AUTOMATIC AMHARIC SPELLING ERROR DETECTION AND CORRECTION USING HYBRID APPROACH

[1]Getnet Assefa
[1]Computer Science Department
[1]Punjabi University, Patiala, India

Abstract: This study is conducted to develop an effective spelling checker system for the Amharic language. Currently there are few natural language processing works that have been done for this language. However, there is no effective spell checker system developed for this language yet. In this study, an effective spell checker mode is developed and then implemented using visual basic programming language. For the prototype development, two basic algorithms were applied in different phases of the study. The Metaphone algorithm is applied for spelling error detection and edit distance algorithm is applied for selecting the most possible correct word for the miss spelled word. For a better usability the developed system is embedded in the Microsoft Office word 2010 word processing system as an add-in. Finally, the developed prototype is tested using test data of 500 words and from these words, 100 of them are miss-spelled deliberately and a dictionary of 125,000 words is used. In the testing phase, the system shows excellent result in its error detection and also in its suggestions.

Index Terms: - *Amharic, Dictionary, Natural Language Processing, Spellchecker*

1. INTRODUCTION

In natural language processing spell checker is one of the main important area which focuses on detecting and correcting spelling errors (Ganfure et al, 2014). They have their own advantage and they could be used independently or embedded within other systems like word processors, optical character recognition, search engines, speech recognition and machine translation, browsers and so on (Hamza et al, 2014).

In different languages, spelling errors are mainly caused by insertion (insertion of the wrong character), deletion (omission of a letter), substitution (using of one letter instead of the other) and transposition (miss positioning of letters).

Spell checker system has two main phases one is error detection phase and the other is error correction (Hamza et al, 2014). The study mainly focuses on developing spelling checker system for Amharic language using hybrid of Metaphone algorithm for error detection and edit distance algorithms for suggestion selection.

1.1. Problem Definition

The Amharic language is a morphology rich language; it has multiple similarly sound letters (symbol redundancy) and also it has phonology clashes. For this reason, it is difficult to develop spell checker system easily. Few works have been done in Amharic spell checker due to the language complexity and also some other reasons those works are not well satisfying. In general, the problems observed in the current existing Amharic spell checker systems are inefficiency in language processing, some are lack of error correction functionality, some doesn't include all Amharic characters in consideration, and also Lack of efficient dictionary

1.2. Objective

The general objective of this study is to develop a well satisfactory Amharic spelling error detection and correction system using a well-organized and effective manner.

1.3. Methodology

The study is done by applying different methodologies in different phases of the study. This section mainly focuses on those methodologies used while conducting the research.

1.3.1. Literature review: To collect a complete knowledge and develop an efficient spell checker system for Amharic language multiple studies in numerous areas like Amharic language property, Amharic morphology, steaming, spell checkers, algorithms and other related documents are studied.

1.3.2. Document collection and analysis: For a better understanding of the existing and related systems, the language property and for better solution finding different documents were collected and reviewed in detail. In the study

the corpus is collected from multiple sources. The collected documents are analyzed by removing different noises in the document. Totally from the collected documents 125,000 of it is used for dictionary preparation and 500 of it used as test data.

1.3.3. Prototype development: While developing the prototypes hybrid of Metaphone algorithm together with edit distance algorithm is applied. In the development edit distance algorithm will be implemented for determining the likelihood of a given word to the correct words in the dictionary and for ranking possible correct words. Finally, the performance of the developed prototype is evaluated by using the selected test data.

1.3.4. Tool: In this research different tools are used in different phases of the study. For the system documentation Microsoft office word 2010 is used. For the corpus preparation, processing and storage Microsoft office Excel 2010 and Microsoft Access 2010 are used and for implementation Microsoft Visual Studio 2010 is used and finally for prototype development, python and VB.NET programming languages are applied in two different steps of the prototype development.

1.4. Scope and limitation

This research is conducted with the aim of developing a satisfactory Amharic spell checker system using a hybrid approach. This paper is only focused on spelling error detection and correction for Amharic language, So Tigrigna and Geez words will not be considered. In addition to this, the system will not consider any real words error, Grammar error, and punctuation errors.

2. LITERATURE REVIEW

This section discusses about previous works related to spell checker.

