Sentence Level N-Gram Context Feature in Real-Word Spelling Error Detection and Correction: Unsupervised Corpus Based Approach

Tsegay Mullu Kassa
Department of Information and Technology, Wachemo University, Hossana, Ethiopia

Kidst Ergetie Andargie
Department of Information and Technology, Wachemo University, Hossana, Ethiopia

Abstract

Spell checking is the process of finding misspelled words and possibly correcting them. Most of the modern commercial spell checkers use a straightforward approach to finding misspellings, which considered a word is erroneous when it is not found in the dictionary. However, this approach is not able to check the correctness of words in their context and this is called real-word spelling error. To solve this issue, in the state-of-the-art researchers use context feature at fixed size n-gram (i.e. tri-gram) and this reduces the effectiveness of model due to limited feature. In this paper, we address the problem of this issue by adopting sentence level n-gram feature for real-word spelling error detection and correction. In this technique, all possible word n-grams are used to learn proposed model about properties of target language and this enhance its effectiveness. In this investigation, the only corpus required to training proposed model is unsupervised corpus (or raw text) and this enables the model flexible to be adoptable for any natural languages. But, for demonstration purpose we adopt under-resourced languages such as Amharic, Afaan Oromo and Tigrigna. The model has been evaluated in terms of Recall, Precision, F-measure and a comparison with literature was made (i.e. fixed n-gram context feature) to assess if the technique used performs as good. The experimental result indicates proposed model with sentence level n-gram context feature achieves a better result: for real-word error detection and correction achieves an average F-measure of 90.03%, 85.95%, and 84.24% for Amharic, Afaan Oromo and Tigrigna respectively.

Keywords: Sentence level n-gram, real-word spelling error, spell checker**,** unsupervised corpus based spell checker
DOI: 10.7176/JIEA/10-4-02
Publication date: September 30th 2020

1. Introduction

Poor spelling is a common challenge faced by people on their day to day lives, to encounter such issues spellcheckers are an essential tool. Spell checking is a sub-field of computational linguistics that aims to detect and sometimes correct words in a text that are misspelled (Ananjot et al 2015). In principle, spell checking process composed of three basic units: the error detector which flags misspelled words by validating them against a lexicon of words; the candidate correction generator which provides alternative corrections for the detected spelling errors; and the error corrector that suggests the best candidate as a replacement for the misspelled word. In a straightforward approach a spell checker is built in dictionary of words to detect errors, and on a corpus based probabilistic model to perform error corrections. In this approach , when a word is not in the dictionary , so it is considered as misspelled word and such type of error is called non-word form spelling error (Pirinen et al 2014). To correct such detected spelling errors, this approach searches words in the lexicon that resemble the erroneous word. However, this approach is not able to check the correctness of words in their context and such error is called real-word spelling error, words that are found in the language lexicon but contextually not correct.

As result, detection and correction of real-word spelling errors are research issues unit now and those error types are a common type of errors made by people with different natural languages (Neha and Pratistha 2012). In the state-of-the-art few works have been adopt n-gram language model to mitigate this issue: The research work (Verberne and Suzan 2002) proposed a tri-gram-based method for real-word error detection and correction, using the British National Corpus. In this study, when a word tri-gram is not found in the corpus then it has an error, otherwise it is considered correct without using the probability information of the tri-gram. To solve the data sparseness problem, the research work (Fossati et al 2007) proposed a method of mixed tri-grams model that combines the word tri-grams model and POS tri-gram model. One more study (Islam et al 2009) also presented a method for detection and correcting multiple real-word spelling errors by adopting a normalized and modified version of the string matching algorithm, Longest Common Subsequence (LCS), and a normalized frequency value. This study applied using Google web 1T 3-gram dataset via word tri-grams language model in detecting and correcting real-word errors. In all study cases, a fixed n-gram context feature is applied to check correctness of words in their context and this ignores the normal scenario of natural languages. In natural languages, the context features are applied at sentence level and this enhances the effectiveness of real-word spelling detection and correction. Since, at sentence level rich contextual features can be extracted to learn the model.

