Correcting Misspelled Words in Twitter Text

Jeongin Kim[1], Eunji Lee[1], Taekeun Hong[2], and Pankoo Kim[1](✉)

[1] Department of Computer Engineering, Chosun University, Gwangju, Republic of Korea
jungingim@gmail.com, eunbesu@gmail.com, pkkim@chosun.ac.kr
[2] Department of Software Convergence Engineering, Chosun University,
Gwangju, Republic of Korea
goodfax2000@naver.com

Abstract. The SNS became popularized by computer, mobile devices, and tablets that are accessible to the Internet. Among SNS, Twitter posts the words of short texts and, it shares information. Twitter texts are the optimal data to extract new information, but as it may contain the information within the limited number of words, there are various limitations. To improve accuracy of extracting information within Twitter texts, the process of calibrating misspelled words shall be taken in advance. In conventional studies to correct the misspelled words of Twitter texts, the relationship between misspelled words and correcting words was resolved by concerning the dependency of co-occurrence words with misspelled words within sentences and morphophonemic similarity, but since the frequency of co-occurrence words of misspelled words is not concerned, it has not resolved to correct misspelled words completely. In this paper, to correct misspelled words in Twitter texts, the use of the character n-gram method concerning spelling information and the word n-gram method concerning frequency of co-occurrence words are to be proposed.

Keywords: Twitter text · Misspelled word · Correcting misspelled words · Character n-gram · Word n-gram

1 Introduction

SNS has become more popularized due to the rapid growth of the use of devices like tablet that is accessible to the Internet. SNS users post their profiles and contents, share messages, pictures, and links, and maintain social relationship through these activities [1]. Among SNS, "Twitter" is one of the most widely used micro blogs. Users transmit a short message of 140 characters called Tweet to share personal opinions and information and follow other users to receive their Tweets [2]. As many users of Twitter may share tweets, it is very influential to propagate information [3]. The emergency landing of passenger flight on Hudson River of New York in 2009 could be taken care of quickly as one of Twitter users announced the accident through Twitter. The texts by Twitter user at the accident were spread faster than conventional news. From the terror of Boston Marathon in 2013, the influence of Tweet could be found. After the terror, Twitter users sold various souvenirs and T-shirts to help victims and hash-tagged "Boston Strong" on their tweets. As this hash-tag "Boston Strong" was spread widely, this became the slogan

J.J. Jung and P. Kim (Eds.): BDTA 2016, LNICST 194, pp. 83–90, 2017.
DOI: 10.1007/978-3-319-58967-1_10

of strong spirit of Boston against adversity. Furthermore, in the terror of France in 2015, the Twitter text information was used. After the terror in France, a group of Twitter accounts tweeted the phrase of welcoming the attack and also tweeted the phrase of warning about more terrors in the future. Through the tweets by terror group, the group of terror could be predicted. As shown above, the Twitter user may recognize the requested information from the bulk Twitter texts, but in case of machine, to analyze the bulk text information, it is required to have a learning process about the characteristics of documents. Yet, in case of short text like Twitter, since the lexical variants are included to compose sentences to hold the information within the limited number of characters, it is limited to extract the information with the conventional method. Therefore, this study is to propose the solution using character n-gram method for spelling information to correct misspelled words and word n-gram method for frequency of co-occurrence words in Twitter. The paper is composed as follows. In the Sect. 2, the relative studies are explained. In the Sect. 3, the basis of selecting Twitter as the subject of correction of misspelled words is stated. In the Sect. 3.2, the overall system composition and the correcting of misspelled words and evaluation using character n-gram, and word n-gram are stated. In the Sect. 4, the conclusion and the future study are stated to conclude the paper.