Yacob (2004), tried to implement Metaphone algorithm for Amharic language. In his work, He encoded letters with similar sound into a single letter to have single representation for words with similar sound. In this is paper, as much as possible the researcher tried to eliminate most errors and in his experiment, he used 116 words and 166 misspelled words. According to his report the Amharic Metaphone algorithm success was up to 90%.

Olani & Midekso (2014) conducted a research in design and implementation of morphology base spell checker for Afaan Oromo language. The language is the most spoken language in Ethiopia and also it is morphological rich language. The researchers designed the spell checker by using dictionary lookup method together with the morphological analyzer. The Microsoft Visual C# programming is used for the development of the spell checker. They used a total of 1811 words of test data, and they tested their system and they reported using the following three matrices, lexical recall 88.62%, Error recall 100% and precession 28.62%.

Hamza et al. (2014) propose an approach for Arabic spell checking which focuses on morphological analysis and edit distance algorithms. For testing, they used a corpus of 2784 misspelled words. They finally tested their system and compare it to Levenshtein distance. They reported the result of the comparison by three error operations such as insertion, deletion, and permutation and found that their approach scores 85%, 81%and 86% success for each error operations respectively and the other Levenshtein distance scores 44%, 81%and 61% success respectively for each corresponding errors operations.

Rasooli et al. (2011) proposes adaptive spell checker approach or Persian language. The approach detected misspelled words using dictionary lookup method. After errors are detected a list of candidate words were generated by using insertion, deletion, substitution and transposition operations. These operations were applied in the wrongly spelled word and the candidate words are ranked based on their similarity to the word. The approach uses the response of the user and the frequency for the future use.

Bassil and Alwani (2012) propose a context-sensitive spelling error detection and correction method based on a dataset containing n-gram word ranging from unigram to 5-gram. The proposed approach had error detector using unigram statistics from Google web 1T dataset and the candidates are generated using character-based 2-gram. Model and the other was context sensitive error correction using a 5-gram statistic from Google web 1T dataset. It considers both real word and nor-real word error. For evaluation, 200,000 words from 300 articles wore used. After that, they reported that the proposed approach corrected 99% non-real word error and 70% of real error.

Shaalan et al. (2012) propose an approach using context-independent spelling correction tool. The paper mainly focuses on knowledge base rules, noisy channel, edit distance and it uses trigram language model which result 98.2% precession and 100%recall. A test set of 400,000words were used, to select correct candidate words for a given misspelled letter they used Levenshtein distance, after that to rank candidates they used two approaches the first approach uses edit distance and noisy channel model and the other uses edit distance, noise model, and other post-processing. According to their report the first approach score 71.3% success and the other approach score 75% success.

3. AMHARIC SPELLING ERROR DETECTION AND CORRECTION MODEL

The proposed system has three major phases those phases are pre-processing, spell checking and spell correction (correction suggestion) phase. The proposed system accepts users input, extract root word and required features by preprocessing user's

input, and check the words validity against the dictionary and if the word found incorrect, the system will provide list of possible correct words closer to the users input by computing the edit distance between user input word and word from dictionary.

3.1. The architecture of spell checker

The architecture below shows the general procedure that the system will go through for validating the correctness of the user input and also for providing the closest possible list of words for the misspelled word as a correction. Preprocessing, spell checking and spell corrections are the three main tasks that the system performs.

