# TamilInayaVaani - Integrating TVA Open-Source spellchecker with Python

Authors: T. Shrinivasan, Nithya Duraisamy, Ashok Ramachandran, Manickkavasakam, Arunmozhi, and A. Muthiah
Email: tshrinivasan@gmail.com , nithyadurai87@gmail.com, ashokramach@gmail.com, manikk.h@gmail.com, aruntheguy@gmail.com, and ezhillang@gmail.com

## Abstract

We report the integration of Tamil Virtual Academy (TVA) Open-Source publication of TamilInaya Vaani spell checker into the existing Python environment for Tamil NLP applications, uses. Significant modification from the baseline spell checker was carried out including porting to Python, integration with Tamil sandhi checker, python packaging and addition of web-interface. We report limitations of this spell checker and future directions for this project.

## 1. Introduction

There are several open-source spelling checkers for Tamil with varying levels of completeness and in order of wider use we list them as, Hunspell, GNU ASpell, Langtool, Open-Tamil Solthiruthi, TamilSpell Checker with BloomFilter [6-10], and now we introduce TamilinayaVaani by this work.

Part of technical problem of correcting Tamil spelling errors in text is due to discrete infinity of vocabulary generation mechanisms in the morphologically rich agglutinative language, conjugation “sandhi” and free-word order properties of Tamil language. This prevents us from seeing much success while applying cookie-cutter techniques like Levenshtein-distance search on a fixed wordlist either exhaustively, a poor choice, or via selective search of edit distance by the Norvig algorithm.

Applying case rules, and stemming the words and using list of commonly occurring errors have shown much success in fairly proprietary spelling checkers like Vaani, MenThamizh (Amma Tamil), etc. which are current state of the art. However, these spelling checkers still have their own limitations including searching for 1-edit distance errors and limited in application by lack of concordance and word-sense disambiguation features to have robust suggestion algorithm.

Recently, after sustained efforts by our collaborators, the Tamil Virtual Academy had released source code of its commissioned project for Vaani Desktop Spell Checker under terms of GPL (v2) license. This allows open-source work on the code despite the limitations of obfuscated nature of the source, and undocumented nature of the spelling checker at time of release.

Our contributors ported the C# version of TamilnaiyaVanni spellchecker to Python language. This provides tons of new opportunities to mix with their own applications and make new desktop applications, plugins to other systems, web and mobile applications.

We converted the python code to a python module. Added a sandhi checker availble in Open-Tamil python library and released as a new python module - **tamilinayavaani**.
Apart from the python version, a javascript port of the same spellchecker is also in progress.

