Qualitative and Quantitative Methods in Libraries (QQML) 6: 371-380, 2017

Automatic Classification of Wikipedia Articles by Using Convolutional Neural Network

Keita Tsuji

Faculty of Library, Information and Media Science, University of Tsukuba, Japan

Abstract: Wikipedia has emerged as an important source of information for university students. It has been reported that the students tend to start their search with Google that leads to Wikipedia articles even in university libraries. Recent research findings indicate that relatively few students actually search and read books. Within this context, we are now developing a system that recommends books based on the articles Wikipedia users read in libraries (the system will be added to web browsers of libraries' desktop PCs). Such a system aims to encourage students to read library books as a more reliable source of information rather than relying on Wikipedia articles. Nippon Decimal Classification (NDC) categories are found to be an effective machine learning method for book recommendation. Therefore, if NDC categories could be assigned to Wikipedia articles, they might be used as an effective tool for book recommendation. Accordingly, we developed a method to automatically assign NDC categories to Wikipedia articles by using convolutional neural network (CNN), which is one of the representative methods of deep learning. We found that the accuracy of assigning top-level (i.e. Main Class) and second-level (i.e. combination of Main Class and Division) of NDC reached 87.7% and 74.7%, respectively. These results were achieved by using titles and categories of Wikipedia articles as input to CNN, while the accuracies obtained by other combinations such as titles, categories, and main texts were relatively poor.

Keywords: automatic classification; Wikipedia; convolutional neural network; Nippon Decimal Classification; NDC; category assignment

1. Introduction

Wikipedia has become an important source of information that has been widely used by university students. It has been reported that the students tend to start their search with Google that leads to Wikipedia articles even in university libraries. The recent research findings indicate that relatively few students actually search and read books (Anbiru et al. 2010). Within

Received: 21.5.2017 Accepted: 2.9.2017 ISSN 2241-1925

this context, the we are now developing a system that recommends books based on the articles Wikipedia users read in libraries (the system will be added to web browsers of libraries' desktop PCs). Such system aims to encourage students to read library books as a more reliable source of information rather than relying on Wikipedia articles. Nippon Decimal Classification (NDC) categories are found to be an effective machine learning method for book recommendation (Tsuji et al. 2013, 2014, 2015). Therefore, if NDC categories could be assigned to Wikipedia articles, they might be used as an effective tool for book recommendation. Based on this background, we developed a method to automatically assign NDC categories to Wikipedia articles by using convolutional neural network (CNN), which is one of the representative methods of deep learning. While some studies have focused on automatically assigning NDC categories to reference records in libraries (Arai & Tsuji 2015), relatively few studies have focused on Wikipedia articles.

As a first step toward automatic assignment of NDC categories to Wikipedia articles, the present study focused on NDC categories of books cited in the articles. We believe that Wikipedia articles fall into the same (or similar) NDC categories as those of books cited in these articles. For instance, if article X cites many books whose NDC categories are "324," X is likely to belong to the "civil law" category. Based on this assumption, we extracted articles that were citing many books whose NDC categories were mostly the same and used such articles as training and testing data for CNN. We considered the most popular NDC category in each article as a label that CNN should output for that article (supervisory signals). For instance, if article X cites many books whose NDC categories are "324," we assume that CNN should output "324" for article X.

For input to CNN, information obtained from titles, categories, and main texts of articles were used. More specifically, we used distributed representation of nouns contained in titles, categories, and main texts that were obtained through word2vec. We examined which combination would produce the best accuracy. The tested combinations were (1) titles only, (2) categories only, (3) main texts only, (4) titles and categories, (5) titles and main texts, (6) categories and main texts, and (7) titles, categories and main texts.

2. Related Studies

Kim (2014) reported the results of various experiments on CNN text classification. Texts used were movie reviews, various products reviews, subjectivity and opinion polarity datasets, Stanford Sentiment Treebank and TREC. Wang et al. (2015) proposed to use CNN for short text classification. Texts used were Google Snippets and TREC. Johnson & Zhang (2015) examined effective use of word order for CNN text classification. Texts used were movie reviews, electronic product reviews

and Reuters news articles. However, few studies have been conducted on classifying Wikipedia articles.

3. Method

In this section, we will explain the training and testing data sets used in this study, matrix input to CNN, and CNN settings.

3.1 Training and Testing Data

The steps below were followed to obtain the required data for training and testing purposes:

(1) The dump of the Japanese Wikipedia was downloaded from https://dumps.wikimedia.org/jawiki/ (as of February 21, 2015).
(2) The pages whose namespace (ns) tag was "0," which means that the page was "Main/Article" (henceforth referred to as "articles") were extracted.
(3) Word2vec in gensim was used to obtain 200-dimension distributed representation for each noun in the Wikipedia articles (we used mecab 0.996 for morphological analysis). We set the parameters *window* and *min_count* for Word2Vec module in gensim to 5 and 10, respectively.
(4) The articles extracted in Step (2) were used to extract the articles, which contain "References."
(5) Articles with "References" that contained bibliographies of books showing their ISBNs were extracted.
(6) The correct and consistent books bibliographies were searched by using the ISBNs in OpenSearch that provided by the National Diet Library of Japan.
(7) Bibliographies of books that did not contain triple-digit NDCs (i.e. which consisted of Main Class, Division, and Section) or titles were excluded.
(8) The number of NDCs in each Wikipedia article was counted to identify the most popular NDC and its frequency in the bibliography. As for the level or depth of NDC, two types were used i.e. Main Class only and the combination of Main Class and Division. For instance, if an article contained six books and their NDCs were 324, 324, 324, 325, 367, and 369, NDC "32" (Main Class is "3" and Division is "2." Incidentally, "32" represents "Law" and "36" represents "Society") was considered as the most popular combination in terms of Main Class and Division, and its frequency was four. When the Main Class was only considered, "3" was the most popular, and its frequency was six.
(9-1) The articles whose Main Class numbers of NDCs obtained through Step (8) were no less than three and the ratios of the most popular ones were no less than 0.8 were extracted. Hence, the example article mentioned in Step (8) was extracted because the number of citing books is six (>=three) and the ratio is 1.0 (>=0.8). This set of articles will be referred to as L1.