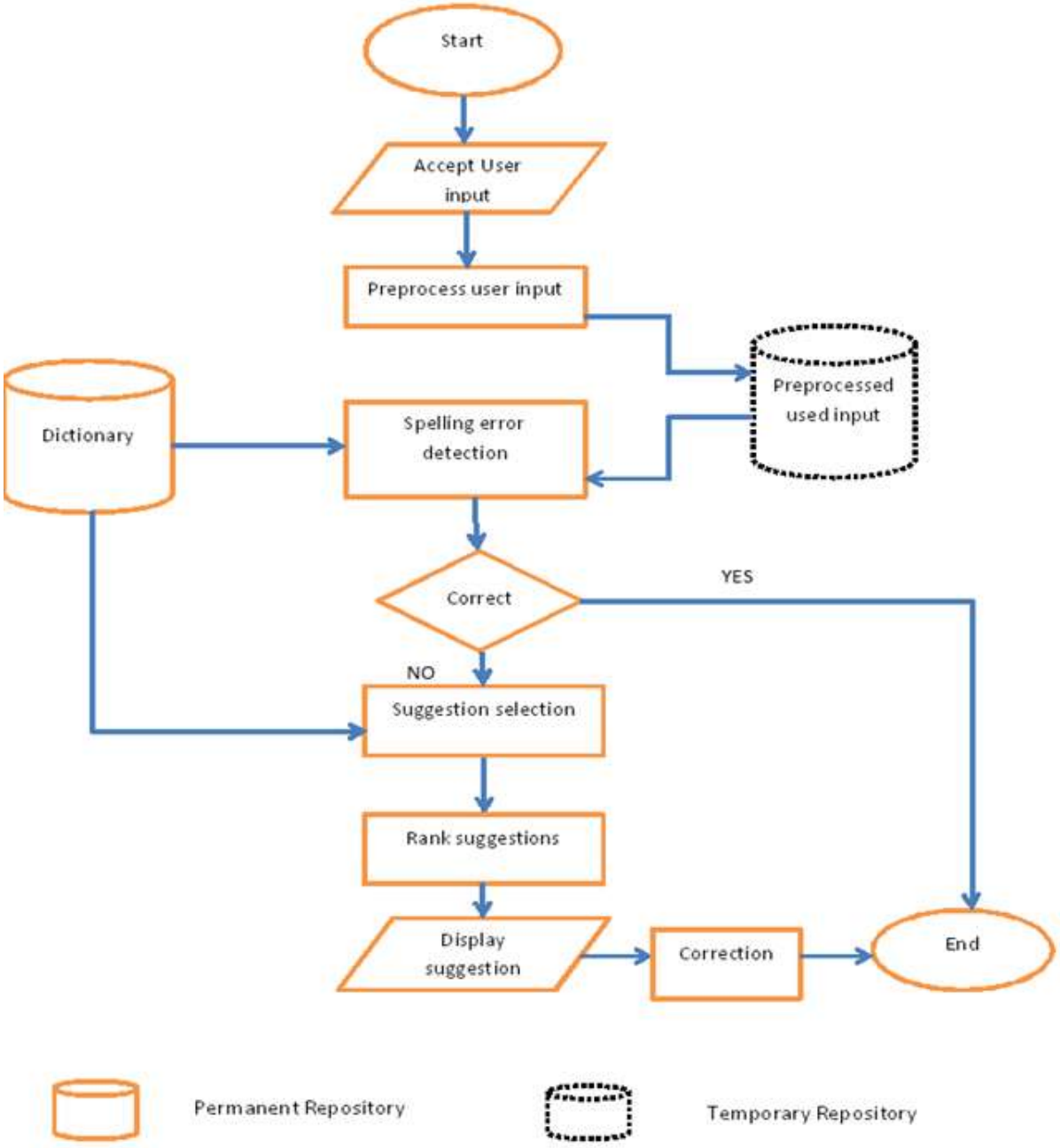

Figure 3.1. The architecture of Spelling checker system model

3.2. Pre-processing

The user input may include numbers or punctuations or words from other languages; therefore, the system eliminates such inputs before spell checking phase began. The output of this phase is used for detecting the word validity.

This phase contains three main tasks that are; punctuation removal, normalization, and steaming. It is applied for automatic key generation.

3.2.1. Punctuation removal

In this section, the system will eliminate punctuations from the users input using the punctuation removal algorithm. In the algorithm a list containing list of possible punctuation marks is used and the input word is checked against this list.

3.2.2. Normalization

One of the causes of spelling errors in Amharic language is the existence of multiple alphabets with similar sound and shape, so in order to detect and correct errors due to such characteristics of the language we are expected to bring those letters in a single common representation. The proposed system will look for this letters in the user input and replace them with their common representation.

3.2.3. Steaming

The main reason why we include steaming in this study is that the Metaphone algorithms that we used for spelling error detection uses steam of a word as a key for similarity measure. The output of this algorithm is used in Metaphone algorithm for error detection as a key of words.

3.3. Spelling error detection

In the prototype development, Metaphone algorithm is used for detecting spelling errors. This algorithm is mainly used for phonetic error correction. It assigns a key to input word to find the more similarly spelled strings with a similar key in the pre-defined dictionary (Khyati, 2012).

