Context Sensitive Query Correction Method for Query-Based Text Summarization

Nazreena Rahman(✉) and Bhogeswar Borah

Department of Computer Science and Engineering, Tezpur University, Sonitpur 784028, Assam, India
{naz1912,bgb}@tezu.ernet.in

Abstract. Contextual spell correction is very important for real word error correction. It gives the correct word for an incorrect word in a particular sentence. The traditional spell checker can correct those misspelled words which are not present in dictionary but here we try to develop a spell checker which can give appropriate word on the basis of the contextual meaning of the sentence. This spell checker is specially applied for error correction in query-based text summarization. Here, we try to combine both semantic based measure and lexical character matching to find the appropriate word for a particular sentence.

Keywords: Contextual spell correction · Real word error · Query-based text summarization · Semantic based measure and lexical character matching

1 Introduction

Errors in words always play a major issue while retrieving information in the field of natural language processing. Spelling errors can be of two types. One is non-word errors where words are not present in dictionary and do not have meaning. For example, if we do not know the correct spelling of *zebra*, we can write *jebra* in place of *zebra*. From the literature survey it is found that research is done extensively on non-word spell correction. Many traditional spell checking softwares are present through which we can correct these non-word errors easily. Some examples are Microsoft Word, GNU Aspell, Ispell, LibreOffice Writer etc. Microsoft word is a word processing software developed by Microsoft, GNU Aspell is a free and open source spell checker, Ispell is mainly used in Unix that supports many western languages word correction and LibreOffice is also a free and open source office suite. The other type of spelling error is real-word error. Here, spelling error in a word accidentally gives another actual word which is present in dictionary. For example, we write *piece* in place of *peace*. This can be possible by typographical errors or writer might get confused with homophone or near homophones. Homophone is a word, whose pronunciation is same with another word but meaning and spelling is different. For example: here and hear. These type of words are said as confused words. Real-word errors are found more

O. Gervasi et al. (Eds.): ICCSA 2017, Part VI, LNCS 10409, pp. 17–30, 2017.
DOI: 10.1007/978-3-319-62407-5_2

in Dyslexic text [1]. Dyslexia is a disorder that finds difficulty in reading, spelling and writing. From the studies, it is found that from 25% to 40% spelling errors are actually a valid english word [2].

Spell correction in real-word errors is much difficult compared to non-word errors. To deal with real-word error correction, syntactic and semantic analysis should be considered along with pragmatic knowledge about the language. In general, many spell checker give the suggestion of many probable words and user has to choose the correct word according to the context.

The rapid and continuous growth of text information makes it difficult to retrieve the exact information. Therefore, query based text summarization can be used for finding the summarized answer according to user's need. In query-based text summarization, we have a query with single or multiple text documents as input texts. In question answering system or information retrieval system, we need to extract those sentences which have similar meaning with the query. Therefore, it is very much important that the query should not contain any error. When the query is inserted, there is possibility that words in the query might be spelled wrongly. Here, we assume that these incorrect words can be of real-word type of errors. Therefore, to correct this type of real-word errors, we have to consider semantic measures. Semantic measures are always considerd to be an important feature to find the similarity and relatedness among words. Semantics is the study of meaning of word that is used to understand human expression through language. Hence, semantic measure will help us in finding appropriate word for incorrect words.

Word-net (Started by Miller in 1985) relations are applied widely in text analysis and artificial intelligence applications. This lexical database, Word-net is used for english language and was created by Cognitive Science Laboratory of Princeton University. Words with same lexical category are organized into synonym sets called synset. Different kind of synsets are related by different semantic relations in Word-net. Concepts in Word-net are linked together in a hierarchical structure. In Fig. 1, an example of words present in Word-net is shown.

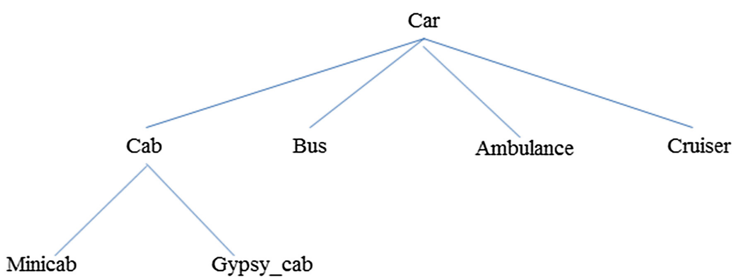

Fig. 1. An example of relationship of words present in Word-net hierarchy

In this paper, a semantics approach (CSQ Method) along with lexical character matching is proposed to find out the correct word for a confused word based on contextually appropriate for the specific sentence. This CSQ method can be applied particularly in query-based text summarization purpose for real-word error correction process. The remainder of the paper is as follows. Section 2 presents various works which are mainly done on real-word context sensitive spell correction, Sect. 3 gives the overview of the proposed CSQ method. Section 4 describes the experimental results. Finally, Sect. 5 covers the conclusion and future plans.