As result, this research finding out whether context-aware spell checking based on word n-gram language model with sentence level n-gram context feature is an effective solution to the problem of real-word spelling errors. For this purpose, we have formulated the following research questions:

i. What proportion of real-word spelling errors can be detected and corrected using fixed n-gram context feature?

ii. What proportion of real-word spelling errors can be detected and corrected using sentence level n-gram context feature?

Despite the data focusing exclusively on under-resourced languages such as Amharic, Afaan Oromo and Tigrigna languages, this study can also be adoptable for other languages as far as unsupervised corpus (i.e. raw text) is available.

2. Data and methodology

2.1. Dataset selection

In order to evaluate the effectiveness of the proposed context sensitive spell checker with deferent context features extraction technique testing data set is required. As explained earlier, the proposed model requires only unsupervised training dataset. As result to training and test the proposed model we collect raw text corpora from HaBit project (Harvesting big text data for under-resourced languages) (HaBit project 2014), which is developed to gather large scale text data (corpora) from web for under-resourced languages. The detail statistical information of corpora (i.e. both training and test) described under below Table 1.

Amharic # sentence	Afaan Oromo # sentence	Tigrigna # sentence
320,000	208,900	139,300

Table 1: Total corpus size statistics

As result, the total textual corpus in domain language split into training and test dataset using 10-fold cross validation. The total corpus is split into 10 manually exclusive subsets of approximately equal size for each language and ten iterations were used to conduct the experiments. For each iteration, we isolated one part of the dataset for testing while retaining nine parts as the training set. Beside this, the erroneous test sets are created by exchanging the location of words in random manner, since there is a problem of actual and well organized test set per domain language for evaluation of the proposed model.

2.2. Proposed model

The main purpose of this paper is to observe effectiveness of using all possible n-gram context features at sentence level during real-word spelling error detection and correction. To do so, as shown in Figure 1 the input of the model is any textual unit (i.e. phrase, sentence or etc) and the language in which text is written should be selected by user. The input text unit is first segment into set of sentence with index information, the index information is useful to control the context analysis of text per sentence. At time the last text unit is taken for n-gram extraction and all possible n-gram features at sentence level are extracted and used. Once, all possible n-grams are generated, the validity of each n-grams form higher size to lower size n-gram (i.e. bi-gram) are checked along the target n-gram language model. When all possible n-gram are not found in the target language model then the last word of text unit is considered as misspelled word. To find correction, all resemble words of each mistaken word are generated form lexicon model and each of resemble words are replaced in place of misspelled word to check its correctness to the given context. These new formulated sequences of word n-grams having higher relative probability in the n-gram language model are presented as candidate suggestions.

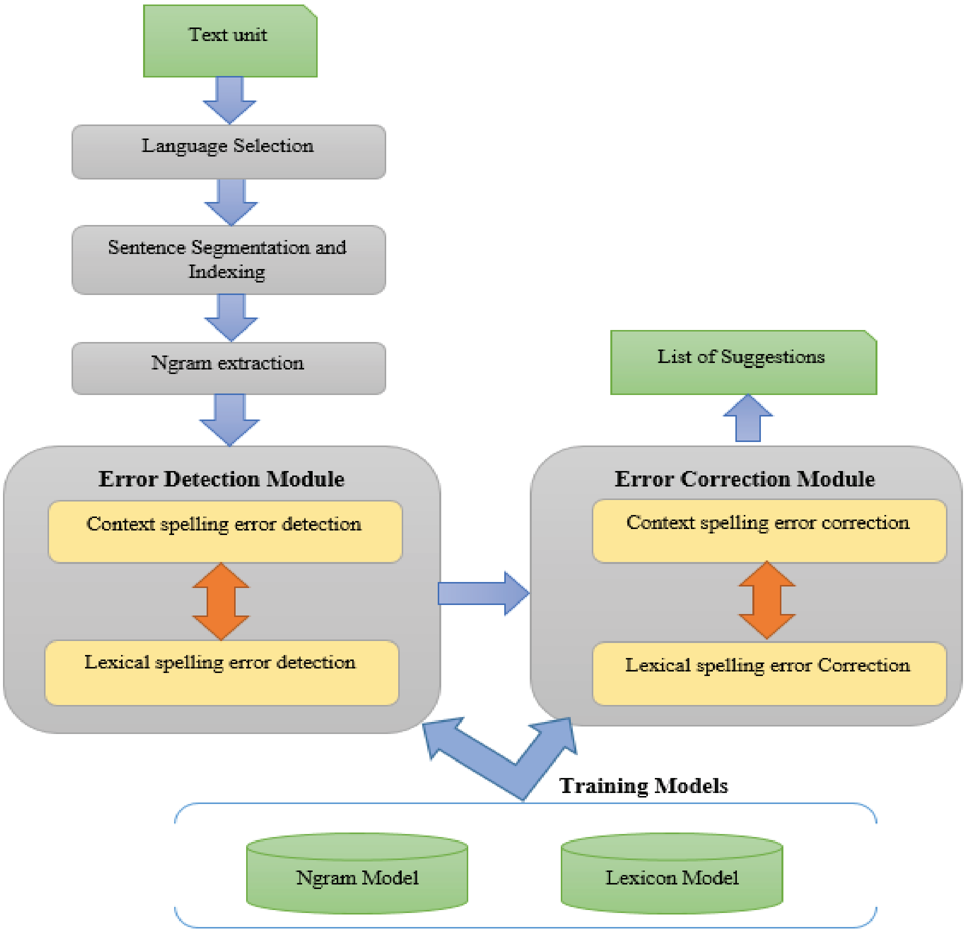
Text unit
Language Selection
Sentence Segmentation and Indexing
Ngram extraction
List of Suggestions
Error Detection Module
Context spelling error detection
Lexical spelling error detection
Error Correction Module
Context spelling error correction
Lexical spelling error Correction
Training Models
Ngram Model
Lexicon Model