# 2. Organization of Python Spell Checker

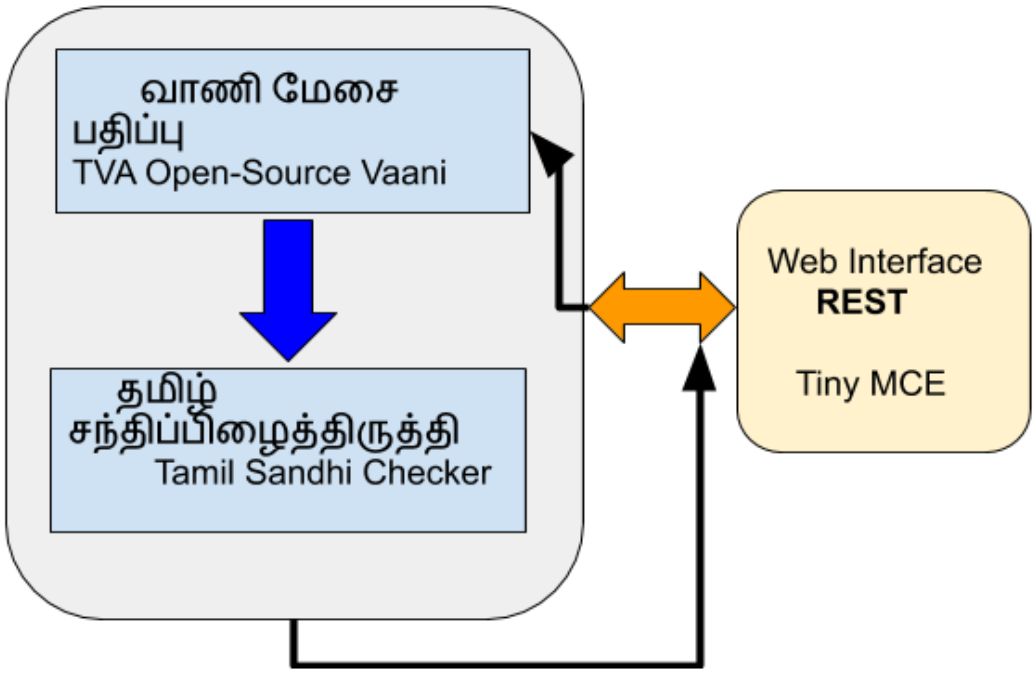


Fig. 1.a Organization of Python module tamilinayavaani; the REST interface portion is outside the module but a static method serves as callback handler for the web server.

# 3. Integration with Sandhi-Checker

Tamil, along with Sanskrit, have concept of sandhi and specific rules on repetition, fusion and exclusion of letters (from a normative form of the word) when two words appear together in context their spelling is altered.

Tamil Sandhi Checker [3] published in 2018, is available as python module “tamilsandhi” distributed along with Open-Tamil package. This contains over 40 rules which enable improved quality of results.

# 4. Python Packaging

This project library can be installed as a Python package **tamilinayavaani** [5] using the commands:

```
$ python3 -m pip install --upgrade tamilinayavaani>=0.13
```

This provides two interfaces, namely for use in-memory and for use on a disk-file as follows,

| Usage Type | Code Sample |
|---|---|
| In-Memory Usage | from tamilinaiyavaani import SpellChecker<br><br>flag,suggs=SpellChecker.REST_interface('வாழை','பழம்')<br><br>expected=['வாழைப்', 'பழம்']<br>assert not flag<br>assert expected[0]==suggs[0] |
| File based Usage | from tamilinaiyavaani import SpellChecker, SpellCheckerResult<br><br>result = SpellChecker(fname).run() #fname is a full filename<br><br># result is a list of SpellCheckerResult objects. |

# 5. Web Interface

Tiny MCE web editor is used as framework to enable interactive editing and spellcheck function from this editor is integrated into the server running tamilinayavaani in backend as a Flask application.

