Preprocessing Method Topic-based Path Model by Using Word2vec

Shoji Nohara and Ryosuke Saga

Abstract— **Studying purchasing factor for product developers in the market place is important. Using text data, such as comments from consumers, for factor analysis is a valid method. However, previous research show that generating a stable model for factor analysis using text data is difficult. We assume that if the target text data are handled well, then the analysis can progress smoothly. This study proposes pre-processing text data by word2vec for factor analysis. Word2vec regards words as vectors in text. Our proposed process is effective, because variables are expressed as the frequency of words in the analysis model. Experiment results also show that our proposed method is helpful in generating an analytical model.**

Index Terms— **Causal Analysis, Word2vec, Topic Model, Structural Equation Modeling.**

I. Introduction

PRODUCT developers in many companies gather customer opinions, especially focusing on text data from reviews or questionnaires. Developing a brand name or evaluating merchandise by using text data is beneficial. However, this approach cannot easily handle massive text data. In recent years, text mining has been used as an important method in market research when dealing with huge amounts of text data.

Topic models are a general method in the field of data-mining. Topic modelling is a machine learning technique that clarifies the structure of a document group by estimating words. The words constitute a topic based on the premise that each document group comprising the corpus belongs to that specific topic. Several studies have analyzed various consumer situations using topic models. For example, Kawanaka et al. proposed a method for analyzing competitive relations of brands using latent semantic analysis (LSA), which is a kind of modelling method [1]. Wajima et al. proposed the identification of a negative factor using latent Dirichlet allocation (LDA), which is effective for many applications [2]. These related studies indicate that factor analysis using topic modelling is an effective and valid approach in handling text corpus (i.e., sets of electronic documents). However, both LSA and LDA cannot define relationships among topics in an analysis model. Kunimoto et al. successfully analyzed the gaming software market by using structural equation modelling (SEM), which is a factor analysis method [3]. They proposed a path model generation process for SEM using hierarchical LDA (hLDA). Meanwhile, Saga et al. proposed using SEM with hLDA targeting Crowdfunding [4]. They also attempted to combine numeric and text data to explain how the identified relationships influence the decision to invest in a crowdfunding project [4]. Although these proposed methods employ visual and quantitative analyses, it remains difficult to generate a model. In addition, the significance level is not mentioned or not high enough to interpret in these works.

In the current study, we propose a pre-processing method that uses a novel technique, Word2vec, for pre-processing target text data. Word2vec is a two-layer neural network that, after obtaining information from the text corpus, outputs the feature vectors of words in the text corpus. Word2vec is also able to compute the similarity of words as a similarity of vectors, thus allowing it to group words based on similarities. By using this technique, we can extract words that are not keywords but are related to keywords. Results show that higher evaluation values (such as the score of GFI) can be gained by using the analysis model

II. Topic-based Path Model Constructed for SEM

SEM analyzes various relationships among several factors, i.e., latent and observed variables. A latent variable is an invisible concept that is used for target analysis. An observed variable is an observable item from the target analysis, and is used to estimate a latent variable. These variables have “causal” and “co-occurrence” relationships. SEM can quantify the influence and strength of these relationships.

A path model is used to understand the relationships among the variables. This model visualizes the factors and the relationships among them, as shown in Fig. 1. In the path model, the observed and latent variables are denoted by a rectangle and an ellipse, respectively. The relationships among the variables are expressed by the unidirectional and bidirectional arrows, which respectively correspond to causal

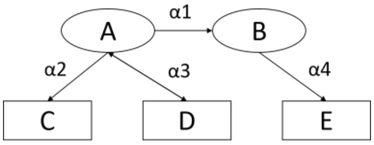

Fig. 1. Path model

This research was supported by KAKENHI 25240049.

The authors are with Computer Science and Intelligent Systems, Graduate School of Engineering, Osaka Prefecture University, 1-1 Gakuen-cho, Naka-ku, Skai-City, Osaka, Japan (e-mail: sv107029@edu.osakafu-u.ac.jp, saga@cs.osakafu-u.ac.jp).

and co-occurrence relationships.

The path model shown in Fig. 1 consists of three observed variables (C, D, and E) and two latent variables (A and B). The relationship between A and D, denoted as α3, is co-occurrence, whereas the other relationships, denoted as α1, α2, and α5, are causal relationships.

Two methods can be used to build the models: the data-driven approach (exploratory factor analysis, EFA) and the hypothesis-based approach (confirmatory factor analysis, CFA). Especially for the former methods, Saga et al. proposed a method that can build the model using text data based on an LSA-approach [5]. Furthermore, Saga et al. [6] and Kunimoto et al. [3] have developed the approach of using multinominal topics based on LDA and hLDA, respectively. By using hLDA, we can automatically extract not only the topics but also the hierarchical structure of these topics from the text data, thus allowing us objectively and understandably construct the path model of SEM with hLDA.