2 Literature Survey

Mays et al. [3] uses maximum likelihood based statistical techniques to find contextually correct word. Their method uses word trigram model. This trigram model computes the conditional probability of a word given by two prior words. This statistical technique models the spelling correction as a speech recognition process. A word string w is generated from text generator. Speller and typist will perform the transformation and a word string y will be produced which might not be similar with w. Finally, linguistic decoder will choose $\hat{w}$ which gives the maximum conditional probability value of w given y.

Golding et al. [4] introduces a new method which is based on trigram and Bayes. Trigram method is based on parts-of-speech trigrams. But this trigram model works only when part-of-speech of the words in the confusion set are different. Here, tagging probability depends on the previous two tags. Moreover, to deal with same parts-of-speech words in the confusion set, their method uses Bayes. Context-word feature based Bayes method uses two types of features: one is context-word and other is collocation. Context-word feature checks if a particular word is present within a certain range of the confused target word and collocation searches for adjacent words of certain length and/or parts-of-speech tagging of confused target word. They combine both the method and named it as Tribayes. This Tribayes uses trigram method for different parts-of-speech tagging and use Bayes method for same parts-of-speech tagging.

Another hybrid Bayesian method [5] is proposed by Golding. This method combines two complementary methods: context word and collocation. Earlier by Yarowsky [6] use decision lists where theses two context word and collocation methods are combined. Decision lists solve the problem by transforming the collected evidence into a single strongest piece of information. Golding initially uses the decision lists hybrid approach to solve the context-sensitive spelling correction problem. However, performance of error correction can be improved by using Bayesian classifier. This classifier does not take only the strongest single piece of evidence but also all available evidences to get better performance. Finally, a further combination is added by using trigram approach when the words in confusion set have different parts of speech and Bayesian approach for same parts of speech.

Hirst and Budanitsky [7] uses Jiang and Conrath semantic similarity measure [8] to find semantically unrelated words according to the context and variation of spelling of words that can be related to the context of the sentence. Their system finds the set of all possible words by insertion, deletion, substitution of single character and transposition of two nearby words. For each word, semantically related words are found from the whole text document and replace the word with highest similarity value.

Fossati and Di Eugenio [9] used Hidden Markov Model (HMM) framework to apply it on mixed trigram model. Each state of the HMM is represented as a pair of parts-of-speech (POS) tagged and a pair of POS tag with a valid dictionary word. The checked word (central word) is considered as a confused word having a set of confusion set. For each confused word present in confusion set, HMM matched the entire sentence. Here, Viterbi algorithm is used to find the most probable sequence of hidden sequence state. If the probability of label of state for a particular central word is higher, then that central word can be taken as a correct word for that particular sentence.

Samanta and Chaudhuri [10] uses bigram and trigram approch to detect and correct real-word errors. Bigram score is calculated by taking left and right neighbor of the candidate key of the sentence and trigram score is generated by the these three words. They consider single error in word detection and correction. Since this correction method depends on the immediate left and right of candidate word, hence their approach can correct errors appearing in alternate words. Initially, they find confusion set for corresponding candidate word using Levenshtein distance [11] from the dictionary. Their model calculates bigram and trigram probability score for each word in confusion set by using Markov chain rule. Here, BYU corpus is used to find n-gram probability. By using Maximum Likehood Estimation, bigram and trigram probabilities are obtained. Sometimes many proper bigrams and trigrams are not found in the corpus, hence they stem the words to increase the accuracy of their model.