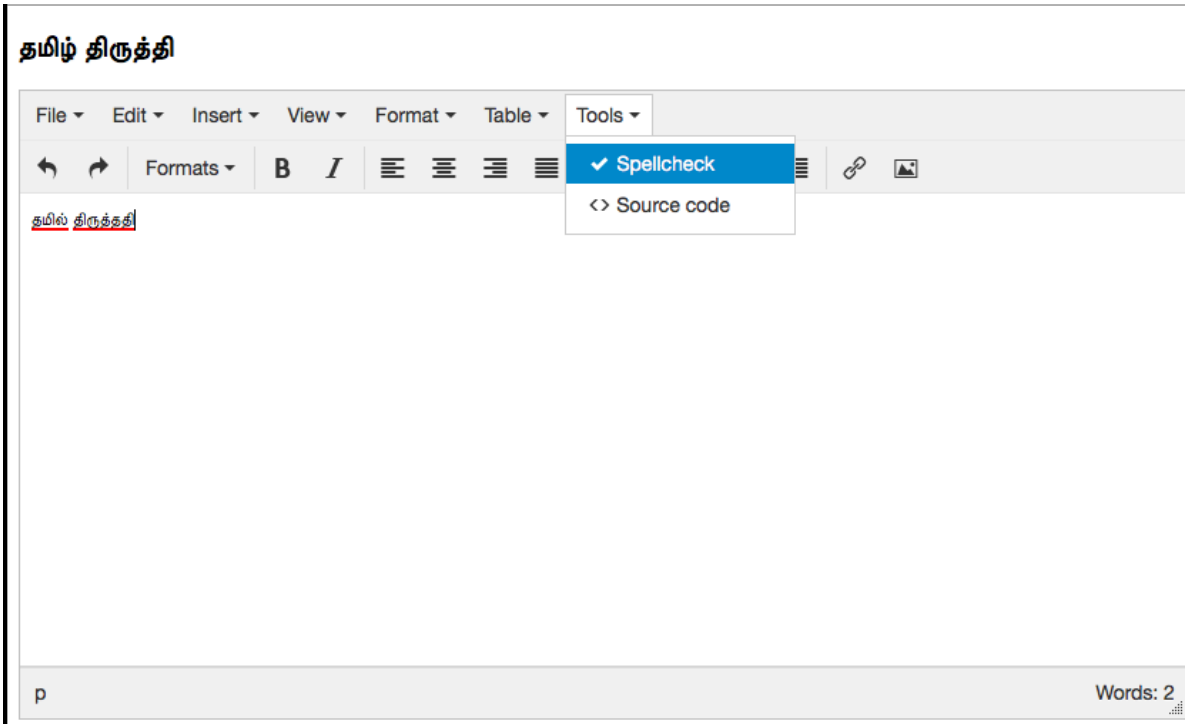


Fig. 2.a. Invoking the spell checker running in the Flask application server in backend.

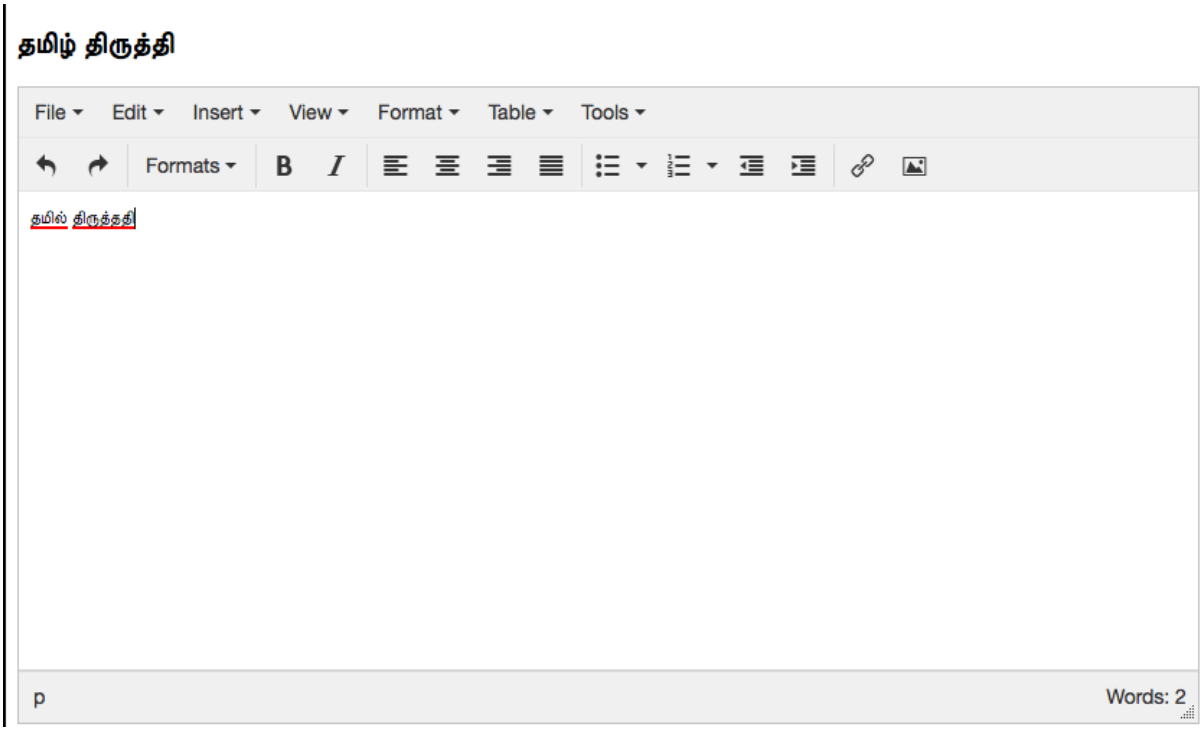


Fig. 2.b. Erroneous words indicated in the text/web-interface.