Figure 1:- Generic model of language independent context-aware spell checker using sentence level n-gram features

2.3. Language Selection

This module is designed to recognize the language of the input text unit. It is back bone of the proposed model, since the modules coming after in the pipeline need to know the language of the input text to perform real-word spelling error detection and correction without language restriction.

There are different types of approaches for language identification of textual document including the character n-gram, words with dictionaries of various languages and use language stop words as backlist (Truica et al 2015). Due to nature of spell checker, which detects and corrects misspellings at phrase or sentence level it is not effective to adopt the above techniques of language identification. Hence, in this investigation the language is specified as a parameter and chosen by the user.

2.4. Sentence Segmentation and Indexing

This module is responsible to split a string of input text into meaningful units called sentences. As shown in Figure 1 this module is fundamental, since identifying and indexing sentence for the given input text is relevant for our context analysis during real word spelling error detection and correction. Beside this, indexing feature is used to enhance the performance (i.e. time and space complexity) during matching of test n-grams along with language n-gram model.

In most past decade research works, the context features that are extracted for detection of real-word spelling errors are at fixed n-gram level and this makes the model poor on detection and correction of real-word spelling errors. Since the number of context features extracted to learn model is not enough. As result, to solve this issue in our proposed model the context features are extracted at sentence level. To achieve the process of splitting up a running text into sentences, the sentence boundary markers are loaded dynamically once the text language is identified. Beside this, a test text unit may not be a sentence (i.e. phrase) and such units are considered as a single sentence to make suitable our sentence indexing operation.

2.5. N-gram Extraction

This module is responsible to extract the n-gram context features of previously indexed sentience and this extraction may vary depending on the types of n-gram context feature extraction technique (i.e. fixed n-gram or sentence level n-gram). During fixed n-gram the number of extracted features are fixed to some specified n-gram size (i.e. N = 2 or 3 or 4 etc) and this reduce the number of context features to learn the proposed model. However, during sentence level n-gram all possible context features higher than two word sequence (i.e. word bi-gram) are used as learn proposed model and this enhance the effectiveness of model on detection and correction of real-word spelling errors. The next example explains how all possible context features are extracted from a given text unit.