However, this approach faces several challenges. The first of these is the identification problem. As the approach utilizes the term frequency of keywords, the zero-frequency problem may occur, because some keywords appear less frequently in some documents. Another problem, which is related to the first one, is the significance level for paths, that is, the fewer the term frequency, the harder the estimations of the path coefficients. Therefore, the generated models consist of insignificant paths that are difficult to explain and not at all reliable.

The root of the abovementioned problems lies in the low frequency of each keyword. As a solution, we aim to increase term frequency artificially. In doing so, the ontology approach is a useful technique, because it semantically regards equivalent words as the same words. However, manually constructing an ontology from scratch, especially for new domains like Kickstarter, can be quite costly. Thus, we utilize the word2vec which automatically creates a similarity structure among words from text data.

III. PRE-PROCESSING BY USING WORD2VEC

A. Word2vec

Word2vec is a simple neural network composed of two layers: hidden and output layers [7][8]. By grouping similar words, distributed representations of words in a vector space help learning algorithms achieve better performance in natural language processing tasks. The neural network of Word2vec includes two architectures: Continuous Bag-of-Words Model (CBOW) and Skip–Gram. The former uses continuous distributed representations of the context. The best performance on the task introduced in the next section is obtained by building a log-linear classifier with four future and four history words as inputs, where the training criterion is to correctly classify the current (middle) word. Using the Skip–Gram model, Mikolov et al. introduced an efficient method for learning high-quality vector representations of words from large amounts of unstructured text data. Unlike many of the previously used neural network architectures for learning word vectors, the Skip–Gram model does not involve dense matrix multiplications.

Another important technique that is used to derive word embedding is called negative sampling. While negative-sampling is based on the Skip–Gram model, it is in fact, optimizing a different objective. What follows is the derivation of the negative-sampling objective.

The word representations computed using neural networks are unique, because the learned vectors explicitly encode many linguistic regularities and patterns. Furthermore, many of these patterns can be represented as linear translations. For example, the result of a vector calculation vec(“Tokyo”) - vec(“Japan”) + vec(“France”) is closer to vec(“Paris”) than to any other word vector.

B. Data Pre-processing

This section describes our proposed text pre-processing using word2vec. First, for learning the word2vec knowledge model, we obtain the corpus as word2vec import text data. After acquiring the knowledge model, the word2vec vector computing is ready for use. Next, word2vec is used to compute the feature vector for every word in the target text corpus, thereby comprising the text data for analysis. At this point, we perform the pre-processing of the target text data. Word2vec can compute word similarities as cosine similarities, and the similarity scores range between 0 and 1. If the score is closer to 1, this indicates that the similarity is the higher. Focusing on target text data, we compute the

TABLE I. RESULT FOR EACH PARAMETER

Model number	Number of keywords	threshold value	GFI	AGFI	RMSEA	Number of variable exceeding significance level
1	3	without wd2vc	0.871	0.792	0.0836	11
2	3	0.9	0.893	0.827	0.0682	10
3	3	0.8	0.894	0.829	0.0669	11
4	3	0.7	0.881	0.807	0.0772	11
5	3	0.6	0.842	0.745	0.1007	10
6	3	0.5	0.833	0.731	0.127	15
7	5	without wd2vc	0.86	0.805	0.0678	9
8	5	0.9	0.889	0.845	0.0472	11
9	5	0.8	0.89	0.845	0.0468	12
10	5	0.7	0.864	0.809	0.0659	9
11	5	0.6	0.844	0.781	0.0774	12
12	5	0.5	0.841	0.777	0.079	5

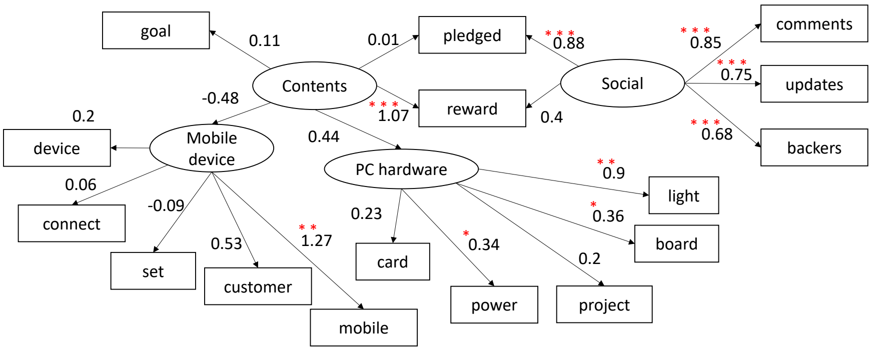

Fig. 2. Five keywords without word2vec (previous model [4])

