Detection and Correction of Accidental Semantic Errors

Dong-Joo Kim[1] and Han-Woo Kim[2]
[1]Department of Computer Engineering, Anyang University
708-113 Anyang5-dong, Manan-gu, Anyang, Kyeonggi-do, 430-714, Korea
[2]Department of Computer Science and Engineering, Hanyang University,
1271 Sa3-dong, Sangnok-gu, Ansan, Kyeonggi-do, 425-791, Korea
E-mail : [1]djkim@anyang.ac.kr, [2]kimhw@cse.hanyang.ac.kr

Abstract: One of the most important characteristics of accidental errors to result from simple mistyping is that there is serious discrepancy between erroneous words and their surrounding context. To detect and correct these errors, contextual information such as n-gram or co-occurrence frequency is needed. However, despite these and other advances, detection and correction of accidental errors in agglutinative languages such as Korean is crawling yet due to relatively freely movable component. This paper presents another method to detect and correct accidental errors using abstract dependency structure to remove functional dependency relations. To capture contextual information of word from abstract dependency structure, our method uses the co-occurrence frequency for words in immediate dependency relations between content words.

1. Introduction

A critiquing system or a spelling checker is the proofreading system that helps to write the right documents through the detection and correction of textual errors: typographical, orthographical, syntactic, semantic, and stylistic errors. The causal factor of typographical and orthographical error exists within the erroneous word, while that of syntactic or semantic error is able to be found at the relation between it and external constituents. Most sentence level errors, such as syntactic and semantic errors, are caused by writer's shortage of linguistic knowledge, while word level errors, such as typographical[1] and orthographical[2] errors, are caused by changes of some characters within a word throughout simple input mistakes or ignorances. The study about this system was begun at 1960s[1] and recently semantic errors and even style errors are regarded as targets of detection and correction.

The detection of sentence level errors needs information about syntactic and semantic relations between words or phrases, irrespective of whether those information is based on the rule or statistics. Also, those sort of errors can be corrected with information similar to that used to detect. However, the detection of typographical and orthographical word level errors differ entirely from it of sentence level errors. Word level errors can be detected by means of the analysis of morphological inner structure, or just simply finding of word from legitimate word list. These errors can be corrected through the alteration of the suspicious letter. In other words, a simple mistyping error may be corrected by insertion, deletion of suspicious letter, and substitution of it to a letter adjacent on the keyboard[1].

However, some typographical errors are already valid words. That is to say, erroneous words by simple input mistakes accidentally exist within legitimate word list, such as *How does antibiotic therapy duffer from conventional therapy*, where *duffer* was typed when *differ* was intended. These errors by simple mistyping result in semantic or syntactic error. Kukich[4] have reported that these sorts of errors account for anywhere from 25% to 50% of word-based errors in a text through the application areas. Many previous conventional systems relying on only reference of word lists can not detect these accidental errors. The most important characteristic of accidental errors are semantically or syntatically inconsistent with surrounding context. Therefore, in order to make appropriate correction of such errors, it seem to require context-sensitive information like partial rules by the subsentence or phrase, n-gram, co-occurrence frequency, or collocational information. Many researchers proposed context-sensitive methods to detect and correct these acidental errors using trigram[5] information, Bayesian classifier[6], decision lists[7], transformation-based learning[8], and latent semantic analysis[9].

In spite of their many achievements, it is still hard to detect and correct accidental errors at agglutinative language. One characteristic of agglutinative languages, such as Korean, Japanese and Turkish, is that the word order is highly flexible. A word, a separable units by space character in Korean, at agglutinative languages consists of two parts, content part and function part. Each part may have serveral morphemes; content part consists of content morphemes and derivational components, and function part consists of function morphemes and inflectional components. Generally, content part can stand alone, but function part always have to be attached to content part. According to these morphological constructions, separable spacing units are not only able to be free from the orderedness of words, but also syntactically movable almost anywhere in a sentence.

Therefore, the countable units like n-gram reflecting just only local context does not capture surrounding context. To capture the context of freely movable components, we will use not general co-ocurrence frequencies for just adjacent words but special co-occurrence frequencies for grammartical relation. By the way, function words tend to be attached to almost all content words and they do not contribute to discrimination. Consequently, we will not consider the relation between funtion word and content word but the relation between content words.