To show each of the context feature extraction techniques with Amharic text unit example

"ልጁ እራቱን በልቶ እንቅልፉን ተኛች / lju 'ratun belto 'nqlfun teNac "

All possible word n-grams using fixed n-gram extraction technique (i.e. in this study we adopt N = 3)

All possible tri-grams

ልጁ እራቱን በልቶ

እራቱን በልቶ እንቅልፉን

በልቶ እንቅልፉን ተኛች

All possible bi-grams

ልጁ እራቱን

እራቱን በልቶ

በልቶ እንቅልፉን

እንቅልፉን ተኛች

All possible word n-grams using sentence level n-gram feature extraction technique (i.e. number of possible ngrams depends on the number of words in given text unit) from above given Amharic text unit is as follows:

All possible penta-gram

ልጁ እራቱን በልቶ እንቅልፉን ተኛች

All possible quad-gram

ልጁ እራቱን በልቶ እንቅልፉን

እራቱን በልቶ እንቅልፉን ተኛች

Beside the above possible n-grams, the above tri-grams and bigrams also possible contextual features extracted using this feature extraction technique. As shown from above example, during sentence level all possible context features are extracted and all these features are used in building the language model. This enables the proposed model more effective on detection and correction of real-word spelling errors. Once, all context n-gram features are extracted from unsupervised training dataset and these used as input for spelling error detection and correction module.

2.6. Error Detection Module

This module is design to detect real-word spelling errors using n-grams language models (i.e. fixed n-gram and sentence level n-gram). Since, to provide a candidate correction for given text unit suspicious words in given text unit should be first and when misspelled words are not detected correctly it is difficult to provide candidate corrections. This module finds suspicious words in a given context by checking the availability of the all possible extracted word n-grams in a text unit.

Once all possible fixed n-gram features from above example are extracted, this module only scans non-checked tri-grams (i.e. all tri-gram and bi-gram that contains last word of text unit). Since, spell checker is a real-word system that checks spelling error of every word when user press word boundary character (i.e. backspace). As shown in the above section, the bold n-grams extracted from Amharic sentence are non-checked n-grams and used for error detection module.

This module checks the spelling correctness of given text unit by lookup the occurrence of non-checked n-grams in target language n-gram model. The lookup is start from higher word n-gram and if not found the checking is continue until bi-gram. In all cases, when none of the possible extracted n-gram are not found in target language n-gram model and the highest relative probability obtained by the most probable spelling variation in target context is compared with the probability of the original word context then the last word (i.e. ተኛች) is considered as suspicious.

2.7. Error Correction Module

Once a real-word spelling error has been detected via previous error detection module, then the proposed model design module that is responsible to generate a list of possible suggestions and this module is called error correction module. To do so, in this module we incorporate four sub-modules: generate candidate words, generate candidate texts, ranking candidate texts, and correct error words.

2.7.1. Generate candidate words

After a real-word spelling error has been detected, candidate correction words are generated for misspelled word in order to suggest correction. To do so, different algorithms have been used for finding candidate corrections in the literature. The minimum edit distance is by far the most popular one and it is deals about the minimum number of editing operations (i.e. insertions, deletions, substitutions and transposition) required to transform one string into another. (Demerau 1964) Implemented the first minimum edit distance based spelling correction algorithm based on the first three types of character transformation, (Levenshtein 1966) developed a similar algorithm for correcting deletions, insertions and transpositions. (Wagner and Fischer 1974) Generalized the technique to cover also multi-error misspellings.

In this proposed work we adopt minimum edit distance technique to generate the resemble words from lexicon to the target misspelled word. Hence, the approach looks the spelling variation of that misspelled word which fit better into the context than the original word. All words that have a minimum edit distance of one from the erroneous word are fetched from the lexicon and considered as candidate correction for the erroneous word.

From the above Amharic text unit the word “ተኛች” is misspelled for given context and it’s resemble words extracted from the lexicon dictionary with minimum edit distance are shown in Table 2.

ተኛ	ሲተኛ
ተኙ	ተኛሽ
ተኚ	ተኛችሁ
ተነሳ	ትተኛላችሁ

Table 2: Word variations for misspelling word “ተኛች”

2.7.2. Generate candidate text

The generation of candidate words does not consider the context feature of given text unit. Since, minimum edit distance during candidate word generation only finds the resemble words for target misspelled word without context consideration. However, to correct real-word spelling errors all candidate resemble words of misspelled word are checked along the given text unit context. To do so, after generating candidate words, new text units are formed by replacing each erroneous word by all of its variation. However, the candidate words are generated only for these non-checked n-grams by replacing the resemble word in place of misspelled word.

