word, call a `dictionary.Antonym(word)`

6. To return Verb and Noun of a sentence, call `dic1=dictionary.Meaning(word)`, `dic1['Verb']` and `dic1['Noun']`

3.3 Multilingual Translation

Goslate achieves multilingual translation, a free python API querying google translation ². The Python code for Multilingual Translation is shown in Figure 4. The native language translation was executed by the following steps:

1. Installation of goslate: `pip install goslate`
2. Import goslate module
3. Create instance `gs= goslate.Goslate()`
4. Call function `gs.get_languages()` to get list of native languages including language name and id
5. Call method `sorted()` to sort language list
6. Make use of for loop to iterate language id and name
7. Finally call method `gs.translate(word to be translated, target language id)`

```
MainWindow
window=Tkinter.Tk()
Frame
frame1=Frame(window,bd=1,bg="#ff0000")
Label
var=StringVar()
label1=Label(frame1,textvariable=var,bg="#ff0000",fg="#ffffff",height=2)
var.set("Welcome to JKVSRG English to Multilingual Dictionary")
Entry Box
entrybox1=Entry(frame1,bd=1,width=30,bg="#ff9d9d",fg="Blue")\
entrybox1.insert(0,"Type a word to search")
entrybox1.bind('<Button-1>', OnEntry1)
Combo Box
combostyle = ttk.Style()
cvar1=StringVar()
Cbox1=ttk.Combobox(frame1, textvariable=cvar1,width=20,values=lang_name)
cvar1.set("Select Language")
Cbox1.bind("<<ComboboxSelected>>", OnSelect1)
Textbox
text1=Text(frame2,bd=1,width=50,height=10,bg="#ff9d9d",fg="#000000")
text1.grid(row=2,column=0)
Button
button1=Button(frame1,text="Go", command=OnClick,bg="#ff0000",fg="#ffffff",cursor='hand2',
activebackground='#e2ffc4',overrelief=SUNKEN,width=5,activeforeground='red',bd=0)
```

Figure 2. Python code for EM Dict Layout

```
Synonyms:
list1=dictionary.synonym(entrybox.get())
for sword in list1:
textbox.insert('0.0', str(sword)+"-"+str(gs.translate(sword,lid).encode('utf8'))+"\n")

Antonyms:
list2=dictionary.antonym(entrybox1.get())
for aword in list2:
    text2.insert('0.0', str(aword)+"-"+str(gs.translate(aword,lid).encode('utf8'))+"\n")

Verbs:
dic1=dictionary.meaning(entrybox1.get())
list3=dic1['Verb']
for vword in list3:
    text3.insert('0.0', str(vword)+"-"+str(gs.translate(vword,lid).encode('utf8'))+"\n")

Nouns:
dic2=dictionary.meaning(entrybox1.get())
list4=dic2['Noun']
for nword in list4:
    text4.insert('0.0', str(nword)+"-"+str(gs.translate(nword,lid).encode('utf8'))+"\n")
```

Figure 3. Python code for EM Dict modules: Synonyms, Antonyms, Verbs and Nouns

```
gs=goslate.Goslate()
languages=gs.get_languages()
lang_id=sorted(languages.keys())
lang_name=sorted(languages.values())
for name in lang_name:
    print name.encode('utf8')
global lid
for key in languages:
    if languages[key]==Cbox1.get():
        lid=key
gs.translate(word, lid)
```

Figure 4. Python code for Multilingual Translation

3.4 Pronunciation and Speech Synthesizers

Native language pronunciation was implemented by using pyttsx, python factory function includes drivers for text-to-speech synthesizers³. Figure 5 shows python code for Pronunciation and Speech Synthesizers. To achieve text-to-speech, the following steps are required:

1. Import module pyttsx
2. Getting a reference to an engine instance from pyttsx.init()

3. Load a speech engine driver implementation
4. Register and unregister event callbacks by using engine object
5. Get and set current speech engine properties to include rate, voice, voices, and volume.
6. Call function say() to speak an utterance according to the properties set
7. start and stop event loops using runAndWait()

```
Engine1 = pyttsx.init()
Engine1.setProperty('rate', 100)
Voices1 = Engine1.getProperty('voices')
for Voice1 in Voices1:
    Engine1.setProperty('Voice1', Voices1[1].id)
Engine1.say(entrybox2.get())
Engine1.runAndWait()
```

Figure 5. Python code for Pronunciation and Speech Synthesizers

4. Result

The proposed EM Dict is an attractive and user-friendly dictionary. It supports more than 105 languages and pronunciation. Figure 6 shows Synonyms, Antonyms, Verb, Noun and Pronunciation in English and Tamil. It installs on all windows. It installs quickly on a system with less memory. It converts text to speech in

pronunciation. It is very useful to mainly students, Language learners, professors, etc..^{5,6}.

