Journal of the Korean Society of Surveying, Geodesy, Photogrammetry and Cartography
Vol. 37, No. 2, 81-89, 2019
https://doi.org/10.7848/ksgpc.2019.37.2.81

ISSN 1598-4850(Print)
ISSN 2288-260X(Online)
Original article

Developing an Alias Management Method based on Word Similarity Measurement for POI Application

Choi, Jihye [1)] · Lee, Jiyeong [2)]

Abstract

As the need for the integration of administrative datasets and address information increases, there is also growing interest in POI (Point of Interest) data as a source of location information across applications and platforms. The purpose of this study is to develop an alias database management method for efficient POI searching, based on POI data representing position. First, we determine the attributes of POI alias data as it is used variously by individual users. When classifying aliases of POIs, we excluded POIs in which the typo and names are all in English alphabet. The attributes of POI aliases are classified into four categories, and each category is reclassified into three classes according to the strength of the attributes. We then define the quality of POI aliases classified in this study through experiments. Based on the four attributes of POI defined in this study, we developed a method of managing one POI alias through and integrated method composed of word embedding and a similarity measurement. Experimental results of the proposed POI alias management method show that it is possible to utilize the algorithm developed in this study if there are small numbers of aliases in each POI with appropriate POI attributes defined in this study.

Keywords : POI Alias Database Management, Point of Interest, Word Embedding, Similarity Measurement, Quality of POI Alias

1. Introduction

The need and demand for the integration and linkage of administrative data based on recent addresses is increasing. Having spatial components, geocoding that transforms sub-spatial data into spatial data and address data contained in administrative data can be used to provide location information (Lee, 2009). In addition, location-based services of administrative data can be possible through integration with Information Technology techniques. The purpose of this study is to develop a POI (Point of Interest) alias database management method for efficient search through POI data, in Korean, which may be used for address representation and word embedding and similarity measurement technique used for text analyses.

POI is mainly data that represents places where people are interested or useful. The data model of POI data is being studied by the OGC (Open Geospatial Consortium) for international standardization (OGC, 2013), and studies are being conducted for standardization and utilization of POI data model in Korea. The KICTA (Korea Information and Communications Technology Association) has established the POI data model (TTA, 2014) that describes the geometric information, feature attributes, and the relationship between

Received 2019. 03. 20, Revised 2019. 04. 09, Accepted 2019. 04. 25
1) Member, Dept. of Geoinformatics, University of Seoul (E-mail: jihye30@uos.ac.kr)
2) Corresponding Author, Member, Dept. of Geoinformatics, University of Seoul (E-mail: jlee@uos.ac.kr)

each item for POI construction and design. It serves three main purposes: object retrieval in location-based services, object representation on a background map and to describe objects in space and time (OGC, 2013; Park *et al.*, 2016). In location-based services, which is the primary purpose of a POI, a POI alias is generated when searching for an object. These aliases can be very diverse and wide-ranging depending on the search attributes of users and typo. For example, 'Seoulsilibdaehaggyo (University of Seoul)' can exist in various ways such as 'Seoulsilibdae (U. of Seoul)', 'Silibdae (U. of S)', 'Silibdaehaggyo (Univ. of Seoul)' and so on. Another example may be when searching for a pub named 'Meogjagolmog (Food Street)' in a POI portal. If the shortened term 'Meoggol (Food St)' is used as a search token, results not matching 'Nonhyeon-dong Meogjagolmog (Nonhyeon-dong Food Street)' will be displayed. If the user specifies its relative location, say, 'Nonhyeon-dong' and search for 'Nonhyeon-dong Meoggol (Nonhyeon-dong Food St)', results matching that keyword yet still be the wrong location that the user is looking for, or less results may be displayed since the search term cannot be found on the map.

To overcome these situations, this study develops a method to manage POI aliases efficiently to improve the accuracy of POI search. In addition, we shall summarize the attributes of POI aliases and define quality for each property to aid in developing POI search and alias management methods. In the second section, we discuss previous works related to this study. We then define considerations for a POI searching and alias database management, and the proposed methodology. The next section demonstrates this through an experimental implementation, and we discuss our conclusions and recommendations for future studies.

2. Related Researches

Hard-matching techniques for POI searching are used in stand-alone systems such as navigation. The hard-matching technique is a method that searches only for an exact match in the alias table, not considering the similarity between the user's query and the words existing in the alias table.