To achieve this goal, we propose the abstract dependency structure, which is the structure removing the dependency between function words and content words. We will count the co-occurrence frequency of pairs with immediate dominant

relation as syntactic and semantic context from abstract dependency structure. After then, we will regard the center word of the context to have low frequency as errouneous. We will try to correct detected errors through the simlar method to previous conventional spelling checkers because they are caused by just simple mistyping. Finally, to evaluate our method, we generate documents randomly including artificial accidental errors. We will not use sentences in agglutinative langugages but English sentences for convinience of experiments; easy acquisition of data and tools and simple test through fast development of prototypes. We think that it may be enough to prove efficiency of our method though sentences used for our experiment are not written in aggultinative language but in English.

2. Previous works

To detect spelling errors, the previous systems exploit available word lists which are lists of reasonable words[1] by considering the four major error types: insertion, deletion, substitution, and transposition. An initial pioneer work[10] generated all candidate corrections by operations corresponding to four error types, and found all probable corrections by filtering them through dictionary look-up. Systems using these methods decides it as erroneous word if the word to be looked over is not found within legitimate word list. And if it is not matched with surrounding context, though there exists within word lists, the system decides it as syntactic or semantic error. As mention in section 1,these errors by simple input mistake, accidental errors, take up a considerable part compared with pure syntactic or semantic error. To detect accidental errors, many researchers proposed the class of the context-sensitive approaches[3][5][6]. They decide basically current examming word as erroneous word if the occurrence probability of a certain word sequence is very low. Contrary to detection of error words, they consider the word as a candidate correction if the probability is very high.

However, they do not capture long-distance dependencies[8]. To overcome this problem, Kang[11] proposes a partial parsing method using error-pattern about the types of errors frequently to mistake. His method for detection of syntactic or semantic errors, similarly to the detection of typographical errors, tends to depend on the reference of knowledge-base. In other words, the sort of this system makes a decision on the basis of whether or not the rules about context exist within the knowledge-base. The knowledge-base has a set of rules only about the context errors as anyone could notice or people frequently make a mistake, because it cannot hold information about every context. These knowledge-based approaches have the advantage in high precision, but also have the disadvantage in low recall rate because they cannot find all patterns for various error contexts. Besides it, the scalability of these approaches is very low because it is not easy to expand rules. Also, they are not robust since it is not capable of coping with small variations of its context.

Above all things, long-distance problem is more serious in the case of agglutinative languages that have space separable component such as *eojeol* in Korean to be possible to move anywhere in a sentence. It is harder to capture the regularized context irrespective of whether by means of statistical method or knowledge-based method. This problem restricts Kang's error-pattern[11] for just only rigid usage form, such as idiomatic and collocational expression, to be created.

3. Detection of accidental errors

This paper proposes a method using statistical context information to overcome the limits of knowledge-based approaches for the detection of syntactic and semantic errors. The major orthographical errors are due to simple input mistakes. Of course, many of errors occurred by these mistakes are just spelling errors, but a quite part of those errors are expanded to syntactic or semantic errors. In other words, unintended words accidentally existed on legitimate word lists are more than pure syntactic or semantic errors. The subjects of our study are limited to syntactic and semantic errors by accidental mistyping, except pure syntactic and semantic errors.

An important characteristic of these kinds of error is that an erroneous word is very inconsistent with its surrounding words. Therefore, this paper tests the associativity of an error word with its surrounding words to detect accidental syntactic and semantic errors.

To detect accidental syntactic and semantic errors, we extract statistical information reflecting co-occurrence of a suspicious word with its surrounding words from the set of correct sentences at a certain domain. The region of fragment to count co-occurrence frequency is restricted within a sentence including the suspicious word. However, the most function words of surrounding words adjacent to the suspicious word are not semantically interrelated. Therefore, to count co-occurrence frequency for directly correlated word pairs, we propose the abstract dependency structure extracting the relations between only content words from the traditional dependency structure.

3.1 Abstract dependency structure

Dependency structure[12] is a class of syntactic structure formulated by the French linguist Lucien Tesnière. Structure is determined by the relation between a word (a head) and its dependents. A relation is a directed edge, and the dependent is the modifier or complement and the head determines the attribute of the dependent. Dependency structures are well suited to language with free word order, such as Korean, since they do not specify a concrete word order. Main verb *is* at Figure 1 is root node to be independent and headless. Root has two dependents *This* and *example*. That is to say, root node *is* dominates two dependents *This* and *example*. Contrary to dominating relation, a dependent *This* is subordinate to *is*, a dependent *example* and other all component are the same.