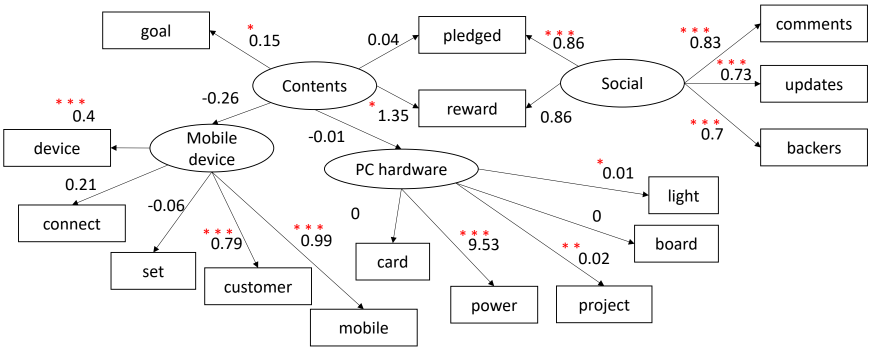

Fig. 3. Five keywords with a threshold value of 0.8

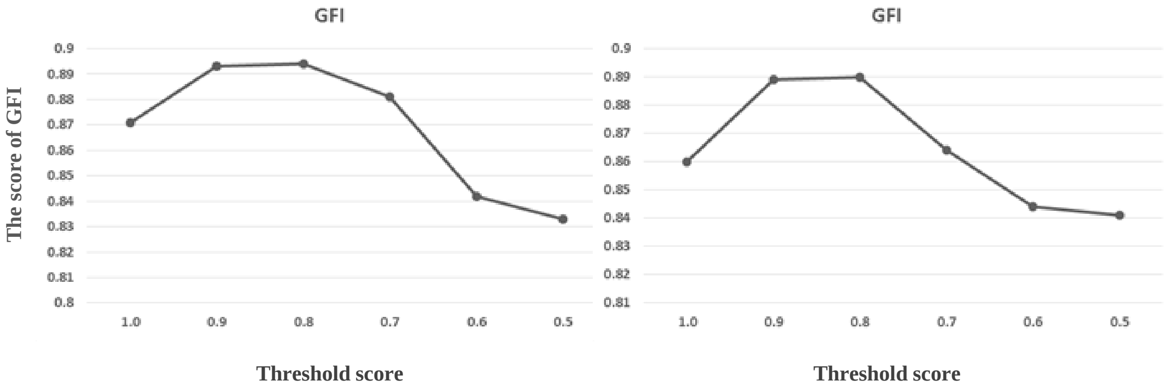

Fig. 4. Relationship between threshold score and GFI (Left: Models 1 to 6; Right: Models 7 to 12)

similarity between two words, and if the similarity exceeds the threshold score, we equalize the compared words.

Next, we obtain the modified term frequency that artificially affects similarity as term frequency, and is denoted by tf_i'

$$tf_i' = tf_i + \sum_{j \in S} sim(i, j) \cdot tf_j \qquad (1)$$

where tf_i is the term frequency of word i appearing in the text, and $sim(i,j)$ shows the similarity between words i and j. In addition, S is a set of words similar to word i, which exceed the threshold value. For example, for three words “A,” “B,” and “C,” if the similarity between “A” and “B” is over the threshold score, and “C” is not similar to either “A” or “B,” then “B” is converted to word “A” times by sim(A, B) and “C” is ignored. Note that, as the threshold value becomes lower, many words may be regarded as the same. This means that the low threshold value transforms the model into more abstracted one.

IV. Experiment

A. *Dataset and Experiment Process*

We perform the experiment to confirm that our proposed approach improves model fitness and significance level of paths. In this experiment, we collected from Kickstarter 84 live hardware category project data from the technology genre from a specific date (April 28, 2015) [9]. The data include the “backers,” which indicate the number of persons who invested in a project; “pledged,” which show the investment amount regarded as funded; “update,” which is the number of updates of a project; “comments,” which indicate the number of interaction with investors; and “reward information” to build the social variable. Regarding the reward information, we collected the smallest amount of money required to obtain a product or an item or to receive the first reward. In this study, we focused on “hardware” category for analysis. Therefore, for the learning corpus for word2vec, we extracted 10,000 pages of text data from a Google web search of the word “hardware.” To maintain generality, we performed additional learning by Text8 corpus (the corpus was the word sample text data in the genism package).

The models were evaluated based on the GFI, AGFI, and RMSEA indices. GFI and AGFI have values between 0 and 1; the higher the value of the model is, the better the model. In general, AGFI is lower than GFI. Meanwhile, RMSEA should be lower than 0.10; if the value is lower than 0.5, then the model is considered a good model and is between compatibility and information quantity. For topic extraction using hLDA, we used the method employed by Mallet [10]. For SEM analysis, we used the SEM package (3.1-5) provided in R (3.2.0) [11][12]. For word2vec, we used the genism package (0.12.4) in Python (3.5.2) [13]. As the parameter of word2vec, we changed the threshold score between 0.5 and 0.9 per 0.1.