In this study, the root of input word is considered as a key and list of possible inflations of that root word are represented under this key.

3.4. Spelling error correction

To correct the miss-spelled word the system will look for the possible list of words in the dictionary which have higher similarity to the wrongly spelled word. The system filters out this possible list of words and sorts them based on their degree of similarity to the miss-spelled word by calculating the edit distance.

3.4.1. Correct word selection

For each erroneously spelled word, the possible lists of correct words are selected by applying Levantine edit distance algorithm. The word with smaller edit distance is labeled as the candidate for the correction. After computing the distance the suggested words are sorted by their distances in ascending order based on its edit distance value.

3.4.2. Candidate ranking

After computing the edit distance for a given wrongly spelled word the suggested words are ranked according to their edit distance; the word with lower distances placed at the top in the ranking position. In the prototype development to sort the list built-in functions are applied.

3.4.3. Spelling corrections

After displaying the list of possible correct words for the detected word the user can choose one of the words from the list shown and then selected word will automatically replace the wrong one in the document. The user also can ignore the word as it is without any correction.

At the system development it is difficult to remember and include all words of the language to the system dictionary; in this reason sometimes the system may detect the correctly spelled word as the wrong one if it is not in the dictionary; in this case, the system allows the user to add the word to the dictionary for future use.

4. IMPLEMENTATION AND EXPERIMENTATION

4.1. Dataset

In this study for testing and dictionary preparation Amharic documents in a different topic and from different sources are collected and from the collected documents more than 125,000 (2.19 MB (2,306,048 bytes) size of) words are used for the dictionary preparation. For the testing phase we used 500 words and from these words, 100 of them were wrongly spelled words.

4.2. Implementation

In this study to develop the complete prototype two steps are followed. In the first step the prototype is developed using python programming and it works for a single word at a time. In this step, the main purpose is to develop the general workflow of the system and come up with independent spell checker system and also to check the model effectiveness.

As specified before spell checkers could be embedded to another system. Under the consideration of this advantage to solve the problem of the developed prototype it is expected to embed the proposed system to other word processing system.

In the second step to develop a fully functional and usable system the system developed in step one is integrated with word processing by using VB programming language. In this step the proposed system is embedded to the Microsoft office word 2010, and to do so VB is used as programing language and Microsoft Access 2010 is used as a database. The integration is done through word add-in.

While opening the Microsoft word 2010 the spell checker add-in and also the dictionary of the spell checker will be loaded. The following figure shows stage when the spell checker add-in is loaded to the word processing system.