Sharma et al. [12] proposes a model for real-word error correction. They use collocation feature by finding the presence of neighboring words. Trigram probability is calculated for each word in the confusion set and highest probability is considered as right one. They also use Bayesian approach to find context features. Bayesian approach finds all the words surrounded by the target words and calls it as features. It finds all nearby words of the target word and calculates the probability of textual information using a training corpus. Their method also uses synonyms of the contextual word if the exact word is not present in corpus. Thus, finally the highest score word is considered as a correct word. Sorokin [13] presents an automatic spelling correction algorithm. The algorithm uses noisy channel model and re-ranking of hypothesis based on features. This language independent model is applied for Russian language. The word-level and the sentence-level features are integrated here. There are three steps to find out the correct word: first step is candidate generation, second step is extraction of n-best list and final step is the feature-based ranking of hypothesis.

From the above study, it is seen that there is no context based spell checker which is used for spell correction particularly in query-based text summarization. However, existing spell checkers can be used for spelling correction in query-based text summarization, but efficiency is quite low while applying this spell checkers. Hence, we try to propose one context sensitive query correction method for query-based text summarization.

It is observed from the extensive analysis and survey of literature that n-gram matching similarity always plays a vital role while dealing with context-sensitive real-word errors. Additionally, we can strengthen the spell checker by using Word-net. Word-net gives different semantic relations which will eventually help us to find out the exact word for a specific sentence.

3 Overview of the Proposed Method: CSQ

3.1 Proposed Framework

The proposed method is a real-word error correction method. We try to correct those real words which are normally considered as confused words. These words can be obtained with the help of the commonly confused word list from the Random House Unabridged Dictionary [14].

Contextual spelling correction method always finds appropriate word which will be suitable for that specific sentence. In general, meaning of a word depends on the immediate left and immediate right words of that particular word. In fact, it is seen that strong semantic relation is there between the previous word (left word) and next word (right word) with the target word. Semantic measure gives contextually similar words which will be appropriate for the incorrect word present in a sentence. Semantic measure can be calculated by using semantic similarity and semantic relatedness. We try to find out the semantic measure by using neighboring contextual information with the target confused word. Semantic similarity always finds similar meaning words but semantic relatedness does not mean that two words or concepts are similar. Semantically related two words are said to be related words by considering their likeliness. For example, in bank, a bank account and a customer are related, hence there is a strong semantic similarity in between the two concepts. With the help of a lexical relation, different concepts are semantically related: like meronymy (hand-finger), antonymy (good-bad). It is also seen that any kind of functional relationship like frequent association or co-occurrence of ideas (tea-India) can be considered as semantically similar concepts.

Two concepts can be related by various ways by not considering only the similarity of two words. To find semantic relatedness, the follows relations are used:

1. Synonymy: Words that sound different but have the same or identical meaning as another word; example: ‘achiever’ is the synonym of ‘success’.
2. Hyponymy: Word or phrase whose semantic field is more specific; example: ‘domestic_cat’ is the hyponym of ‘cat’.
3. Hypernymy: It is also known as a superordinate, is broader than that of a hyponym; example: ‘feline’ is the hypernym of ‘cat’.

4. Meronymy: Denotes a constituent part of, or a member of something; example: 'heart' is a meronym of 'body'.
5. Holonymy: Defines the relationship between a term denoting the whole and a term denoting a part of, or a member of, the whole; example: 'body' is a holonym of 'heart'.
6. Troponymy: Troponymy is the presence of a 'manner' relation; example: 'smile' is troponym of 'laugh'. It is used for verb.
7. Entailment: Any verb A entails verb B, if the truth of B follows logically from the truth of A; example: 'morning walk' is the entailment of 'early rising'.
8. Antonymy: A word having opposite meaning; example: 'bad' is a antonym of 'good'.

Now, to find out the semantic relatedness, we use hso similarity (Hirst and St-Onge) [15] using Word-net. The main idea of hso measures is that the similarity between two concepts is a function of the length of the path linking the concepts and the number of directions between the two concepts in the taxonomy. This semantic relatedness measure includes has-part, is-made-of, is-an-attribute-of type of relations. It is more generalized concept than semantic similarity concept. We can find similarity between words of different parts-of-speech. This path based measure classifies relation in Word-net in terms of direction; for example upward direction *is-a*, horizontal direction *has-part*. The following Fig. 2 shows the path lengths and Fig. 3 shows the number of changes of directions between two concepts or words that are present in Word-net. Here, we consider *is-a* relation between two nouns.