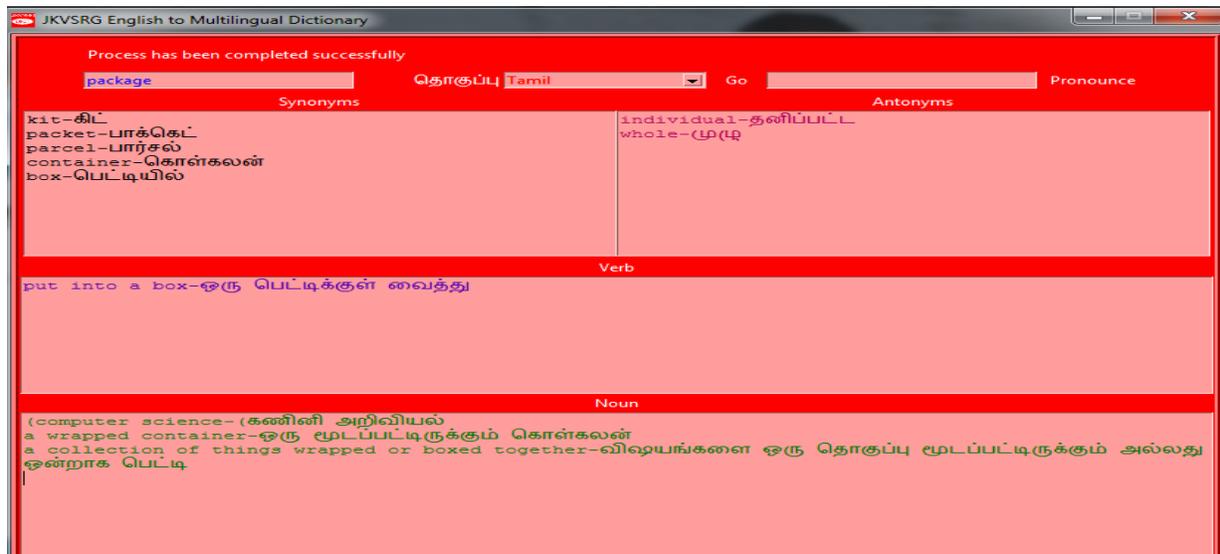


Figure 6. EM Dict showing Synonyms, Antonyms, Verb and Noun in English and Tamil

5. Conclusion

English to Multilingual Dictionary is globally interrelated with the people of other socio-cultural communities. EM Dict is an InnoWordnet dictionary which was developed from Open Source Python

Libraries and Google translation. It offers pronunciation, synonyms, and its related concepts in More than one language. By providing this study, we hope that professional designer, developer, and publishers obtain advantageous to develop and seek

professional translation. In future, we plan to expand this interface by Multilingual File Translator.

6. References

1. <https://pypi.python.org/pypi/PyDictionary/1.3.4>
2. <https://pythonhosted.org/goslate/>
3. <http://pyttsx.readthedocs.io/en/latest/engine.html>
4. https://www.tutorialspoint.com/python/python_gui_programming.htm
5. <https://jkvsrg-english-to-multilingual-dictionary.en.softonic.com/>
6. <https://sourceforge.net/projects/jkvsrg-eng-multilingual-dict/>
7. <https://www.altalang.com/beyond-words/2010/01/19/evaluating-machine-translation-the-present-and-future-of-multilingual-search/>
8. <https://www.quora.com/Why-do-people-speak-different-languages-around-the-world-Why-dont-we-speak-the-same-language>
9. <https://www.daytranslations.com/blog/2015/01/human-translation-important-5885>
10. <https://www.cse.iitb.ac.in/~pb/papers/icon15-indowordnet-dictionary>
11. Hutchins, W.J. 1995. Machine Translation: A Brief History. In Koerner, E.F.K.; Asher, R. E. (Eds.) A concise history of the language sciences: from the Sumerians to the cognitivists, Oxford: Pergamon Press, 431-445.