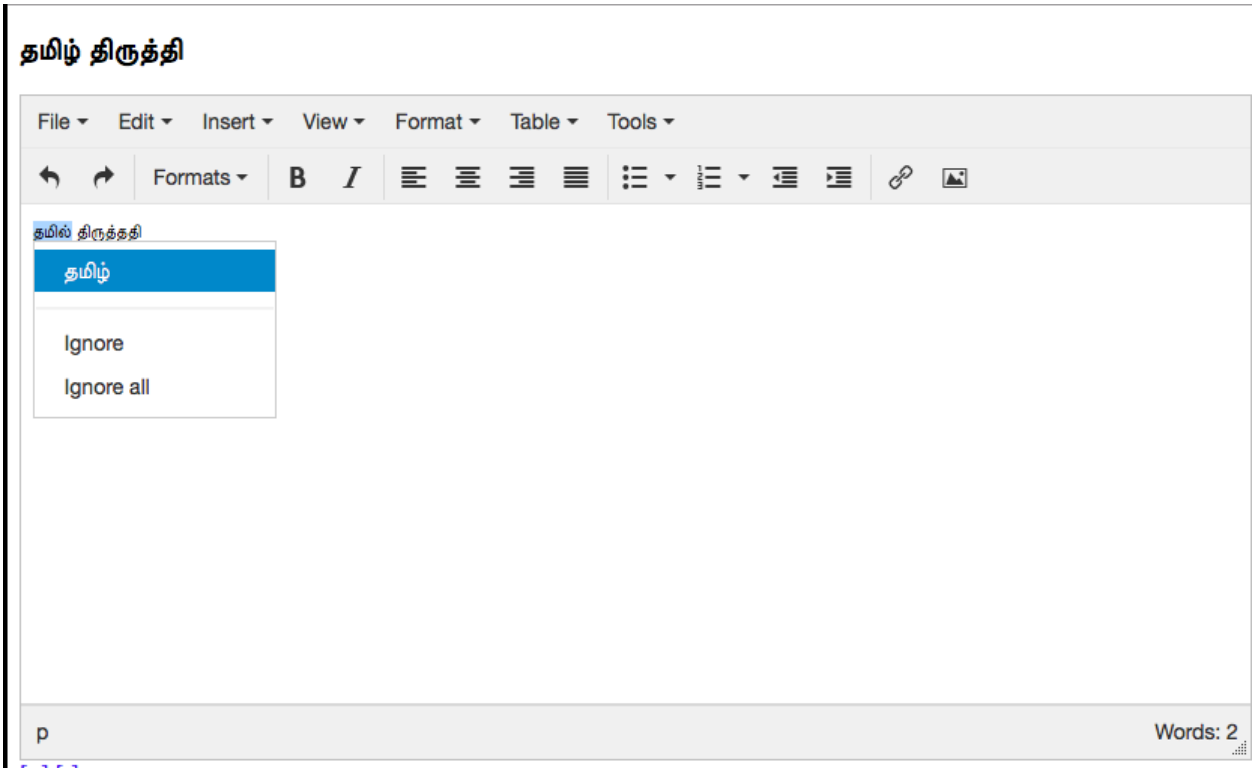


Fig. 1.c. Alternatives for first erroneous word