For more clarification let’s see the candidate texts that are generated with resemble word variations for above Amharic text unit. All possible candidate text via fixed n-gram contextual features (i.e. only non-checked n-grams are taken for candidate text generation)

All possible candidate text for tri-grams “በልቶ እንቅልፋን ተኛች”

በልቶ በልቶ እንቅልፋን ተኛ	በልቶ እንቅልፋን ሲተኛ
በልቶ በልቶ እንቅልፋን ተኙ	በልቶ በልቶ እንቅልፋን ተኛሽ
በልቶ በልቶ እንቅልፋን ተኚ	በልቶ እንቅልፋን ተኛችሁ
በልቶ እንቅልፋን ሊተኛ	በልቶ እንቅልፋን ትተኛላችሁ

Table 3: Possible candidate texts generated for non-checked tri-grams

Similarly for all non-checked n-grams extracted with fixed n-gram or sentence level n-gram context feature extraction techniques, all possible candidate text units are extracted using this module.

2.7.3. Ranking candidate corrections

Once all possible candidate text units are identified then the list of candidate suggestion are ranked based on their relative probability value computed from target n-gram language model. The relative probability of new text units that are selected as suggestion for the original text unit depends on the types of ngram extracted form training language model. The relative probability of every previously generated candidate texts are computed as follows:-

$$relativeProbngram = \frac{occurance\ of\ ngram}{\sum \quad all\ target\ ngrams\ occurrence} \qquad equation\ (1)$$

Where

$relativeProbngram$ is the probability degree of given n-gram along the given target n-gram language domain

$occurance\ of\ ngram$ is the ngram frequency in a given target n-gram language domain

$\sum \quad all\ target\ ngrams\ occurrence$ is the sum of frequency of all target n-gram in given language domain

The list of candidate suggestions are ranked such that the most probable considered as best fit to the given context. In order to differentiate candidate suggestion based on their relevance to the given context, in this research we use the above equation 1. Hence, the user is provided with the top n suggestions for choosing the most suitable one. If the top two choices have equal probabilities then ranking could be based on a similarity measure, like minimum edit distance between the suggestions and the erroneous word.

Candidate text	Relative probability
በልቶ በልቶ እንቅልፉን ተኛ	59.75
በልቶ እንቅልፉን ሊተኛ	42.03
በልቶ እንቅልፉን ሲተኛ	35.03
በልቶ በልቶ እንቅልፉን ተኙ	0.0
በልቶ በልቶ እንቅልፉን ተኚ	0.0
በልቶ በልቶ እንቅልፉን ተኛሽ	0.0
በልቶ እንቅልፉን ተኛችሁ	0.0
በልቶ እንቅልፉን ትተኛላችሁ	0.0

Table 4:- Ranked candidate text for original misspelled text

Minimum edit distance and word n-gram frequency could be combined together. In case of equal minimum edit distance, the most frequent will be considered highest or they could be interpolated to rank the candidates. The ranked candidate texts for all possible tri-grams generated previously are shown in the above Table 4.

2.7.4. Correct Error Words

In the case of fully automatic system, the detected error words are replaced with the words given in the sentence with the highest probability. However, in the case of interactive system the top *n* candidate words for each suspicious word are suggested to the user to choose the best correction from.

The correction module for either fixed n-gram or sentence level n-gram is similar and in case of fully automatic system the variation that gives the highest probability in the context is compared with the original suspect word. When the variation probability in the context is higher than the original probability and all other variation in given domain language then the real-word misspelled words are replaced with given text unit having highest probability.

However, in case of this paper the grant to choose the right candidate text unit is given for end user. Since, system like spell checker are interactive system and the approach provides the ranked candidate suggestion for misspelled context. This resolve the problem of false positives, the detected errors are flagged as suspicious words with their possible corrections. Then it is the user's job to recognize whether the original word or one of its candidate suggestions is what was intended.

Once the relative probability of new n-grams formulated with new word variations and those n-grams having higher relative probability from the original text unit are taken as suggestion. As result, according to the above Amharic text unit, only two candidate text (i.e. በልቶ እንቅልፉን ተኛ, በልቶ እንቅልፉን ሊተኛ, and በልቶ እንቅልፉን ሲተኛ) are given as suggestion.