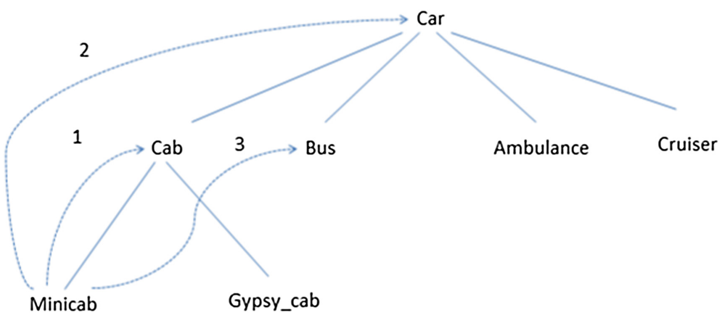

Fig. 2. A fragment showing path lengths in Word-net hypernym hierarchy

In Fig. 2 it is shown that path lengths between Minicab to Cab is 1, Minicab to Car is 2 (path is from Minicab to Cab and Cab to Car) and Minicab to Bus is 3 (path is from Minicab to Cab, Cab to Car and Car to Bus). Shortest path similarity says that if the shortest path distance between two sense or words in a graph is short, it signifies that two words are more similar. In fact, shortest path similarity calculates the number of edges between two concepts or words in the

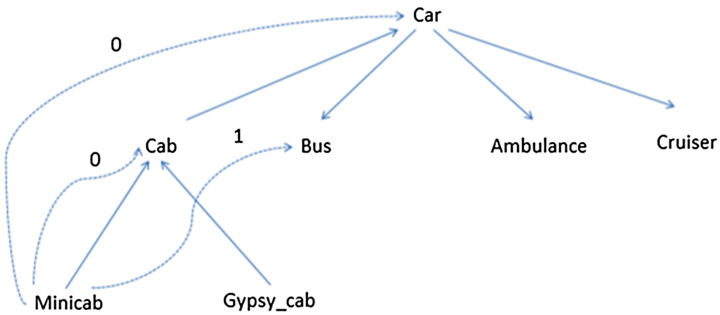

Fig. 3. A fragment showing changes of path directions in Word-net hypernym hierarchy

thesaurus graph. Here, two words having same parents are considered as more similar and words that are far away in the network are considered less similar.

From the Fig. 3, it is clear that the number of changes of direction between Minicab to Cab and Minicab to Car is 0. But, for Minicab to Bus, number of changes of direction is 1, as to go from Minicab to Bus, we have to go first to upward direction (from Minicab to Car) and again have to traverse to downward direction (from Car to Bus). Therefore, semantic relatedness always makes an effort to find a path which is not too long and also number of changes of direction is less. The required hso equation is

$$path\,weight(c_1, c_2) = 2 * c - path\,length\,(c_1, c_2) - (k * direction\,changes\,(c_1, c_2)) \tag{1}$$

Here, c and k are the constants and values are $c = 8$ and $k = 1$. We have to normalize the semantic relatedness value as we get 16 as highest score if two words are completely similar. For, normalization, we use the following equation:

$$\frac{x - x_{min}}{x_{max} - x_{min}} \tag{2}$$

Moreover, we try to find similarity of confused words based on lexical character matching. Here, we assume that the appropriate word in the confused wordset may be present in the title of the input text document. Therefore, we take each word in the input text title and match with the words present in commonly confused wordset. Scores are given based on the same number of characters present between input text title words and words in confused wordset. Here, longest common contiguous subsequence is considered in sequence matching. The equation for sequence matching between two words or strings is as follows:

$$Sequence\,Matching\,Score = 2.0 * \frac{M}{T} \tag{3}$$

Here,
M = Matching characters between two strings.
T = Total number of characters present in both the strings.

Now, these two similarity scores are added. The equation will be

$$Total\ Similarity\ Score = A + B \tag{4}$$

To get more accuracy in result, we use weighting parameter for two different similarity values. Hence, the equation for total similarity will be:

$$Total\ Similarity\ Score = \alpha * A + \beta * B \tag{5}$$

Presence of direct word always gives significant result, hence we give higher priority to sequence matching character similarity score (B) than semantic relatedness scores (A). We tested the method on training set and optimize the accuracy by giving following values of α and β: $\alpha = 0.40$, $\beta = 0.60$; where $\alpha + \beta = 1$. Range of total similarity score is 0$\leq$ Total Similarity Score $\leq$1.

The method chooses that word having the highest score. Finally, replacement of the correct spelled word with the incorrect word is done to get the correct query.