2 Related Work

Richard Beaufort [6] has proposed the normalization by sharing the similarity of spelling check method and machine translation method. The normalization of system was based on the training model using word phrases. The validity of the method for French was checked by using the 10-fold cross verification. Choudhury [7] studied about the characteristics of natural language and texts that can be summarized and established the word level model. The Hidden Markov Model was composed to determine the type of word deformed from a normal word and the probability of deforming. The structure of Hidden Markov Model may find the words of types that could not be seen before through the linguistic analysis of SMS data. The variables of Hidden Markov Model may estimate the word arrangement of SMS texts and Standard English parallel phrases through the lesson of machine. Hassan [10] proposes the text normalization system of social media for social media text. Upon the unlabeled text phrase, the n-gram sequence is composed to use the random work similar to the context. By using this method by Hassan, there is no limit in domain and language, and in processing the social media text and in preprocessing stage of NLP application program, it is useful. Kobus [11] proposed the method of automatic normalization by using the uniqueness of text written by a machine like e-mail, blog, and chatting. For French SMS messages, it has normalized spelling by using another method that is not used for the automatic voice recognition device. Liu [12] proposed the method of normalizing the non-standard lexical and many abbreviations used in SMS and Twitter. The approach of character-level block as a divided word was proposed, and it has been combined with the conventional method.

3 Correcting Method and Test Evaluation of Misspelled Word of Tweet Text

This chapter proposes the correction system of misspelled words using character n-gram and word n-gram for misspelled words, the limitation in extraction of information from Twitter. Figure 1 is an overall diagram of system to extract and correct misspelled words from Twitter text.

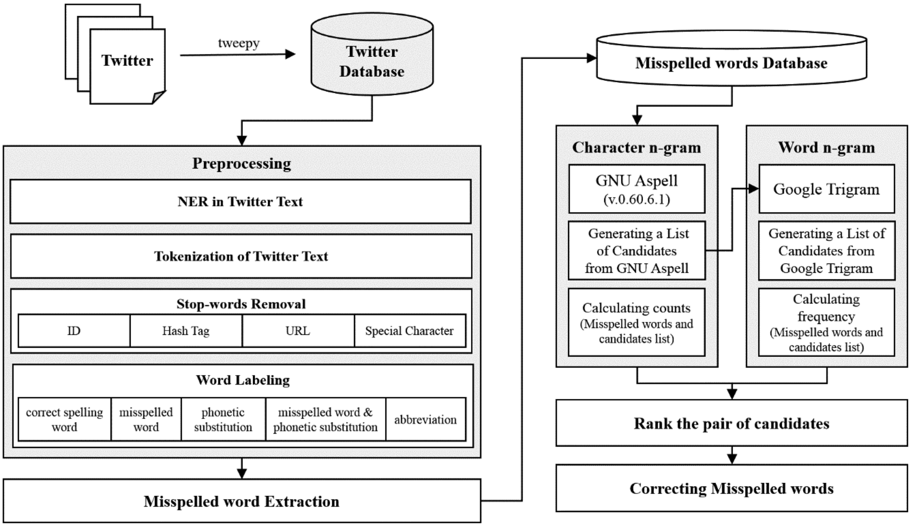
Twitter
tweepy
Twitter Database
Preprocessing
NER in Twitter Text
Tokenization of Twitter Text
Stop-words Removal
ID
Hash Tag
URL
Special Character
Word Labeling
correct spelling word
misspelled word
phonetic substitution
misspelled word & phonetic substitution
abbreviation
Misspelled word Extraction
Misspelled words Database
Character n-gram
GNU Aspell (v.0.60.6.1)
Generating a List of Candidates from GNU Aspell
Calculating counts (Misspelled words and candidates list)
Word n-gram
Google Trigram
Generating a List of Candidates from Google Trigram
Calculating frequency (Misspelled words and candidates list)
Rank the pair of candidates
Correcting Misspelled words

Fig. 1. Overall diagram of system to extract and correct misspelled words from Twitter text.