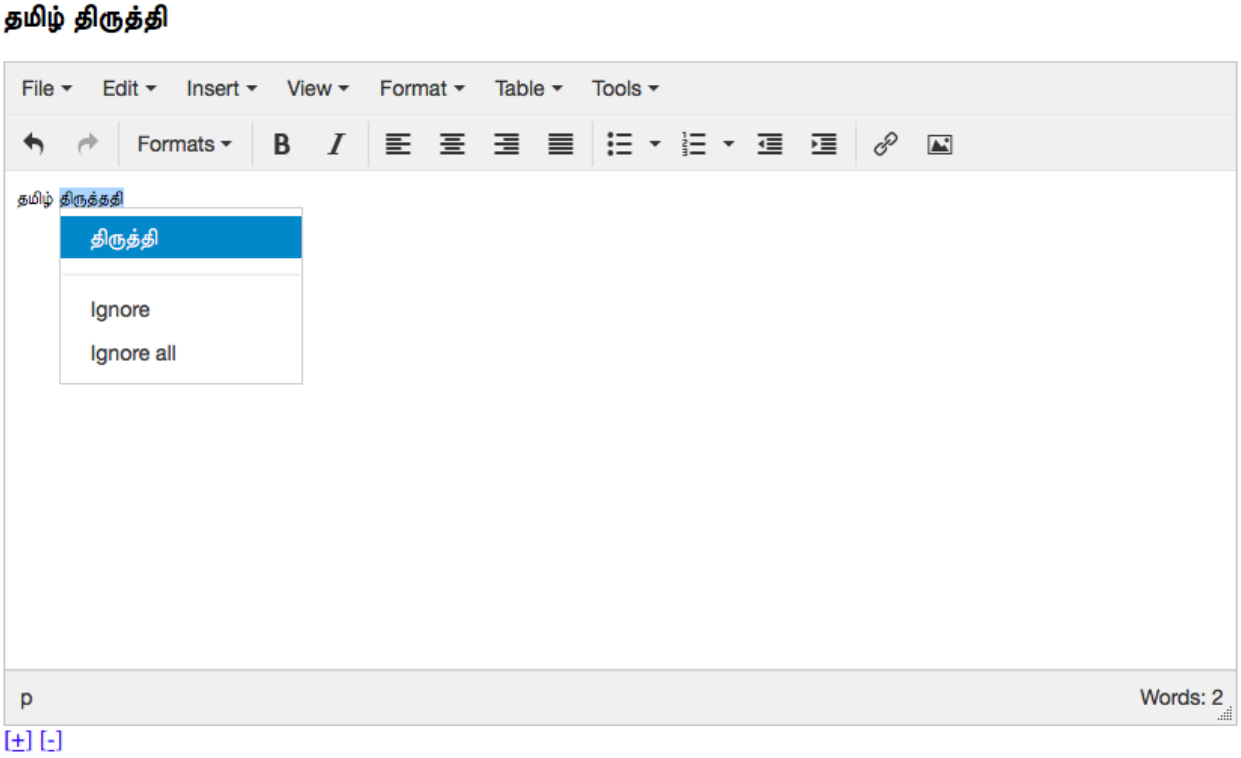


Fig. 2.d. Alternatives for second word after correcting first word.