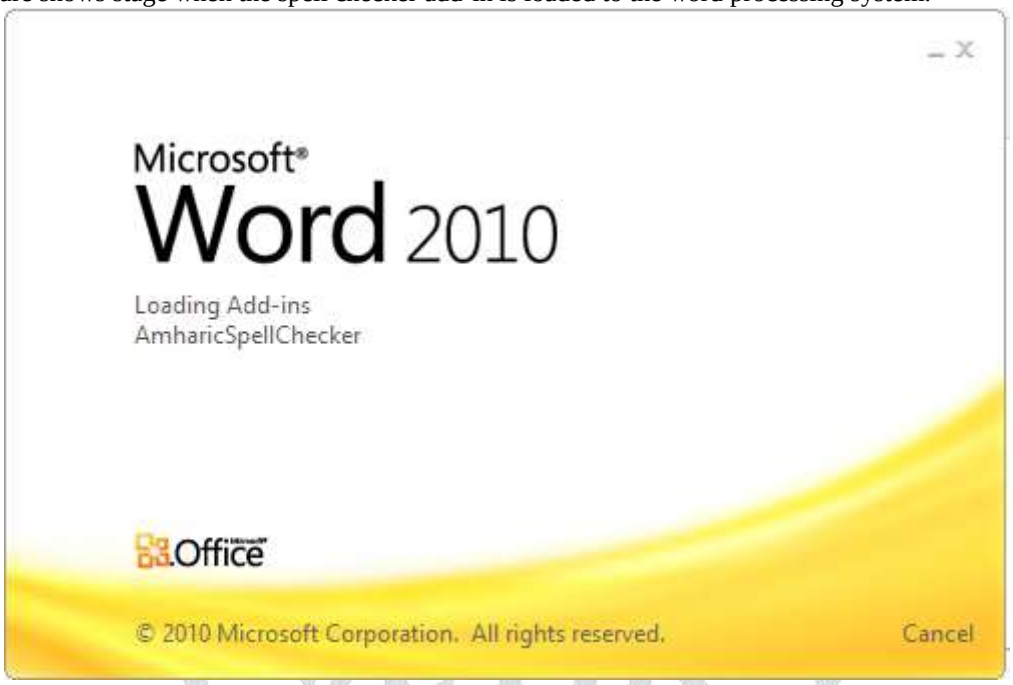

Figure 4.3. Loading spell checker add-in

After typing any Amharic word in the Microsoft word text area to initiate the spell checker the user is expected to choose the "spell checker [ቃል ማረሚያ]" Ribbon from the main menu of Microsoft Office and click on the spell checker button.

After spell checking the possible list of suggestion is shown using list box widget to the left side of the text area. The following diagram shows how the suggested words are displayed for the error detected.

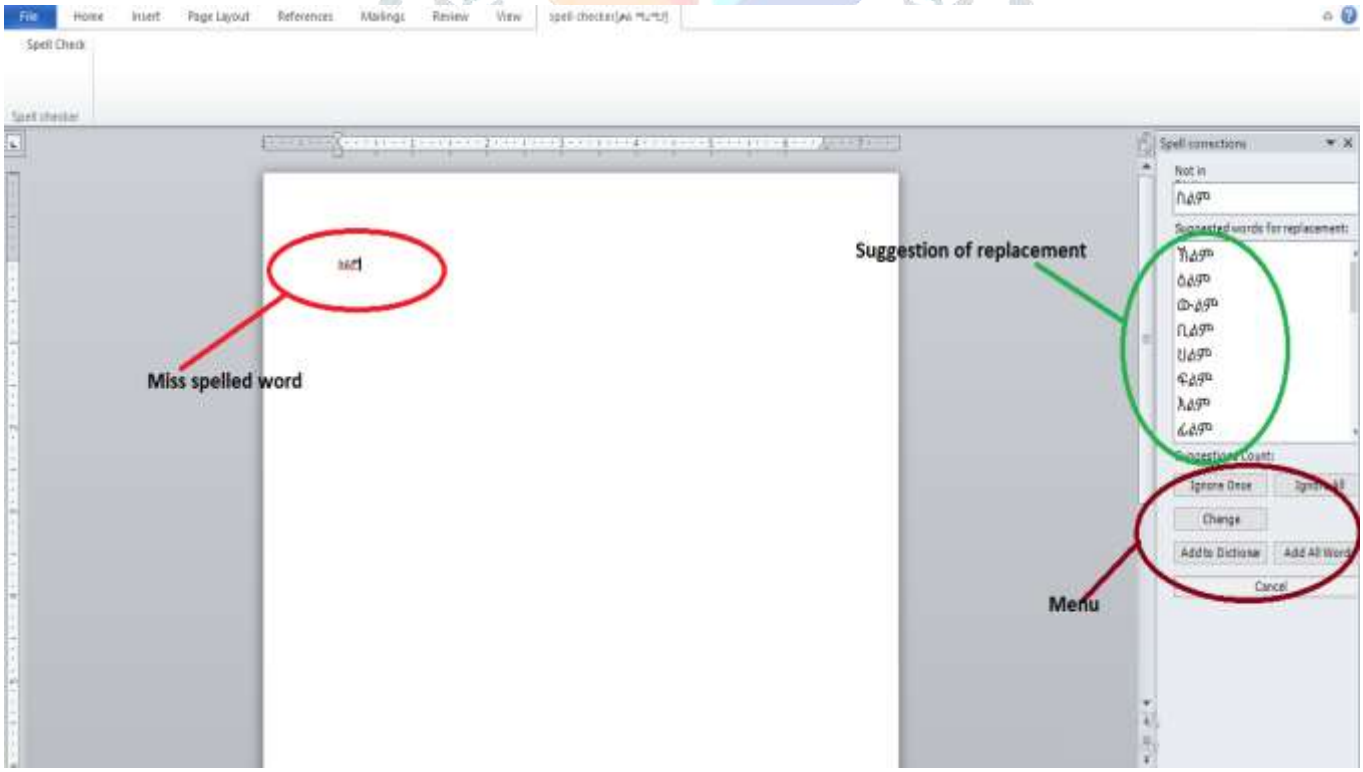

Figure 4.5. Interface of correction suggestion task

To display the suggestions to the user Task Control pane is used and it includes text box which contains the wrong word, list box containing list of suggestions for the wrong word detected, and buttons for different actions taken by the user.