3.1 Extraction Method of Misspelled Words Within Twitter Text

To collect Twitter texts, the tweepy, the python openAPI, is used. The collected Twitter text is composed of various languages. Since this paper is about the misspelled words from Twitter texts in English, only the Twitter texts in English were extracted to be established as the database of Twitter. The texts stored in the Twitter database goes through the pre-process step. The pre-process step is conducted as the following order: 1. NER in Twitter Text, 2. Tokenization of Twitter Text, 3. Stop-words Removal, 4. Word Labeling. In a process of NER (Named Entity Recognition) in Twitter Text, the Named Entity of organization, location, and person within Twitter texts are NER. Since NER are unnecessary words for correction of misspelled words, they are removed from Twitter texts [17]. The NER used for Twitter texts was the Named Entity Recognition and the Standford NER Software developed at Stanford University. Table 1 shows the example of recognizing the Named Entity from Twitter texts.

Table 1. Example of recognizing the Named Entity from Twitter texts.

Twitter	
1	Hey/O @zaynmalik/O I/O just/O recieved/O my/O order/O fulfillment/O for/O MindOfMine/O thanks/O brother/O
2	I/O added/O a/O video/O 2/O a/O @YouTube/O laylist/Ohttp://youtu.be/2gGYWCLwYOI?a/O 50/O Cent/O x/O **Chris/PERSON** Brown/PERSON -/O I/O am/O The/O Man/O -LRB-/O Live/O in/O **Oakland/LOCATION**-RRB-/O
3	Just/O to/O refresh/O **Samantha/PERSON**,/O the/O planes/O hit/O the/O twin/O towers/O and/O **Pentagon/ORGANIZATION** b4/O the/O military/O action/O./O./O #auspol/O #lookitup/O
4	#DLL/O day/O 2tmrw/O!/O Planning/O time/O for/O the/O marketplace/O among/O other/O fun/O things/O./O CU/O soon/O @CampbellMira/O @MrNgoTDSB/O @teaching24seven/O @ZeliaMCT/O

Tokenization of Twitter Text process separates the Twitter texts based on the space among them. After that, The Twitter texts goes through the process of removing stopword that does not take a big role in showing contents of the context. In a Stopword removal process, the Stopword is removed in an order of ID, hashtag, URL, and special characters. To remove ID and hashtag, if the initial character begins with @, it is recognized as ID to be removed, and if the initial character is #, it is recognized as hashtag to be removed. In case of URL, if the initial character string begins with http, youtu, bit, or others, it is recognized as URL to be removed. At last, the special characters !, $, %, or others are removed. Table 2 shows the results of stopword removal from twitter texts.

Table 2. Results of stopword removal from twitter texts.

Twitter	
1	Hey, I, just, recieved, my, order, fulfillment, for, thanks, brother
2	I, added, a, video, 2, a, playlist, 50, Cent, x, I, am, The, Man, Live, in
3	Just, to, refresh, the, planes, hit, the, twin, towers, and, b4, the, military, action
4	day, 2tmrw, Planning, time, for, the, marketplace, among, other, fun, things, CU, soon

In the Word Labeling process, the Twitter text after removal through NER and tokenization are labeled into five types of correct spelling words (csw), misspelled words (mw), phonetic substitution (ps), misspelled words & phonetic substitution (mp), and abbreviation (ab). The correct spelling words and misspelled words were labeled by using the GNU Aspell dictionary (v6.06), and the chat word dictionary was used to label the phonetic substitution, misspelled words & phonetic substitution, and abbreviation. Table 3 shows an example of word labeling using the chat word dictionary and the GNU Aspell dictionary.

To correct the misspelled words in the Twitter texts, the words labeled as mw among 5 types of labeling (csw, mw, ps, mp, and ab) are extracted. In this study, the misspelled words are corrected by using character n-gram and word n-gram methods

Table 3. Example of word labeling using the chat word dictionary and the GNU Aspell dictionary.