3.2 Description of Proposed Method (CSQ)

In query-based text summarization, user has to enter the query. It is quite possible that a user may enter incorrect words. Hence, we assume that the query contains wrong words.

The pseudo code of Context Sensitive Query Correction Method for Query Based Text Summarization (CSQ) method is as follows:

```
Data: Query (Q_i) and Title of the Input Text (T_i)
Result: Correct Query (Q_correct)
Do the stop word removal and stemming of the query
for each confused word w in Q_i do
    Find out the confused wordset (C_w) using dictionary
    for each word c in C_w do
        Find out the average semantic measure score of the word c with
        the previous word P_w and the next word N_w of Q_i using Eq. 1
        Find out the sequence matching of the target word with the title
        of the input text document (T_i) using Eq. 3
        Sum up all scores (score) using Eq. 5
        Replace w with c having highest score
    end
end
```

Algorithm 1: Steps of CSQ Method

3.3 Example Computation

To experiment the proposed method, DUC 2005 datasets are used (http://duc.nist.gov). Each dataset has 50 queries with 50 different topics. We take those queries where confused words are present. CSQ method is applied on DUC 2005 datasets. Here, we try to show it by taking a simple example. We take the query example as, "Identify and describe types of organized crime that crosses borders or involves more than one country". When the query is entered, it is written with spelling mistake as "Identify and describe types of organized crime that crosses *boarders* or involves more than one country" Now, we apply the CSQ method and results are as follows:

1. Pre-process the query by removing the stop words and stemming the query.
2. Initially, CSQ Method checks the presence of confused words in the query. Here, it finds the confused word 'boarder' and for 'boarder' the method gets this confusion wordset {boarder, border} accordingly. The score of semantic measure score (A) of confused word with next and previous word is shown in Table 1.

Table 1. A: Score

Confused words	Score1
Boarder	0.05
Border	0.08

3. Sequence matching of the target word from the confusion set is calculated with the title of the input text document and score (B) is shown in Table 2.

Table 2. B: Score

Confused words	Score3
Boarder	0.61
Border	0.67

4. Finally, we calculate the total similarity value by adding two similarity scores (Table 3):

Table 3. Total similarity scores of confused words

Confused words	Total score
Boarder	0.386
Border	0.434

Here highest score is 0.434 for "border". Hence "boarder" will be replaced by "border" in the query and it will be written as, "Identify and describe types of organized crime that crosses borders or involves more than one country".

4 Experimental Results and Discussion

4.1 Comparison of CSQ Method with Real-Word Spell Checkers

We try to evaluate CSQ method with other existing real-word error correction methods. Here, Baseline method, Hidden Markov Model Tagger and popular Ginger Software are used to compare with CSQ method. We use DUC 2005 and 2006 datasets. Each dataset contains 50 queries with 50 different topics. We take those queries where confused words are present.

The baseline method is based on frequency of occurrence of a word in the training corpus. For example, if the confused set is {piece, peace}, then the baseline method finds most occurred word between these two confused words from the training corpus and it suggests to change (or remain same) the word to the most common word in the test corpus.

The hidden markov model (HMM) is mainly used for parts of speech tagging. This probabilistic tagger chooses the tag sequence with highest probability. Hidden markov model is a special case of Bayesian interference for noisy-channel models. It helps to find out the appropriate word according to the context of the sentence. HMM tagger only works for different parts-of-speech words. The ginger spell checker checks every word depending on the context of the sentence.

To find out the accuracy of the spell checker, we use the following equation:

$$Accuracy = \frac{words\,correctly\,recognized\,by\,spell\,checker}{Total\,no\,of\,confused\,words} \tag{6}$$

To find out the sentence level precision and recall, we use the following measures:

- true positive (TP): correct word which is recognized as correct word by the spell checker.
- false positive (FP): incorrect word which is recognized as correct word by the spell checker.
- false negative (FN): correct word which is recognized as incorrect word by the spell checker.
- true negative (TN): incorrect word which is recognized as incorrect word by the spell checker.