The set-based POI search method based on applying a set concept has been developed for a stand-alone system that can handle massive data (Ko and Lee, 2013). An index is generated by rearranging each letter appearing in the POI data in the database in ascending order, and then, the reverse index is created again. The generated reverse index text data is subjected to a text search process to extract POI data most suitable for a user query. Currently, the set-based POI search technique has been extended to support duplicate letters (Ko and Lee, 2013).

Xu *et al.* (2012) proposed an address matching engine for location data that do not contain spatial coordinates, such as address, phone number, and zip code. Based on the word segmentation technique, an engine that matches the natural language was proposed. This engine can convert non-spatial data into coordinates. Sasaki *et al.* (2018) (2018) also proposed an algorithm implementing a sequenced route query based on semantic similarity scores of the POIs in the route.

Kim *et al.* (2009) studied the method of matching POI information with a road name address map to construct the spatial information platform. The data set used in the experiment was the building object of the road name address map, the POI of the car navigation system, and the POI obtained through the POI search API. Experimental results are shown in Fig. 1, which is the best match between CNS (Car Navigation System) POI matching and Web POI matching. In the case of semantic matching, where a matching table is created for a pair of objects having the highest similarity obtained from by analyzing the similarity of the search keyword to the name of the object of comparison, it can be seen that the CNS POI matching result is relatively high.

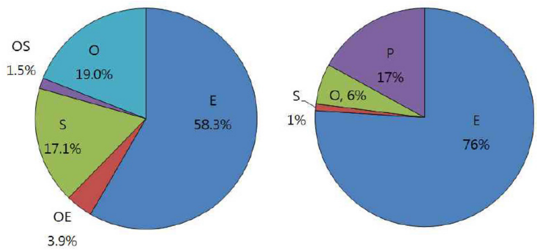

Fig. 1. Matching result of CNS POI (left) and Web POI (right) (Kim *et al.*, 2009)

3. Considerations for managing POI discovery and alias database

In Section 3, we consider the considerations for efficient alias database management proposed in this study. We investigate the attributes of the model and algorithm of word embedding and similarity measurement.

3.1 Word Embedding Overview

Word embedding refers to the conversion of text data into vector data. Recently used word embedding algorithms based on the concept of distributed representation have implications within the meaning of words, so that the distance between similar words is close, and the distance between non-similar words can be distant (Harris, 1954; Elman, 1991; Glenberg and Robertson, 2000).

Word2Vec is a simple neural network structure developed by Google in 2013 with one hidden layer. The input data is a Text Corpus and the output data is a word vector. First, the vocabulary of Word2Vec model is constructed using training text data and learn vector expression of word. The simplest way to identify a learned expression is to use a tool that finds the closest word from a particular word (Google, 2013).

Word2Vec has two architectures as illustrated in Fig. 2, namely Skip-gram and CBOW (Continuous Bag-of-Words) (Mikolov *et al.*, 2013a, 2013b). Skip-gram is a method of predicting neighboring words in the center of a word. It predicts a space in a sentence such as "___ is thicker than ___" (i.e., "*blood* ... *water*") and constructs a vector including the relation with neighboring words through the center word. CBOW allows you to predict the blanks in the sentence "Like ____, like son." (i.e., "*father*") as a way of predicting the center word through the surrounding words.

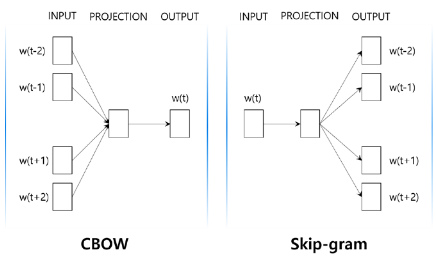

Fig. 2. Architecture of CBOW and Skip-gram model (Mikolov *et al.*, 2013a)

Fig. 3 illustrates when Skip-gram is applied among the two models of Word2Vec to POI and POI Alias data. There are five input data ['Word A', 'Word B', 'Word C', 'Word D', 'Word E'] and a window size of 2.