4.3. Testing

In prototype development, two types of testing mechanisms are used one is unit testing and the other is system testing.

After completing prototype development the whole system is tested by using the test data prepared before. The test data contains 500 words and from these words, 100 of them were miss-spelled deliberately. According to this test generally the system scores 98% effectiveness in its error detection and correction functionalities. With respect to its usability the system is easily usable and it is not complicated to use. In addition to this the system is more adaptive to new words that the users can add new words to the dictionary easily for the future use this makes the system effectiveness to be growing time to time.

5. CONCLUSION AND FUTURE WORK

The developed system is developed using the combination of phonetic algorithm and also Edit distance algorithm and it shows more than 95 percent effectiveness in its error detection and also in its correction suggestion capability. However, the speed of processing is not considered carefully in this experiment and in the future expansion of this word processing time will be the main issue.

The developed system has one major limitation which is related to its speed. In this thesis, the model development was the major focus of the study for this reason the high-speed data processing approach is recommended for the future developments.

In addition to this, the system also could be embedded in all other MS office products and also other text processing systems. In general, for the future work, I recommend to implement this system in all Microsoft office products under the consideration of its processing time.

Acknowledgment

First of all, I am very thankful to the almighty God for entitling me to this opportunity and for being with me through my entire journey. Many Thanks to Dr.Dharam Veer Sharma for his supportive comment, supervision, and patience till the accomplishment of this study. Without his regulation and assistance, this work will not be completed. Last but not least, I am very thankful to my families especially my mother, and father for their valuable support throughout this study.

REFERENCE

1. Al-Jefri, M. M., & Mahmoud , S. A. (2013). Context sensitive rabic spell checker using context word and N-gram language model. *Taibah University International Conference on Advances in Information Technology for the Holy Quran and Its Sciences*,:258-263.
2. Agugna, G. (2017, 09 27). *amharic*. Retrieved 01 02, 2018, from Library.bu.edu: http://Library.bu.edu/amharic
3. Attia, M., Pecina, P., Samih, Y., Shaolan, K., & Genabith, J. (2012). Improved spelling Error detection and correction for Arabic. *International Conference on Computational Linguistics COLING* , :103-112.
4. Baljeet, & Harshandeep. (2015). Design and implementation of HINSPEL-Hinde spell checker using hybrid approach. *International Journal of scientific research and management(IJSRM)*, :2058-2060.
5. Barari, L., & QasemiZadeh, B. (2005, Dec 19). CloniZER Spell Checker. *Adaptive, Language Independent Spell Checker*, Pp:19-21.
6. Bassil, Y., & Alwani, M. (May 1, 2012). *Context-sensitive Spelling Correction Using Google Web 1T 5-Gram Information.* Beirut, Lebanon: Canadian Center of Science and Education, :32-40.
7. Bhaire, V., Jadhav, A., Pashte, P., & G., M. M. (April 2015). SPELL CHECKER. *International Journal of Scientific and Research Publications, Volume 5, Issue 4*, :1-3.
8. Bhatti, Z., Ahmad, W., Imdad, A., & Dil Nawaz , H. (2014). Phonetic based SoundEx & ShapeEx algorithm for Sindhi Spell Checker System, :1-9.
9. Ganfure, G. O., & Midekso, D. (2014). Design and implementation of morphology base spell checker. *International journal for scientific & technology research*,:118-125.
10. Hamza, B., Hicham, G., Abdellah, Y., & Mostafa, B. (2014). For an Independent Spell-Checking System from the Arabic Language Vocabulary. *(IJACSA) International Journal of Advanced Computer Science and Applications*,:113-116.
11. Hemma, & sunitha. (2015). Spell Checker for Non-Word Error Detection:. *International Journal of Advanced Research in Computer Science and Software Engineering*,:360-363.
12. indianmirror.com. (n.d.). *sindhi-language*. Retrieved 1 2, 2018, from indianmirror.com: http://www.indianmirror.com/language/sindhi-language.html
13. Iveth. (2010, JANUARY 6). *assimilation.html* . Retrieved 3 3, 2018, from phoneticsandphonology701.blogspot.in: http://phoneticsandphonology701.blogspot.in/2010/01/assimilation.html
14. Jurafsky , D., & Martin, J. H. (2000). *Speech and Language Processing.* United States of America: Prentice-Hall, Inc.
15. Khaled , S., Amin , A., & AbdAllad, G. (n.d.). Autimatic spelling checking for Arabic.:1-8.
16. Khyati , S., Mayuri , P., Jikitsha , S., & Dr. kalpesh , L. (2012). Comparative Study of Spell Checking Algorithms and Tools. *International Journal of Advanced Research in Computer Science Volume 3*, :421-426.
17. Leslau, W. (1995). *Reference Grammar of Amharic.* Wiesbaden, Germany: Harrassowitz Verlag.
18. Mon, A. M., & Thein, T. (2013). Myanmar Spell Checker. *International Journal of Science and Research (IJSR)*, :333-339.