To know about the capacity to detect correct sentences, we can use following recall and precision equations.

$$R_c = \frac{TP}{TP + FN} \tag{7}$$

$$P_c = \frac{TP}{TP + FP} \tag{8}$$

F-measure gives the harmonic mean of recall and precision values. Hence, the equation will be:

$$F = \frac{2 * R_c * P_c}{R_c + P_c} \tag{9}$$

Table 4. Performance measure with baseline systems for real-word errors

Method name	Accuracy	Recall	Precision	F
CSQ	76.92%	83.33%	90.90%	86.95%
Baseline method	38.46%	41.66%	83.33%	55.54%
HMM tagger	15.38%	16.66%	66.66%	26.65%
Ginger	46.15%	50%	85.71%	63.15%

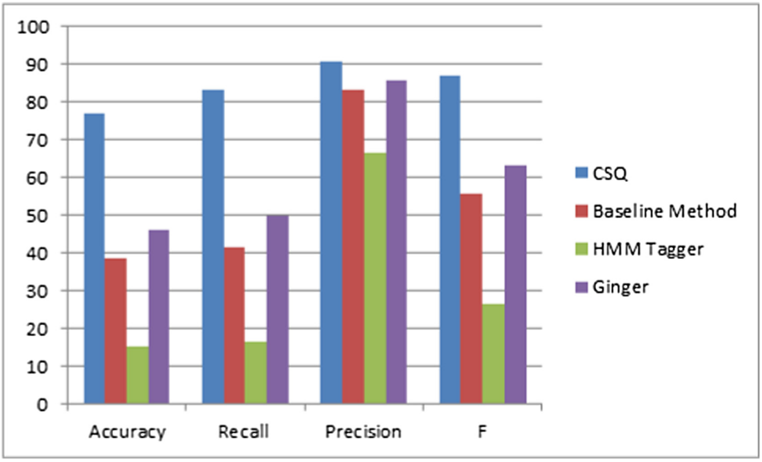

Fig. 4. Comparison of performance between CSQ and other existing methods for real-word errors

By using the above equations, CSQ method is compared with other existing systems. Results are shown in following Table 4 and Fig. 4.

From the above comparison, it is clear that CSQ method performs better in terms of all accuracy, precision, recall and F-measure values.

4.2 Comparison of CSQ Method with Non-word Spell Checkers

This CSQ Method also works well for non-word error detection and correction. Hence, we can also apply it for non-word error correction. To apply this method, we just need to find out the suggested wordset using dictionary. We apply CSQ method with every word in suggested wordset. Here, 50 queries are taken for doing the evaluation. We use TAC (Text Analytics Conference) 2009 datsets (http://tac.nist.gov). There are 44 documents each having 2 topics. For each topic, there are ten text documents.

We compare the CSQ method with some baseline methods like Longest Common Substring, Character Similarity, Microsoftword Spell Corrector and Minimum Edit Distance. Longest common substring (LCS) finds the longest string or strings between two or more strings. Character similarity (CS) finds the maximum common characters between two strings.

In a word document, Microsoft Spell Corrector (MSC) gives many suggested words for an incorrect word. Minimum Edit Distance (MED) gives minimum number of operations required to transfer from one word to another word. Insertion, deletion and substitution operations are used here for transformation from one string to other string. Detailed results are shown in following Table 5 and Fig. 5.

Table 5. Performance measure with baseline systems for non-real word errors

Method name	Accuracy	Recall	Precision	F
CSQ	82%	83.67%	97.61%	90.10%
LCS	18%	69%	64.3%	66.6%
CS	48%	85%	82.8%	83.9%
MSC	64%	89.2%	89%	89.1%
MED	42%	84%	80.8%	82.4%

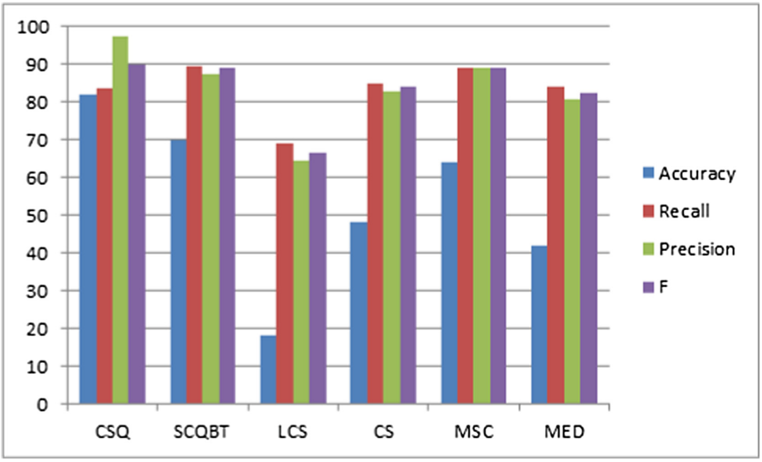

Fig. 5. Comparison of performance between CSQ and other existing methods for non-word errors

From the above comparison, it also proved that this CSQ method performs well in terms of accuracy. We also get high precision and F-measure value for all other existing systems except for recall values. Hence, we can use CSQ method both for real word and non-real word errors.