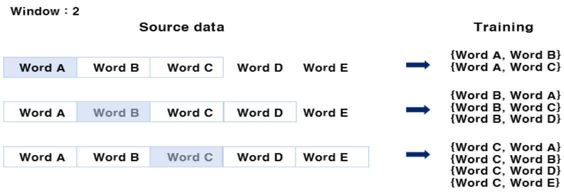

Fig. 3. Illustration of the Word2Vec Skip-gram model learning

Since the skip-gram is a method of predicting surrounding words with a center word, the first word 'Word A' is the first center word. When the center word is determined, a filter is generated before and after the center word by the window size. In Fig. 3, because the window size is 2, two words before and 2 words after the center word should be set as the surrounding words of 'Word A'. However, since there is no word to the left of 'Word A', you will learn only about the words ('Word B', 'Word C'). At this time, 'Word B' and 'Word C' are learned with respect to the center word.

The next step is to move the center word to the right and learn the surrounding words. To do that, set the next word, 'Word B', as the center word. 'Word A' becomes the surrounding word on the left, and 'Word C' and 'Word D' becomes the surrounding word on the right. In this way, all the words in the training data are learned by sliding according to the window size.

3.2 Overview of similarity measurement

In the field of text data and NLP (natural language processing), a similarity measure refers to the degree of likeness between text data. Similarity measure methods include the use of cosine similarity, Jacquard Coefficient, Eucledian Distance, and Tanimotto Coefficient (Gomaa and Fahmy, 2013). In this study, we use text matching algorithm based on Gestalt approach. Algorithms based on the Gestalt approach focus on pattern matching. Therefore, the algorithm is usually implemented in search of spelling errors and reading errors when used in text datasets or in database searches. The algorithm based on Gestalt approach handles all whitespace and special characters, so that the result of similarity measurement between two words is more than 60%, two words are defined as similar (Ratcliff and Metzener, 1988).

The Gestalt approach-based algorithm is based on Eq. (1). A comparison is made for the measurement of similarity between words. N_a is the total number of spellings of the word a, N_b is the total number of spellings of the word b, and N_c is the sum of the spellings matched between the word a and the word b. First, the most common parts between two words are grouped, and the left and right words of the remaining two words are compared, and the similarity scores are obtained by the same number of spellings. At the end, the degree of similarity between two words is determined by measuring the degree of similarity using the number of spellings ($N_a + N_b$) of two words and the number of spellings (N_c) overlapping between two words.

$$S = \frac{N_c}{Na + N_b} \tag{1}$$

4. Proposed POI alias database management method for POI searching

In this section, we discuss the attributes of POI aliases, how to search POIs, and how to efficiently manage POI alias databases based on the above-mentioned considerations. In particular, we define attributes of POI and its quality before development of POI search and alias database management method.

4.1 Design for qualified POI aliases

Before searching for POIs and managing the alias database, we first define the quality and attributes of POI aliases. The quality of the POI alias in this study is the most important because it dictates if it can be added to the alias database, which in turn is added to various POI names with the same attributes. The data used in this study is based on the POI data produced by the NGII (National Geographic Information Institute) and various kinds of alias data were needed in addition to this data. We define the properties of the POI alias to create additional POI aliases. To accomplish this, 20 official POI names are transmitted to 5 ordinary people who use object search by using location-based services, and in total 100 POI alias data are collected based on their POI object search activity. The official name of the POI is based on the POI data produced by the NGII. The collected POI aliases were analyzed and classified into four attributes as shown in Table 1, and the experimental data to be used in this

Table 1. Four attributes of POI alias

Attribute	class	Attributes of aliases
Attribute 1	1	Case where only one special character is removed from POI official name
	2	Case where 2 special characters are removed from the POI official name
	3	Case where 3 or more special characters are removed from POI official name
Attribute 2	1	Case where the degree of shortening of POI official name is 1 Ex) Seoulsilibdaehaggyo (University of Seoul) : Seoulsilibdaehag (Univ. of Seoul)
	2	Case where the degree of shortening of POI official name is 2 Ex) Seoulsilibdaehaggyo (University of Seoul) : 'Seoulsilibdae (U. of Seoul)'
	3	Case where the degree of shortening of POI official name is 3 or more Ex) Seoulsilibdaehaggyo (University of Seoul): Silibdae (U. of S)
Attribute 3	1	Change the order of the POI names (the length of the changed words is the same)
	2	Change the order of the POI names (the length of the first word in the modified word is long)
	3	Change the order of the POI names (the length of the word after the changed word is long)
Attribute 4	1	Case using one space in POI official name
	2	Case using two space in POI official name
	3	Case using three or more spaces in the POI name

study were grouped and reconstructed. The strength of each attribute was subdivided into three grades for each of the four attributes (Table 1).