Twitter	
1	(Hey, csw), (I, csw), (just, csw), **(recieved, mw)**, (my, csw), (order, csw), (fulfillment, csw), (for, csw), (thanks, csw), (brother, csw)
2	(I, csw), (added, csw), (a, csw), (video, csw), **(2, ps)**, (a, csw), (playlist, csw), **(50, ps)**, (Cent, csw), (x, csw), (I, csw), (am, csw), (The, csw), (Man, csw), (Live, csw), (in, csw)
3	(Just, csw), (to, csw), (refresh, csw), (the, csw), (planes, csw), (hit, csw), (the, csw), (twin, csw), (towers, csw), (and, csw), **(b4, mp)**, (the, csw), (military, csw), (action, csw)
4	(day, csw), **(2tmrw, ab)**, (Planning, csw), (time, csw), (for, csw), (the, csw), (marketplace, csw), (among, csw), (other, csw), (fun, csw), (things, csw), **(CU, ps)**, (soon, csw)

3.2 Method of Generating Word Pairs to Correct Misspelled Word

In this section, the process of generating a word pair of misspelled word and correct spelling word is be stated. In a process of character n-gram, the misspelled words and the GNU Aspell dictionary are used to generate the list of candidates. To evaluate the similarity in form between misspelled word and candidates, the LCS (Longest common Subsequence) method is used. The LCS algorithm finds the longest common subsequence from two strings of characters. Here, the partial string is different from a substring. The partial string refers to the derived string that may erase some characters but does not change the order. In other words, the partial character string shall be continuous, but the partial string does not have to be continuous. The partial character string of phrase is always a partial string, but the partial string does not have to be partial character string at all-time [18]. In this paper, the length of LCS between misspelled word and candidates are measured. For example, the longest common partial character

Table 4. Length of LCS and the frequency of 3-gram.

Misspelled word	Candidate word	LCS length	3-gram
Recieved	Received	7	223,448
	Relieved	7	55
	Receives	6	0
	Receive	6	3,392
	Revived	6	150
	Receiver	6	0
	Reserved	6	886
	Deceived	6	0
	Receded	6	0
	Recited	6	203
	Relived	6	116
	Perceived	6	80
	Recede	5	0
	Rived	5	0

string is “abegceb”, and thus the length of LCS is 7. In a process of word n-gram, the 3-gram of misspelled words is generated, and the frequency of google 3-gram is calculated. The 3-gram of misspelled word is composed of misspelled word and surrounding words. For example, in a sentence, “Hey I just recieved my order fulfillment for thanks brother.”, the 3-gram of misspelled word “recieved” can be generated as (I just *), (just * my), or (* my order). For the empty space (*) of 3-gram of misspelled word, the words of candidates composed through character n-gram are entered. The 3-gram of misspelled word is searched on the google 3-gram to calculate the frequency. To generate a word pair of misspelled word and correct spelling word, the length of LCS measured by character n-gram and the 3-gram frequency of misspelled word are used. Table 4 shows the length of LCS between misspelled word and candidates and the frequency of 3-gram of misspelled word.

However, the range of data value by the frequency of 3-gram of misspelled word and the length of LCS are different from each other. Therefore, there shall be a process of normalization to make the ranges of two data to be the same. The Eq. 1 is used for normalization of the length of LCS, and the Eq. 2 is used for normalization of the frequency of 3-gram of misspelled word.

$$NLCS_{length} = \frac{LCS_{length}}{MaxLCS_{length}} \tag{1}$$

$$NF_{3gram} = \frac{f(trigram, googletrigram)}{Maxf(trigram, googletrigram)} \tag{2}$$

At last, to correct misspelled words, a pair of word composed of misspelled word and correcting spelling word is made. To select the correct spelling word of misspelled word, the maximum value of candidates is measured. To measure this maximum value, the sum of length of normalized LCS and frequency of 3-gram of normalized misspelled word is used. The Eq. 3 is used to calculate the sum of length of normalized LCS and frequency of 3-gram of normalized misspelled word.

$$NLCS_{length} + NF_{3gram} \tag{3}$$

Among candidates of misspelled words, the candidate with the maximum value is selected for a pair of misspelled word and correct spelling word. The misspelled word within the Twitter text is substituted with the correcting spelling word of pair of misspelled word under the appearing order of misspelled word.