For more clarification, the general process of both real-word spelling error detection and correction is summarized in Figure 2 below.

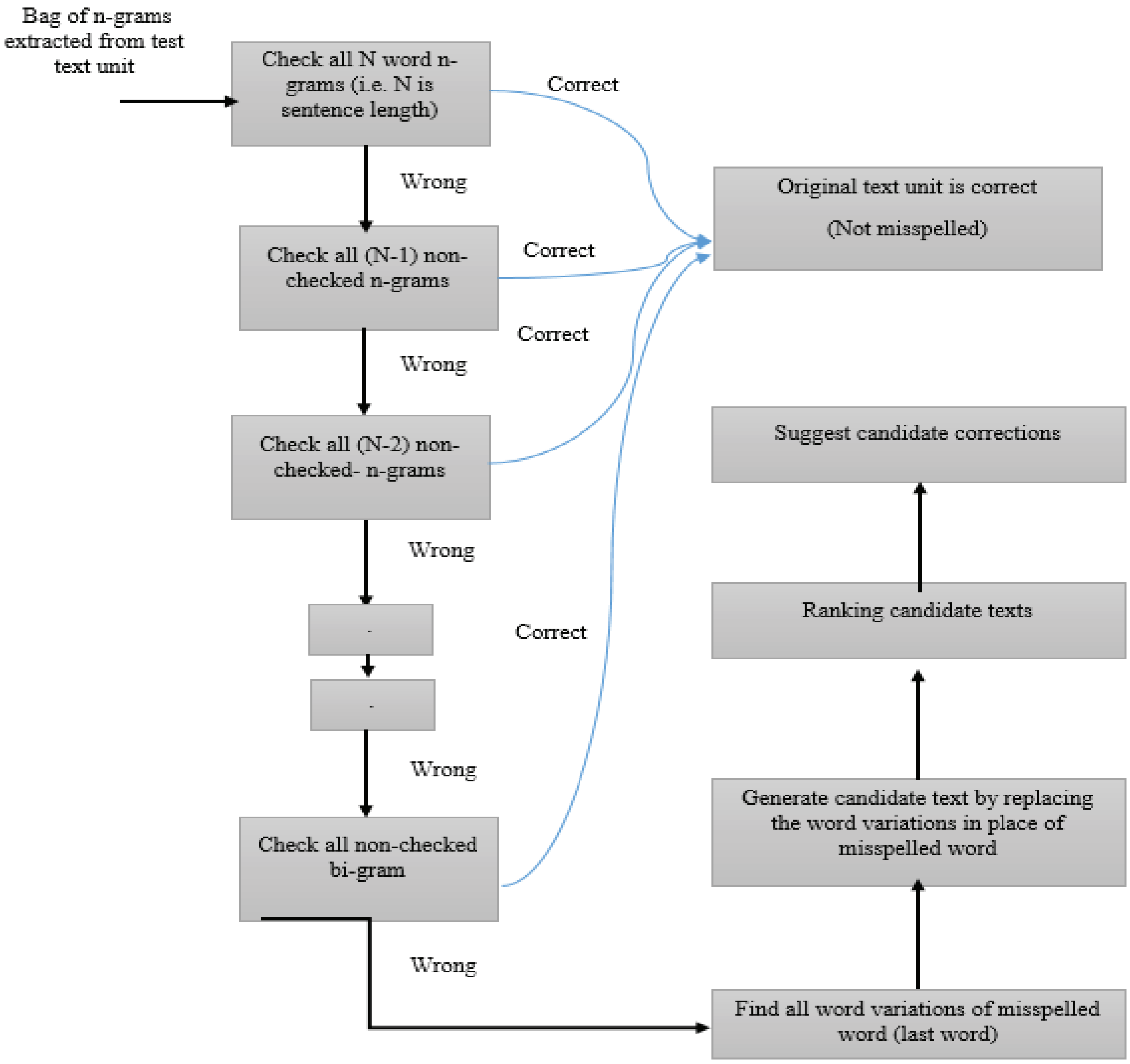
Bag of n-grams extracted from test text unit
Check all N word n-grams (i.e. N is sentence length)
Correct
Wrong
Original text unit is correct (Not misspelled)
Check all (N-1) non-checked n-grams
Correct
Correct
Wrong
Check all (N-2) non-checked- n-grams
Suggest candidate corrections
Wrong
Ranking candidate texts
Correct
Wrong
Generate candidate text by replacing the word variations in place of misspelled word
Check all non-checked bi-gram
Wrong
Find all word variations of misspelled word (last word)