The first attribute of the POI alias is to exclude special characters from the official name of the POI. For example, if the official name of the POI is 'Jji&Kko', it means to mark it except for the special character '&'. The second POI alias property is defined as a shortening of the formal name of the POI, for example you can abbreviate 'Ediyacoffee (Ediya Coffee Shop) ' or 'Hoegiyeogjeom (Hoegi Station-Branch)' when 'Ediyacoffee(Hoegiyeogjeom)' is the official name. The third attribute is to change the order of business name and location in the official name of POI. 'Ediyacoffee(Hoegiyeogjeom)' can be represented as '(Hoegiyeogjeom)Ediyacoffee'. The fourth attribute is to use spacing. 'Ediyacoffee(Hoegiyeogjeom)' can be variously expressed as 'Ediya coffee (Hoegiyeogjeom)'. In addition, 'Ediyacoffee(Hoegiyeogjeom)' alias 'Ediyacoffee' may mean extending the meaning of the formal name of the POI was found, but it is not consistent with the POI alias in the algorithm of this study, so we decided to exclude.

The mean and standard deviation of similarity measures between POI canonical names and their aliases were calculated using the official name of the POI (Table 2). Attributes 1 and 4 have little difference in the similarity measurement results between data, and the average of the two attributes exceeds about 80%. If we look at the quality of an alias as being determined by how it can improve the alias database's coping with unexpected POI names, it can be said that these attributes contribute little to the alias quality. Conversely, attribute 2 has a large standard deviation, and attribute 3 has the lowest average, so these attributes contribute largely to the alias quality.

Table 2. Mean and standard deviation of Similarity Measures, according to attributes

	Mean	Standard Deviation
Attribute 1	89.665	3.43
Attribute 2	71.017	15.46
Attribute 3	53.423	8.26
Attribute 4	83.429	3.40

4.2 Proposed POI searching method

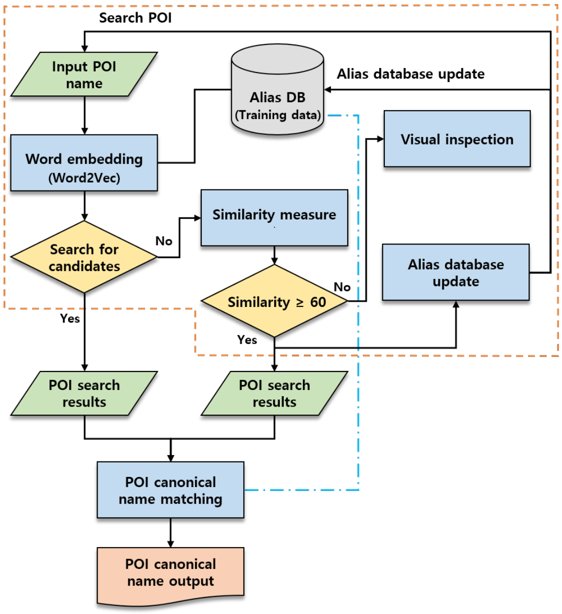

Fig. 4. Flowchart of POI search and alias database management algorithm

The overall algorithm flow diagram developed in this study for the accurate search of POIs and efficient alias database management is shown in Fig. 4, and the pseudocode that implements this is shown in Table 3. The POI search method developed in this study consists of two major parts: POI retrieval through word embedding and POI retrieval through similarity measurement.

Table 3. Pseudocode for POI searches and alias databases management

FindPOI&UpdateAliasDB	
in	*Search POI*
out	*POI formal name*

```
SearchPOI, outputPOI : String
s, i, c : integer
allalias, candidate, similarity : list
aliasTable : 2d list
dict : dictionary
count = Countor
W2V : Word embedding model (Word2Vec)
ST : Similarity measure
ws : window size
vs : vector size
epochs : epochs

Begin
  import W2V
  tuning model of W2V using aliasTable, ws, vx
  train model of W2V using aliasTable, epochs
  Input SearchPOI
  IF (find candidate using W2V) not Null
    fine and match candidate using dict
    IF candidate not Null
      FOR c = 0 to length(candidate)
        count = counting same value, candidata[c]
      candidate = count DESC
      outputPOI = candidate[0]

  ELSE
    FOR s = 0 to length(allalias)
      similarity += ST between SearchPOI and allalias
    FOR i = 0 to length(similarity)
      IF similarity[i] >= 60
        candidate += similarity[i]
    candidate DESC
    outputPOI = candidate[0]
    IF outputPOI not Null
    outputPOI = fine and match outputPOI using dict
    update dict and allalias, aliasTable
    train model of W2V using aliasTable, epochs
    find candidate about SearchPOI using W2V
    fine and match candidate using dict
    IF candidate not Null
      FOR c = 0 to length(candidate)
        count = counting same value, candidata[c]
      candidate = count DESC
      outputPOI = candidate[0]

  End
End
```