B. Result and Discussion

Table 1 shows the result of the analysis. The evaluation measure is better when the threshold score is 0.8. Models 2 and 6 have higher GFI and AGFI indices and lower RMSEA scores compared with the other models. Therefore, our proposed process of using the SEM with hLDA analysis model is useful. We also show the relations between word similarity and GFI (see Fig. 2 and 3.), which indicate that a word similarity score of 0.8 has the best evaluation value.

As can be seen, Models 6, 9, and 11 have better scores in the number of variables exceeding the significance level. The results of the conventional and the proposed models in Fig. 2 and 3, respectively, are compared. The asterisks *, **, and *** indicate significance at the 10%, 5%, and 1% levels, respectively. In Model 5, the number of variables that reach the significance level increases.

One common problem encountered by previous research is that it is difficult to construct a stable analysis model. This suggests that some analysis models do not have good evaluations in terms of GFI and AGFI scores. In comparison, our proposed method successfully increased the evaluation value. In addition, Fig. 2 and 3 show there exist optimal similarities with which to equalize words. Fig. 4 shows that a model with a threshold value under 0.7 does not indicate a good evaluation. When the similarity is low, the words that do not have a strong relationship are regarded as keywords. Hence, the observed variables lose touch with the models, because words that comprise the topic change significantly, as in the case with the models with a threshold value 0.9, which is worse than 0.8. Hence, it is difficult to distinguish a model’s threshold value 0.9 from the conventional models (without word2vec).

In Model 5, p-values from some variables are not computed, because the number of words is insufficient. Meanwhile, in Model 6, p-values are computed in all variables. This is because by using word2vec, our proposed process can regard a word, which is similar to an actual keyword, as a keyword in itself. Thus, the term frequency of each keywords is high enough to compute.

V. Conclusions

This study proposed a method of pre-processing for text-based analysis with hLDA and SEM by using word2vec. The basic idea of the proposed process is that pre-processing text data as corpus for topic modelling is an effective approach in performing factor analysis by using SEM. Previous research suffered from such problems as an unstable construction analysis model and low significance levels. Here, we proposed the use of word2vec to achieve keyword flexibility in the text corpus for topic modelling. The results demonstrate that our proposed approach successfully resolved the abovementioned problems.

For our future work, we plan to combine word2vec with topic modelling. In other words, we will combine neural-network and LDA. Moreover, we aim to improve the proposed process so that word2vec can adjust to many kinds of special text data, such as Twitter posts and customer reviews.

Acknowledgment

This research was supported by KAKENHI 25240049.

References

[1] S. Kawanaka, A. Miyata, R. Higashinaka, T. Hoshide, K. Fujimura, “Computer analysis of consumer situations utilizing topic model,” 25th Annual Conference of the Japanese Society for Article Intelligence, 2011

[2] K. Wajima, T. Ogawa, T. Furukawa, S. Shimoda, “Specific negative factors using latent dirichlet allocation,” DEIM Forum, 2014

[3] R. Kunimoto, H. Kobayashi, R. Saga, “Factor Analysis for Game Software Using Structural Equation Modeling with Hierarchical Latent Dirichlet Allocation in User’s Review Comments,” International Journal of Knowledge Engineering vol. 1, no. 1, 2015, pp54-58.

[4] R. Saga, S. Nohara, “Factor analysis of investment judgment in crowdfunding using structural equation modeling,” The Fourth Asian Conference on Information Systems, 2015 pp41

[5] R. Saga, T. Fujita, K. Kitami, K. Matsumoto, “Improvement of Factor Model with Text Information Based on Factor Model Construction Process,” IIMSS, 2013, pp222-230

[6] R. Saga, R. Kunimoto, “LDA-based path model construction process for structure equation modeling,” Artificial Life Robotics vol. 21, issue 2, 2016 pp155-159

[7] T. Mikolov, K. Chen, G. S. Corrado, J. Dean, “Efficient estimation of word representations in vector space,” CoRR, abs/1301.3781, 2013

[8] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, “Districted representations of words and phrases and their compositionality,” Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems. Proceeding of a meeting held December 5-8, Lake Tahoe, Nevada, United States, pages 3111-3119, 2013

[9] Kickstarter, https://www.kickstarter.com/

[10] MALLET: A Machine Learning for Language Toolkit: http://mallet.cs.umass.edu, 2002

[11] The R Project for Statistical Computing. http://www.r-project.org/

[12] Fox, J.: Structural Equation Modeling with the SEM Package in R. Structural Equation Modeling, vol. 13, 2006, pp465-486.

[13] Genism: A topic modeling free python library. https://radimrehurek.com/gensim/index.html, 2016