Figure 2: Proposed model with sentence level n-gram feature extraction real-word spelling error detection and correction in detail

3. Results and interpretations

This section describes detail experimental results of both detection and correction effectiveness of proposed context-aware spell checker along with different context feature extraction technique (i.e. fixed or sentence level n-gram). To do so, we adopt recall, precision, and F-measure effectiveness measurement metrics.

The experimentation of this proposed model performed with four experimental techniques: Experiment 1 and 2 are detection effectiveness of proposed model via fixed n-gram context feature extraction technique. Experiment 3 and 4 are design to evaluate the effectiveness of proposed model via sentence level n-gram feature extraction technique. The detail experimental results with the above experimental techniques in terms of both detection and correction effectiveness are stated under the following tables. As shown in Table 5 below, the detection capability of proposed model along with fixed n-gram and sentence level n-gram feature extraction technique are evaluated. To evaluate the effectiveness of proposed model we adopt recall, precision and F-measure evaluation metrics

Experimental Techniques	**Amharic**			**Afaan Oromo**			**Tigrigna**		
	Average R	Average P	Average F-m	Average R	Average P	Average F-m	Average R	Average P	Average F-m
Experiment 1	77.25%	75.34%	77.12%	76%	76.09%	75.59%	73%	73.47%	70.91%
Experiment 2	89.78%	90.19%	90.03%	87.78%	85.02%	85.95%	86%	85.08%	84.24%

Table 5: Proposed model via fixed n-gram and sentence level n-gram feature extraction technique for real-word spelling error detection experimental result per test corpus

As shown from Table 5, the experimental result indicates that a proposed model with sentence level n-gram context feature extraction technique achieves better result. Since, to make the proposed model more effective on detection of real-word spelling errors, the model needs rich contextual feature to learn the properties of language.

However, in case of fixed n-gram (i.e. tri-gram) the number of extracted contextual features are limited and this not enable the proposed model to learn more about language properties. As result, the detection effectiveness of proposed model along with fixed n-gram feature extraction technique is less effective than sentence level n-gram feature extraction technique.

Beside this, as shown in Table 5 the effectiveness of real-world spelling error detection via sentence level n-gram feature extraction vary per language domain. This is due to variation of quality and size of unsupervised training corpus between domain languages.

Moreover, we also evaluate the effectiveness of the proposed model in terms of providing correct suggestions for detected real-word spelling errors. To do so, similarly we evaluate the capability of providing correct suggestions of proposed model using recall, precision and F-measure evaluation metrics. As result, the correction effectiveness of the proposed model along with two experimental techniques (Experiment 3 and 4) are presented in Table 6 below.

Experimental Techniques	**Amharic**			**Afaan Oromo**			**Tigrigna**		
	Average R	Average P	Average F-m	Average R	Average P	Average F-m	Average R	Average P	Average F-m
Experiment 3	76.23%	77.11	76.02	75.33	73.09	74.30%	71.5%	72.14%	71.91%
Experiment 4	88.78%	84.94%	86.89%	83.79%	81.90%	82.03%	82.84%	80.91%	81.04%

Table 6. Proposed model via fixed n-gram and sentence level n-gram feature extraction technique for real-word spelling error correction experimental result per test corpus

Similarly the above experimental result as shown in Table 6 indicates proposed model along with sentence level n-gram feature extraction technique achieves a promising result in terms of providing correct candidate suggestions for detected real-word error words. Since, as we stated earlier the sentence level n-gram able to extract rich contextual features and this enables the proposed model to learn more about properties of language on providing correct candidate suggestions.

However, the proposed model along with sentence level n-gram features extraction technique not more effective on both detection and correction of real-word spelling errors. This is due to lack of using well qualified and large size of training corpus to solve the problem of data sparseness. The larger the corpus the better the expected result on both detection and correction of real-word spelling errors.