4.3 Proposed POI alias database management method

POI aliases that do not exist in the POI alias database are not included in the training data of Word2Vec, which is the word embedding method used in this study, so that the POI search is performed through the similarity measurement algorithm. When an accurate result is obtained through the similarity measurement algorithm, the POI alias database is updated, and the Word2Vec model is re-trained and re-searched. This allows searching of POI aliases that are not trained in the Word2Vec model and are not existing in the existing alias database, and for the updating of the POI alias database.

The orange box in Fig. 4 is a flowchart of the algorithm for managing the POI alias database. First, if the name of the POI received as the input value through the search algorithm fails to search the appropriate candidate group using the Word2Vec model, the official name of the POI is searched through the similarity measurement method. Through the similarity measure between the input value and the alias database, the most similar word among the words having the similarity degree of 60% or more is derived and the user is confirmed whether the alias database is updated or not. If the degree of similarity is less than 60%, the user is requested to confirm the input text in the POI search. When it is confirmed that the input data of the POI name retrieved to the user matches the POI data derived from the result, the input value POI name is newly added to the alias database and the Word2Vec model is re-trained through the updated alias database based on the input value POI name. After the training is completed, a search is repeated for the POI nickname entered as the search value, and the official POI name is derived as the result.

5. Experimental Implementation for the Proposed Method

The experimental data used for this study include POIs existing in Seoul based obtained from NGII. The data used in the experiment is based on the assumption that the official names of POIs do not contain English names or any typographical errors. The algorithms for analyzing and experimenting on the POI search and alias database management methods are based on Python 3.6 programming language and Anaconda. The related data including POI official name and alias are exported as .txt (tab-separated) form of MS Excel We implemented POI search and alias database management method by loading POI data in the demo program. We also implemented this experiment using the genism and difflib Python libraries.

5.1 POI search and alias database management algorithms

This experiment evaluates and verifies the feasibility of the algorithm using the algorithm developed in this study for accurate POI search and efficient management of POI alias database. The data used in the experiment consisted of only 6,000 POI names in the alias database (training data). Experimental data consisted of 6,000 POI names and a total of 30,000 POI names including data from the alias database. A data subset containing 250 randomly generated names was produced.

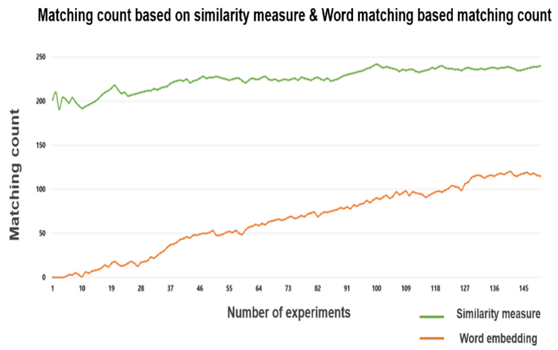

Fig. 5. Number of correct matches of POI search results through word embedding and similarity measurement according to the number of experiments

The green graph Fig. 5 shows the number of POIs found or updated in the alias database among the 250 experimental data based on the similarity measurement algorithm, and the orange graph shows the number among the same subset found through word embedding. Fig. 6 shows a steady rise in the graph showing the of POI match rate among the test subset.