There exist all six relations , (*is*, *this*), (*is*, *example*), (*example*, *an*), (*example*, *of*), (*of*, *grammar*), and (*grammar*, *dependency*) in Figure 1. We will count the co-occurrence frequency by every pairs to extract contextual information. However, relations between words and functional words are not distinctive to capture surrounding context since they are

able to co-occur with almost every words. Therefore, we will count the frequency for relations between only content words such as adjective, averb, verb, and noun. If function words intervene between content words then they are removed, like as *example* → *of* → *grammar* to *example* → *grammar*. We will call this structure, such as Figure 2, to *abstract dependency structure*. Therefore, we will count every pairs from the abstract dependency structure as co-occurrence frequency, like as (*is*, *example*), (*example*, *grammar*), and (*grammar*, *dependency*). Their directionality may be ignored to lessen the data sparseness problem. That is to say, (*is*, *example*) and (*example*, *is*) may be dealt as same thing. Futhermore, the relations jumping over a word may be counted as co-coccurrence frequency to reduce the adverse impact on low frequence. For example, if there exist (*is*, *example*) and (*example*, *grammar*) relations, even the (*is*, *grammar*) relation may be counted, in which case all relations in Figure 2 are (*is*, *example*), (*example*, *grammar*), (*grammar*, *dependency*), (*is*, *grammar*), and (*example*, *dependency*).

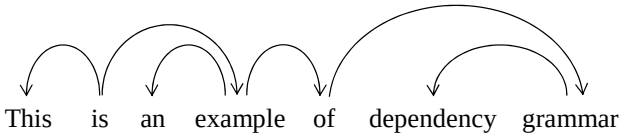

Figure 1. An example for the original dependency structure

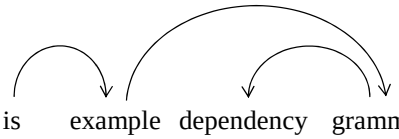

Figure 2. An example for the abstract dependency structure

Of course, this frequency may fail to catch more useful information for surrounding context, specially about syntactic information. Many function words are often isolated in inflectional languages such as English, and it is impossible for them freely to move without any legal reason. Accordingly, their position plays a very important syntactic role since they are not able to have full use of their syntactic facilities until they are placed at the proper position.

In comparison with words in inflectional languages, words in agglutinative languages such as Korean are more freely movable because they have already both semantic and syntactic elements. In addition to this higher mobility, function words in such languages can not stand alone and they have to be attached with content words. Therefore, there is nothing they can do but play a role as complements. Of course they have important role to determine case like subjective case, objective case, and so on. Nevertheless, this work does not consider these semantic cases since it concentrates on only relations between semantic contents.

3.2 Detection

Ideal detection algorithm of accidental errors is simliarly to procedure to count co-occurrence frequency. In order to get abstract dependency structure and extract relations between content words within a sentence, first, the algorithm carry out parsing target sentence. After then, it examines the relative frequency of extracted relations from the collection of co-occurrence frequency. If the relative frequency of a word is more than a certain threshold value, the algorithm decides it as valid word, if not, the algorithm decides it invalid word, accidental error.

However, if there exists a erroneous word within the sentence, it is quite possible to fail to parse the sentence and it may be impossible to get abstract dependency structure. Therefore, we present a method to examine all pairs of content words within a sentence.

A head word in dependency relation has quantitative valency which refers to the capacity of a word to take a specific number and type of arguments. These quantitative valency is from 0 (monovalent), 1 (univalent), 2 (divalent) to 3 (tryvalent), which is the number of obligatory or optional complements. And each word within a sentence must have just one ascendant except a root node. Therefore, the minimum number of relations within a sentence is the sentence length (the number of consisting words) minus 1. Also, the maximum number of relations is three times the sentence length. Of course, there is absolutely not in case three times the sentence length since it is impossible for all words within a sentence to be intransitive verb and to have two object; a direct object and a indirect object.

Therefore, all pairs of content words within a sentence are examined, if a word has more than one relation larger than a certain thershold value, it is decided as a legitimate word. However, it is more possible for each word to have relations in a longer sentence. To avoid this problem, we have to do the normalization as below equation.

$$nv = \frac{\text{sum of relative frequences for all word relations}}{\text{\# of content words}}$$

For example, if a certain content word w_i has relations of w_j, w_{j+1}, ..., w_{j+n} within a sentence whose the number of content words is l ($l \geq n+1$), then normalized value $nv = \sum_{p=0}^{n} rel\_freq(w_i, w_{j+p})/l$

4. Correction

Detected accidental syntactic and semantic errors result from simple mistyping. Consequently, our proposed system corrects them using the method similar to spelling error correction according to some empirical principles[1][10], because erroneous words are typographically not much different from intended words. However, they have already existed on the word lists and look like legitimate words. Therefore, it is impossible to we try to correct them with two different methods.