4 Conclusion

In this paper, the method of correcting misspelled words within the Twitter texts is proposed. The Twitter texts are collected by using tweepy, or python openAPI. The extraction of misspelled words within Twitter texts are done through NER in Twitter Text, Tokenization of Twitter Text, Stop-words Removal, and Word Labeling. A pair of misspelled words extracted is generated by using character n-gram and word n-gram methods. In a character n-gram method, the list of candidates is generated, and the length of LCS is measured. The list of candidates is generated by using the GNU Aspell dictionary. The length of LCS is measured by using the LCS algorithm. In a word n-gram method, the 3-gram is generated with misspelled words and surrounding words, and the frequency of 3-gram is measured. Two words on the left and two words on the right with misspelled words within the Twitter texts are used to be generated. The frequency of 3-gram is measured by searching the google 3-gram. Since the LCS length of candidates measured in character n-gram and word n-gram methods and the frequency of 3-gram have different ranges from each other, and thus they are normalized to be summed up. Among candidates, the word with the maximum value is selected as the correct spelling word to be paired up with a misspelled word. The misspelled words of Twitter texts are corrected by substituting the correct spelling word of pair of misspelled word generated by the method proposed in this paper. As a result of correcting misspelled words within the Twitter texts, the method proposed in this paper is effective.

Acknowledgments. This research was supported by the Human Resource Training Program for Regional Innovation and Creativity through the Ministry of Education and National Research Foundation of Korea (NRF-2014H1C1A1073115) and This research was supported by SW Master's course of hiring contract Program grant funded by the Ministry of Science, ICT and Future Planning (H0116-16-1013).

References

1. Wilson, C., Boe, B., Sala, A., Puttaswamy, K.P.N., Zhao, B.Y.: User intereactions in social networks and their implications. In: Proceedings of the 4th ACM European Conference on Compter Systems, pp. 205–218 (2009)
2. Kim, J., Ko, B., Jeong, H., Kim, P.: A method for extracting topics in news twitter. Int. J. Softw. Eng. Appl. **7**(2), 1–6 (2013)
3. Vespignani, A.: Modelling dynamical processes in complex socio-technical systems. Nat. Phys. **8**, 32–39 (2012)
4. Beaufort, R., Roekhaut, S., Cougnon, L.A., Fairon, C.: A hybrid rule/model-based finite-state framework for normalizing SMS messages. In: Proceedings of the 48th Annual Meeting of the ACL (ACL 2010), pp. 770–779 (2010)
5. Choudhury, M., Saraf, R., Jain, V., Mukherjee, A., Sarkar, S., Basu, A.: Investigation and modeling of the structure of texting language. Int. J. Doc. Anal. Recogn. **10**(3), 157–174 (2007)
6. Hassan, H., Menezes, A.: Social text normalization using contextual graph random walks. In: The 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013), pp. 1577–1586 (2013)
7. Kobus, C., Yvon, F., Damnati, G.: Normalizing SMS: are two metaphors better than one? In: The 22nd International Conference on Computational Linguistics (COLING 2008), pp. 441–448 (2008)
8. Chen, Y.: Improving text normalization using character-blocks based models and system combination. In: The 24th International Conference on Computational Linguistics (COLING 2012), pp. 1587–1602 (2012)

9. Jung, J.J.: Online named entity recognition method for microtexts in social networking services: a case study of twitter. Expert Syst. Appl. **39**(9), 8066–8070 (2012)
10. Longest common subsequence problem. http://en.wikipedia.org/wiki/Longest_common_subsequence_problem