5 Conclusion

An effective context sensitive spell checker is suggested here which is based on both semantic measure and lexical character matching. Lexical character matching can be achieved by finding similar characters between two words and semantic measure can be found by using different semantic relations. CSQ method is helpful for real-word as well as non-word spelling correction particularly in query-based text summarization. This spell checker can correct more than one incorrect word present in a sentence. Semantic measure is calculated by using hso similarity. This new spell checker outperforms many existing contextual real -word and non-word spell checkers.

Though CSQ method performs substantially high compared to other existing spell checkers, but this method is only implemented on small confusion wordset. We can improve the scalability of identification and correction of wrong words by applying on large confusion wordset. This CSQ method also depends on word bi-gram. Therefore, if the incorrect word is the first or last word of the sentence, then bi-gram semantic measure can not find due to absence of previous or next word of the target word. In addition, we can improve this CSQ spell checker by introducing new similarity measure which can find semantic measure between words of different parts of speech.

References

1. Pedler, J.: Computer correction of real-word spelling errors in dyslexic text. Ph.D. thesis, University of London (2007)
2. Kukich, K.: Techniques for automatically correcting words in text. ACM Comput. Surv. (CSUR) **24**(4), 377–439 (1992)
3. Mays, E., Damerau, F.J., Mercer, R.L.: Context based spelling correction. Inf. Process. Manag. **27**(5), 517–522 (1991)
4. Golding, A.R., Schabes, Y.: Combining trigram-based and feature-based methods for context-sensitive spelling correction. In: Proceedings of the 34th Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, pp. 71–78 (1996)
5. Golding, A.R.: A Bayesian hybrid method for context-sensitive spelling correction. arXiv preprint arXiv:cmp-lg/9606001 (1996)
6. Yarowsky, D.: A comparison of corpus-based techniques for restoring accents in Spanish and French text. In: Armstrong, S., Church, K., Isabelle, P., Manzi, S., Tzoukermann, E., Yarowsky, D. (eds.) Natural Language Processing Using Very Large Corpora, pp. 99–120. Springer, Heidelberg (1999)
7. Hirst, G., Budanitsky, A.: Correcting real-word spelling errors by restoring lexical cohesion. Nat. Lang. Eng. **11**(1), 87 (2005)
8. Jiang, J.J., Conrath, D.W.: Semantic similarity based on corpus statistics and lexical taxonomy. arXiv preprint arXiv:cmp-lg/9709008 (1997)
9. Fossati, D., Di Eugenio, B.: I saw tree trees in the park: how to correct real-word spelling mistakes. In: LREC (2008)
10. Samanta, P., Chaudhuri, B.B.: A simple real-word error detection and correction using local word bigram and trigram. In: ROCLING (2013)

11. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet Physics Doklady, vol. 10, pp. 707–710 (1966)
12. Gupta, S., et al.: A correction model for real-word errors. Procedia Comput. Sci. **70**, 99–106 (2015)
13. Sorokin, A.: Spelling correction for morphologically rich language: a case study of Russian. In: BSNLP 2017, p. 45 (2017)
14. Flexner, S.B., Hauck, L.C., et al.: Random House Unabridged Dictionary. Random House, New York (1993)
15. Hirst, G., St-Onge, D., et al.: Lexical chains as representations of context for the detection and correction of malapropisms. WordNet: Electron. Lexical Database **305**, 305–332 (1998)

Computational Science and Its Applications – ICCSA 2017
17th International Conference, Trieste, Italy, July 3-6, 2017, Proceedings, Part VI
Gervasi, O.; Murgante, B.; Misra, S.; Borruso, G.; Torre, C.M.; Rocha, A.M.A.C.; Taniar, D.; Apduhan, B.O.; Stankova, E.; Cuzzocrea, A. (Eds.)
2017, XXXVI, 799 p. 315 illus., Softcover
ISBN: 978-3-319-62406-8