The first method is to look up the most similar word to the accidental error word within the word lists except itself, where words to be looked up are limited to words adjacent to the erroneous word. Another is the method based on the locality of word occurrence, which is the property that the likelihood of occurrence of word is greater at a spatial location near

the occurred word. Accordingly, the method based on the locality is to look up the most typographically similar word of words within a certain window area. The first method has disadvantages that the extent having to be searched is very large and the corrected word may be changed into another syntactic or semantic error. On the other hand, it has an advantage that adaptability for low frequency words may be able to be improved. The second method has a serious problem that it is less likely to correct errors occurred in short documents. However, it has an advantage that its correctness for long documents is very high.

5. Evaluation

In order to evaluate the correctness of our system, we selected about two hundred sentences of less than 15 length from *SUSANNE* corpus[13], and removed the annotated phrase-structure information since we intended to use information of the dependency relation. After then, we randomly selected again about twenty sentences from them, and automatically generated erroneous sentences including one imaginary accidental error word by an error-free sentence. These errors are different from original error-free word by substitution, insertion, or deletion of a certain character. Other remaining 180 sentences was used to capture surrounding context through counting co-occurrence frequency and they were manually annotated with the abstract dependency structured information. Experiment was performed for erroneous words except errors to be able to be detected by the critiquing system included at word processor of Microsoft corporation because most errors to be detected by this checker are just simple spelling error.

Experiment was performed by varying threshold values. Co-occurrence frequency information used was two types: one was the frequency reflecting directionality and another was no reflecting. In addition to these two types, we used also the frequency information of immediate dominant or subordinate relation and of relation jumping over one word. In the case of correction, we used two types, method based on the reference of legitimate word lists and based on the locality of word occurrence. The best correctness of our system was about 74% and recall rate was about 86% for detection.

6. Conclusion

This paper presents a context-sensitive method to detect and correct accidental errors in agglutinative languages, such as Korean. Although we used English sentences to evaluate our method just by the reason of short development time and more convinient testing environment, we presented a possibility of successful detection and correction of accidental errors in agglutinative languages. Remaining issues are to test for more many sentences and to evaluate for sentences of agglutinative language.

References

[1] F. Damerau, "A Technique for Computer Detection and Correction of Spelling Errors," *Communications of the ACM*, vol. 7, no. 3, pp.171-176, 1964.

[2] C. Sterling, "Spelling Errors in Context," *British Journal of Psychology*, vol. 74, pp.535-364, 1983.

[3] A. R. Golding and D. Roth, "A Winnow-Based Approach to Context-Sensitive Spelling Correction," *Machine Learning*, vol. 34, no. 1-3, pp.107-130, 1999.

[4] K. Kukich, "Automatic Spelling Correction: Detection, Correction and Context-Dependent Techniques," Technical Report, Bellcore, Morristown, NJ 07960, 1991.

[5] E. Mays, F. Damerau, amd R. Mercer, "Context Based Spelling Correction," *Information Processing and Management*, vol. 25, no. 5, pp.517-522, 1991.

[6] W. Gale, K. Church, and D. Yarowsky, "A Method for Disambiguating Word Senses in a Large Corpus," *Computers and the Hymanities*, vol. 26, pp.415-439, 1993.

[7] D. Yarowsky, "Decision Lists for Lexical Ambiguity Resolution: Application to Accent Restoration in Spanich and French," In *Proceedings of* 32*nd Annual Meeting of the Association for Computational Linguistics*, Las Cruces, NM, 1994.

[8] L. Mangu and E. Brill, "Automatic Rule Accquisition for Spelling Correction," In *Proceedings of* 14*th International Conference on Machine Learning*, Morgan Kaufmann, 1997.

[9] M. Jones and J. Martin, "Contectual Spelling Correction Using Latent Semantic Analysis," In *Proceedings of* 5*th Conference on Applied Natural Language Processing*, Washington, DC, 1997.

[10] J. Peterson, "Computer Programs for Detecting and Correcting Spelling Errors," *Communications of the ACM*, vol. 23, pp.676-687, 1980.

[11] K. Mi-Yong, Y. Aesun, and K. Hyuk-Chul, "Improving Partial Parsing Based on Error-Pattern Analysis for a Korean Grammar-Checker," *ACM Transactions on Asian Language Information Processing*, vol. 2, no. 4, pp.301-323, 2003.

[12] L. Tenière, *Èlèments de Syntax Structure*, Eds. Klincksieck, 1959.

[13] G. Sampson, *English for the Computer: The SUSANNE Corpus and Analytic Scheme*, Oxford University Press, Oxford, 1995.