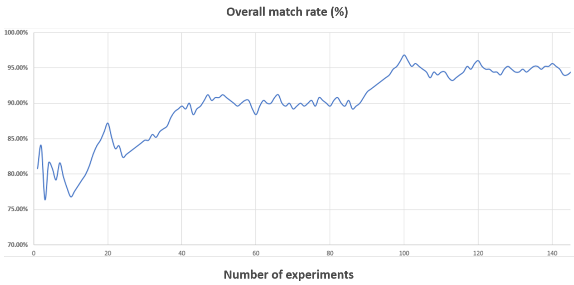

Fig. 6. Match rates of POI results according to numbers of experiment

In Word2Vec model of word embedding, the number of POI matching results was very low at the beginning of the experiment because the data at the first training was just 6,000 POIs. This is because the training data is not related to the surrounding words. However, it is evident that the performance of Word2Vec is getting better by gradually adding aliases to the alias database through similarity measure, and by re-training accordingly, related words are gradually formed around the POI official name. In the case of the similarity measurement based on the first experiment, 200 or more matches were matched, and the database and the Word2Vec model were retrained by adding an alias to the alias database. As a result, similarity measure can be confirmed that the number of matching increases as the number of experiments increases.

Based on the POI data assumption that excludes POI aliases for datasets with alphanumeric names and those with punctuation marks, if the POI and the four attributes corresponding to each POI are managed in the POI alias database through this experiment, algorithms for one POI search and alias database management can be utilized.

5.2 Attributes and quality of POI aliases

For this experiment, we consider that the algorithm developed on this research is applicable if the number of attributes that the alias satisfies is at least four, based on the similarity measurement algorithm. The test is run to determine the minimum number of alias databases that can obtain a match rate of about 95% or more.

The 100 POIs per attribute were randomly extracted from the dataset to generate the experimental data for this part, for a total of 400. In Fig. 7, the X-axis represent five data sets as follows. Dataset 1 is data with only 6,000 POI official name data, and Dataset 2 is a data set with 6000 aliases with attribute 3, which is written by changing the order of the words before and after the data set 1. Dataset 3 is a dataset with Dataset 2 plus 6000 aliases with property 2 with the property to abbreviate the POI. Dataset 4 is a Dataset 3 added with an Attribute 4 alias including spacing, and Dataset 5 is a data set having Attribute 1, created by removing special character in Dataset 4.

As a result of experiment based on POI official name and similarity measurement algorithm without any aliases, 73% of the total 400 data were obtained as proper POI name. As a result of defining the quality of the POI Attribute, Attribute 3 which changes the order of the POI regular name is added to the alias database first, and 400 experimental data are retested based on the similarity measurement algorithm. As with the first data set, 73% POI matching was done when only the POI formal name existed, but 92% matching rate was shown by adding the changed data of POI formal name. Next, by adding an alias with Attribute 2 to abbreviate the POI name, 400 experimental data were re-tested based on the similarity measurement algorithm, and the matching rate was 98%. After adding the Attribute 4 including the spacing and the Attribute 1 data for removing the special character, the matching rate was 99% and 100%, respectively.

From these results, we can see the quality of the POI quality is defined primarily by Attributes 3 and Attribute 2, which are described in Table 1, as these increase the number of matches significantly from a 73% to 98% match rate, followed by Attribute 4 and Attribute 1, which further increases the match rate from 98% to 100%.

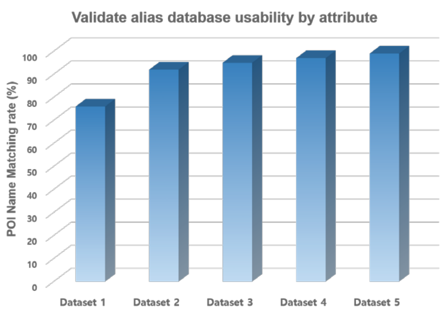

Fig. 7. Validate alias database usability by attribute

6. Conclusion

In this study, we developed an accurate POI search method and an efficient POI alias database management method based on word embedding and similarity measurement. We classify the attributes of the POI aliases into four classes. These classes involve removing a special character, shortening or abbreviating, changing word ordering and to add excess space characters, respectively. Each attribute was further subdivided into three classes depending on each respective characteristic, 1 as the weakest and 3 as the strongest.

In this study, POI search and efficient POI alias database management method based on POI alias attributes have been developed. In the case of POI search, the POI name is converted into vector data through word embedding, and the alias data is trained based on the Word2Vec model so that similar aliases are located close to each other. Based on this, we developed and implemented a method of retrieving and deriving the formal name of POI. In the case of POI alias database management, when a user attempts to search by a different name than the POI name trained through the Word2Vec model, the alias matching the input value is searched through the Gestalt-based matching algorithm. We developed and implemented a method to update the input POI name to the alias database and retrain the Word2Vec model, in case the user inputs a non-official name for the POI.