4. Conclusion

In this study, we investigate language independent real-word spelling error detection and correction using all possible n-gram features at sentence level. To do so, we incorporate five high level modules such as language selection, sentence segmentation and indexing, n-gram extraction, error detection, and error correction. A large text corpus of domain language was collected and used as training of proposed context-aware spell checker. To evaluate proposed model erroneous test sets are created by exchanging the location of words in random manner due to lack of actual and well organized test set. In this study for demonstration purpose the number of supported language is limited to Amharic, Afaan Oromo and Tigrigna.

This study conducts four experiments to evaluate and compare effectiveness of both fixed n-gram and sentence level n-gram context features on detection and correction of real-word spelling errors. Beside this, for evaluation purpose we adopt recall, precision, and F-measure measurement metrics. The experimental result indicates, using sentence level n-gram context feature achieves a promising result when comparable with fixed n-gram context feature on both detection and correction of real-word spelling errors. Moreover, the researchers recommends to use POS n-gram along with word n-gram, this may resolve data sparseness problem during real-word spelling error detection and correction using sentence level n-gram context feature. Finally, the future work is bound towards extending the proposed model for other languages and improving the performance of the system.

Reference

Amanjot Kaur, Paramjeet Singh, and Shaveta Rani.*(*2015*)*. Spellchecking and Error Correcting System for text paragraphs written in Punjabi Language using Hybrid approach. International Journal of Advanced Research in Science, Engineering and Technology. Vol. 2, Issue 11

Pirinen, Tommi, Lindén, and Krister. *(*2014*)*. State-of-the-Art in Weighted Finite-State Spell-Checking. In: Proceedings of CICLing.

Neha and Pratistha. *(*2012*)*. Spell Checking Techniques in NLP. International Journal of Advanced Research in Computer Science and Software Engineering. Vol2, Issue 12

Fossati, Davide, and Eugenio. *(*2007*)*. A Mixed Trigrams Approach for Context Sensitive Spell Checking. Alexander Gelbukh. Computational Linguistics and Intelligent Text Processing 4394: 623–633

Islam, Aminul, and Diana. *(*2009*)*. Real-word spelling correction using Google Web IT 3-grams. Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing. Vol 3 - EMNLP ’09 3(August):

1241.

Verberne and Suzan. (2002). Context-sensitive spell checking based on word trigram probabilities Context-sensitive spell checking based on word trigram probabilities. Master's thesis, University of Nijmegen, February-August.

Haddad and Yaseen. (2007). Detection and Correction of Non-Words in Arabic: A Hybrid Approach. International Journal of Computer Processing of Oriental Languages (IJCPOL) 20(4): 237–257.

Alkanhal, Mohamed I., Mohamed A. Al-Badrashiny, Mansour M. Alghamdi, and Abdulaziz O. Al-Qabbany. (2012). Automatic Stochastic Arabic Spelling Correction with Emphasis on Space Insertions and Deletions. IEEE Transactions on Audio, Speech, and Language Processing 20(7): 2111–2122.

Ben Othmane and Ben Ahmed. (2012). Detection of semantic errors in Arabic texts. Artificial Intelligence 1: 1–16.

Damerau. (1964). A technique for computer detection and correction of spelling errors. Communications of the ACM, 7(3): 171-176

Levenshtein. (1966). Binary codes capable of correcting deletions, insertions and reversals. Sov. Phys. Dokl. 10 (Feb): 707-710

Wagner and Fischer. (1974). The string-to-string correction problem. JACM 21,1 (Jan.): 168-178

HaBiT Project.(2014). Habit-project.eu. http://habit-project.eu/. Accessed: 14- Jun- 2019.

Truica, Velcin, and Boicea. (2015). Automatic Language Identification for Romance Languages using Stop Words and Diacritics.

Authors Profile

Mr.Tsegay Mullu Kassa received Bachelor Degree in Computer Science from Dilla University, Ethiopia. He received Master's Degree in Information Technology from Jimma University. His research interest includes natural language processing, Information Retrieval, Artificial intelligence, big data, data mining and image processing.

Mirs.Kidst Ergetie Andargie received Bachelor Degree in Computer Science from Dilla University, Ethiopia. She received Master's Degree in Information Technology from Jimma University. Her research interest includes natural language processing, big data, data mining and Information Retrieval.