19. *most-spoken-languages-in-africa*. (n.d.). Retrieved 01 03, 2018, from www.africaranking.com: http://www.africaranking.com/most-spoken-languages-in-africa
20. Parmar, V. P., & Kumbharana. (19, July 2014). Study Existing Various Phonetic Algorithms and Designing and Development of a working model for the New Developed Algorithm and Comparison by implementing it with Existing Algorithm(s). *International Journal of Computer Applications (0975 – 8887) Volume 98– No.*, :45-49.
21. Rajashekara Murthy, V. M. (2012). A NON-WORD KANNADA SPELL CHECKER USING. *International Journal of Engineering Sciences & Emerging Technologies*, :43-52.
22. RASOOLI , M. S., KAHEFI, O., & MINA, B. (Nov. 2011). Effect of Adaptive Spell Checking in Persian. *Natural Language Processing andKnowledge Engineering (NLP-KE), 2011 7th International Conference* (pp. Pp:161-164). Tokushima, Japan: IEEE.
23. Shaalan, K., Allam, A., & Gomah, A. (2003). Towards Automatic Spell Checking for Arabic. *Conference on Language Engineering, Cairo Egypt*, :240-247.
24. Shaalan, K., Samih, Y., Attia, M., Pecina, P., & van Genabith, J. (2012). Arabic Word Generation and Modelling for Spell Checking. *Eighth International Conference on Language Resources and Evaluation* , :719-725.
25. Shah, K., Patel, M., Sheth, J., & Lad, D. (2012). Comparative Study of Spell Checking Algorithms and Tools. *International Journal of Advanced Research in Computer Science*, :421-426.
26. Singh, S., Nalawade, S., Gonte, S., Quadri, M. Y., & Godse, S. (2016). Content Improvisation by Spell Checking, Grammar Checking, Tone Checking and Scoring. *International Research Journal of Engineering and Technology (IRJET)*,:1332-1335.
27. Smetanin, N. (2011, MARCH 23). Retrieved 03 03, 2018, from http://ntz-develop.blogspot.in/2011/03/phonetic-algorithms.html
28. Sundby, D. (n.d). Spelling correction using N-grams. Lund Institute of Technology :1-6.
29. Tessema, T. T. (2014, October). *Word Sequence Prediction for Amharic Language.* Retrieved 01 10, 2018, from www.aau.edu.et: http://www.aau.edu.et/library/resources/aau-institutional-repositoryelectronic-thesis-and-dissertation
30. Wasala, A., Weerasinghe, R., & Pushpanan, R. (2010). A Data-Driven Approach to Checking and Correcting Spelling Errors in Sinhala. *The International Journal on Advances in ICT for Emerging Regions*, :11-24.
31. Yacob, D. (2004). Application of the Double Metaphone Algorithm to Amharic Orthography. *International Conference of Ethiopian Studies XV*, :921-932.
32. Zhang, Y., He, P., Xiang, W., & Li, M. (1 June 2016). Discriminative Reranking for Spelling Correction. *Digital Scholarship in the Humanities, Volume 31, Issue 2*, :411–427.