In this study, POI data used in the experiment do not contain English names, except for POI alias data, and in which typographical errors were eliminated, and we investigate the applicability of the algorithm developed in this study and the quality of POI aliases despite these assumptions. If the small numbers of aliases have Attribute 3 and Attribute 2, which are described in Table 1, we can achieve a POI search matching rate of 95% or more. In other words, if the POI alias database manages the POI alias name and Attributes 3 and 2, which are appropriate for each POI, even if a POI alias that does not exist in the alias database is input as search value, the POI is searched through similarity measurement and added to the POI alias database.

As a limitation of this study, similarity cannot be detected for words that do not exist in the training data of the model when Word2Vec model used in word embedding is used. It is necessary to study how to efficiently manage POI search and alias database using only machine learning or deep learning model.

Acknowledgements

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2017R1D1A1B03028890).

References

Elman, J.L. (1991), Distributed representations, simple recurrent networks, and grammatical structure, *Machine Learning*, Vol. 7, No. 2-3, pp. 195-225.

Glenberg, A.M. and Robertson, D.A. (2000), Symbol grounding and meaning: a comparison of high-dimensional and embodied theories of meaning, *Journal Of Memory And Language*, Vol. 43, No. 3, pp. 379-401.

Gomaa, W.H. and Fahmy, A.A. (2013), A survey of text similarity approaches, *International Journal of Computer Applications*, Vol. 68, No. 13, pp. 13-18.

Google. (2013), Word2vec, https://code.google.com/archive/p/word2vec/ (last date accessed : 14 January 2019).

Harris, Z.S. (1954), Distributional structure. *Word*, Vol. 10, No. 2-3, pp. 146-162.

Kim, J.O., Huh, Y., Lee, W.H. and Yu, K.Y. (2009), Matching method of digital map and POI for geospatial web platform, *Journal of Korean Society for Geospatial Information System*, Vol. 17, No. 4, pp. 23-29.

Ko, E.B. and Lee, J.W. (2013), Implementation of a set-based POI search algorithm supporting classifying duplicate characters, *Journal of Digital Contents Society*, Vol. 14, No. 4, pp. 463-469. (in Korean with English abstract)

Lee, J. (2009), GIS-based geocoding methods for area-based addresses and 3D addresses in urban areas, *Environment and Planning B: Planning and Design*, Vol. 36, No. 1, pp. 86-106.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a), Efficient estimation of word representations in vector space, *arXiv preprint arXiv:*, pp. 1301.3781.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., and Dean, J. (2013b), Distributed representations of words and phrases and their compositionality, *In Advances In Neural Information Processing Systems* , pp. 3111-3119.

OGC(Open Geospatial Concortium). (2013), Points of interest (POI) Standards Working Group Charter, https://portal.opengeospatial.org/files/?artifact_id=54800 (last date accessed : 12 April 2019).

Park, J.H., Kang, H.Y., and Lee, J. (2016), A spatial-temporal POI data model for implementing location-based services, *Journal of the Korean Society of Surveying, Geodesy, Photogrammetry and Cartography*, Vol. 34, No. 6, pp. 609-618.

Ratcliff, J.W. and Metzener, D.E. (1988), Pattern-matching-the gestalt approach, *Dr Dobbs Journal*, Vol. 13, No. 7, pp. 46.

Sasaki, Y., Ishikawa, Y., Fujiwara, Y., and Onizuka, M. (2018), Sequenced Route Query with Semantic Heirarchy. *Proceedings of the 21st International Conference on Extending Database Techology*, 26-29 March 2019, Lisbon, Portugal, pp. 37-48.

TTA. (2014), POI (Point of Interest) data model, http://www.tta.or.kr/data/ttas_view.jsp?rn=1&by=desc&rn1=Y&standard_no=TTAK.OT-10.0360&order=publish_date&publish_date=%C2%A7ion_code%3D&nowpage=1&totalSu=1&pk_num=TTAK.OT-10.0360&nowSu=1 (last date accessed : 12 April 2019).

Xu, C., Li, Q., and Yong, W. (2012), The Design and Implementation of Address Matching Engine. *Proceedings of the International Conference on Geo-spatial Solutions for Emergency Management an the 50th Anniversary of Chinese Academy of Surveying and Mapping*, 14-16 September 2009, Beijing, China. ISPRS Archives Volume XXXVIII-7/C4, pp. 118-120.