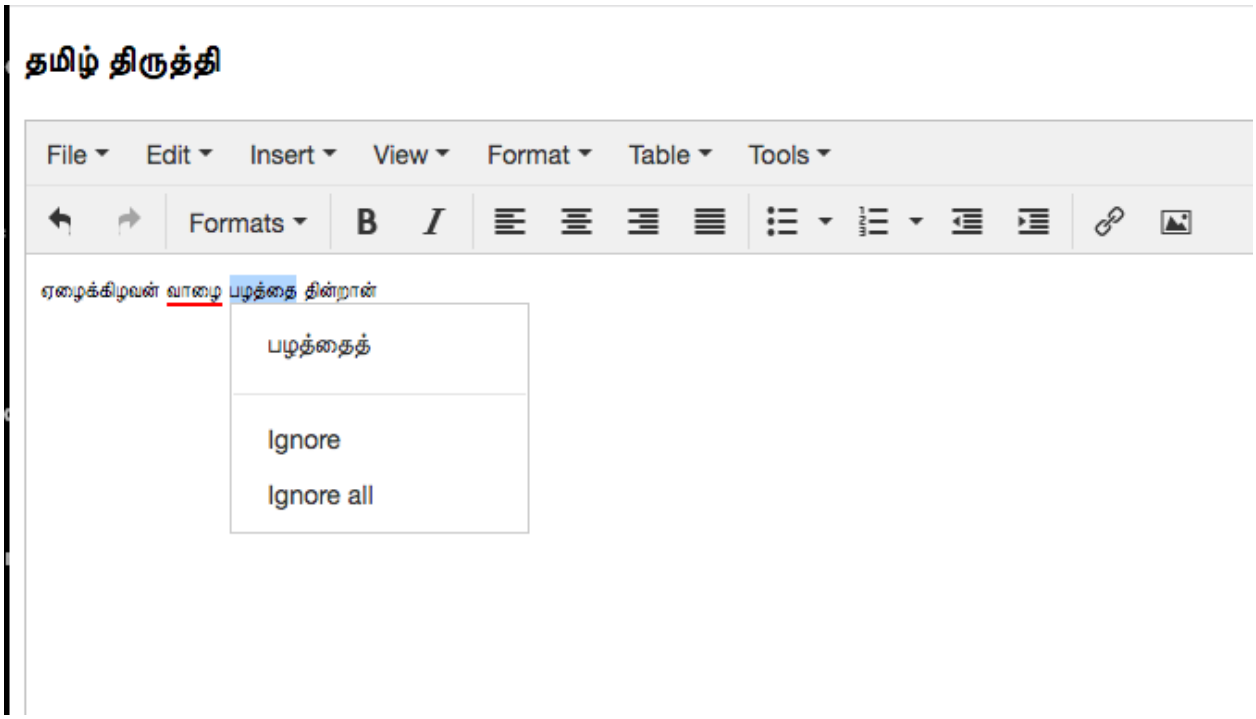


Fig. 3. Sandhi checker in action through the same web-interface.

Further work can involve providing a common API for all Tamil spelling checkers. A good (common) interface would be pyspeller project including the following methods:

1. API to check if its known or unknown (Dictionary membership)
2. Set edit-distance=2 (default)
3. Correction: method to yield best alternate suggestion to word
4. Suggestions: alternates for misspelled word.

# 6. Limitations

Even though the code for Desktop Vaani is made open-source by TVA, the file format and design of the JSON file is kept proprietary making it hard to update the word-list and/or rules associated with the spelling error correction. This is somewhat of a major limitation to update the spell checker compared to the other alternatives like Hunspell, Aspell, etc. mentioned in sec.1 which allow generation of custom affix files for expanding vocabulary of the spell checker. This limitation stands despite the ability of the spell checker to support the custom user word list, as is standard in practice.

# 7. Improving performance with golden words corpus

We have collected 25,83,000 unique tamil words and 1,53,548 unique tamil nouns from various public available tamil text and released as public repositories. We are cleaning these strings to build a golden words of corpus.We can use these words for quick lookup for a given word is there in the corpus or not. If not, we can then apply the invoke the tamilinayavaani spellcheker. This will help to increase the performace on processing large amount of text.

# 8. Conclusion

In this work we report successful creation of an open-source component spelling checker, named *tamilinayavaani*, using the C#(C-sharp) open-source desktop version of Vaani, from TVA. We updated the base version of the software to use sandhi checker from work of Nithya [3]. We integrated this open-source component into a web based user interface via Tiny MCE, and published a Python package for use by everyone under terms of same GPLv2 license.

# References


1. Tamil Virtual Academy publication of Desktop Vaani
2. Vaani - Tamil Spelling Checker by Neechalkaran: http://vaani.neechalkaran.com/ (accessed 2020).
3. நித்யா துரைசாமி, "தமிழுக்கான கட்டற்ற சந்திப் பிழை திருத்தி - உருவாக்கம், பயன்பாடுகள்," தமிழ் இணைய மாநாடு (2018).
4. PySpellchecker package https://pyspellchecker.readthedocs.io/en/latest/, (accessed 2020).
5. Python package for TamilInaya Vaani - https://pypi.org/project/tamilinayavaani/,(accessed 2020).
6. Hunspell, http://hunspell.github.io/ (accessed 2020)
7. Aspell, http://aspell.net/ (accessed 2020)
8. Langtool, https://languagetool.org/ (accessed 2020)
9. M. Annamalai, T. Shrinivasan, **Algorithms for certain classes of Tamil Spelling correction** (https://arxiv.org/abs/1909.10063) (2019)
10. Malaikannan S, et-al, "Improving spellchecker speed with BloomFilter," to appear in INFITT TIC (2020).
11. All Tamil Words - https://github.com/KaniyamFoundation/all_tamil_words
12. All Tamil Nouns - https://github.com/KaniyamFoundation/all_tamil_nouns
13. Series of blog posts on building an open source spellchecker for tamil - https://goinggnu.wordpress.com/category/spellchecker/