(9-2) Similarly, the articles whose combinations numbers of Main Class and Division of NDCs obtained through Step (8) were no less than three and the ratios of the most popular ones were no less than 0.8 were extracted. Thus, the example article mentioned in Step (8) was NOT extracted because the ratio is 0.67 (=4/6 < 0.8). This set of articles will be referred to as L2.
(10) It is assumed that the most popular Main Class of NDC in each article in L1 is the one, which should be assigned to that article by CNN. For instance, the Main Class of NDC that should be assigned to the example article mentioned in Step (9-1) is "3." Accordingly, a set of pairs of the abovementioned Main Class of NDCs and correspondent articles were developed as the training and testing data for CNN. Similarly, a set of pairs of the abovementioned combination of Main Class and Division of NDCs and correspondent articles concerning L2 set were developed.

The total number of articles in the Japanese Wikipedia is 2,689,050 articles, while the number of articles obtained through Step (2) was 1,104,962. The number of nouns in Step (3) input to word2vec was 3,645,356. The number of articles obtained through Steps (4) and (5) were 95,194 and 28,154, respectively. The number of ISBNs contained in 28,154 articles, and thus used in Step (6) was 66,295 (including duplicates). The number of bibliographies obtained through Step (7) was 61,173 (including duplicates). The number of articles that contained at least one such bibliography was 26,879. Therefore, NDCs were obtained from 28.2% (= 26,879 / 95,194) of articles. For more detailed information, see Tsuji (2016).

The numbers of articles of L1 and L2 sets are 3,985 and 3,091, respectively. These were divided into training and testing data sets. The number of articles in the testing data set from both L1 and L2 sets were 300. While the rest were included in the training data set, i.e. 3,685 (=3,985-300) for L1 and 2,791(=3,091-300) for L2 set, respectively.

3.2 Matrix Input to CNN

First, the titles from each article in L1 (and L2) set were extracted and represented by 5 times 200 matrix whose *i*-th row is the 200-dimension vector (obtained through Step (2)) for *i*-th noun in title. If the number of nouns in a certain title was less than five, the corresponding row was set to zero vector. If the number of nouns was more than five, only the first five nouns would be adopted in the title. Henceforth, this matrix was referred to as *tl*.

Then, the first five categories from each article in L1 (and L2) set were extracted and represented by 25 times 200 matrix whose (*i* times *j*)-th row is the 200-dimension vector (obtained through Step (2)) for *j*-th noun in *i*-th category. If the number of nouns in a certain category was less than five, the corresponding row was set to zero vector. If the number of nouns was

more than five, only first five nouns in each category would be adopted. Henceforth this matrix was referred to as *cg*.

Finally, ten nouns from the main text of each article in L1 (and L2) set whose TF-IDFs were the highest were extracted. TF-IDF of noun X in article Y is defined as "the frequency of X in the main text of article Y" times "log (the number of articles/the number of articles whose main texts contain X)". These were represented by 10 times 200 matrix whose *i*-th row is the 200-dimension vector (obtained through Step (2)) for noun whose TF-IDF was the *i*-th highest in the main texts. Henceforth this matrix was referred to as *tx*.

Seven types of matrices were tested as the input to convolutional neural network. They were (1) *tl*, (2) *cg*, (3) *tx*, and their vertically concatenated matrices, as follows: (4) *tl+cg*, (5) *tl+tx*, (6) *cg+tx* and (7) *tl+cg+tx* (where "+" represents the vertical concatenation). The numbers of their rows are 5, 25, 10, 30, 15, 35, and 40, respectively. They all have the same number of columns, i.e. 200.

3.3 CNN Settings

The convolutional neural network is well-known for its ability to automatically classify images. The previously mentioned matrices can be considered as (2D) images. Similar to visual recognition, five sets of heights of filters were tested, i.e. (a) {1, 2, 3}, (b) {3, 4, 5}, (c) {1, 2, 3, 4, 5}, (d) {3, 4, 5, 6, 7}, and (e) {1, 2, 3, 4, 5, 6, 7}. The widths of all filters were the same, i.e. 200. Furthermore, three kinds of number of filters for each filter size for L1 set were tested, i.e. (i) 20, (ii) 30 and (iii) 40. While for L2 set, we tested 100, 150, 200, 250, 300, 350, 400, 450, and 500. Therefore, for instance 150 (=5 times 30) filters were used when the combination of (d) and (ii) were tested for L1 set.

Tensorflow was used to form and train CNN. The *strides* and *padding* for *tf.nn.conv2d* function were set to [1, 1, 1, 1] and *VALID*, respectively. Rectified linear units (*tf.nn.relu*) were used after *tf.nn.conv2d*. Then we used max pooling. The *ksize*, *strides* and *padding* for *tf.nn.max_pool* function were set to [1, IH-FH+1, 1, 1], [1, 1, 1, 1] and *VALID*, respectively, where IH and FH represent the height of input matrix and that of the filter. Softmax function was used in the final output layer. The size of mini batches, the number of epochs, dropout rate, and learning rate were 300, 300, 0.5, and 0.0001, respectively. The data flow is illustrated in Figure 1.

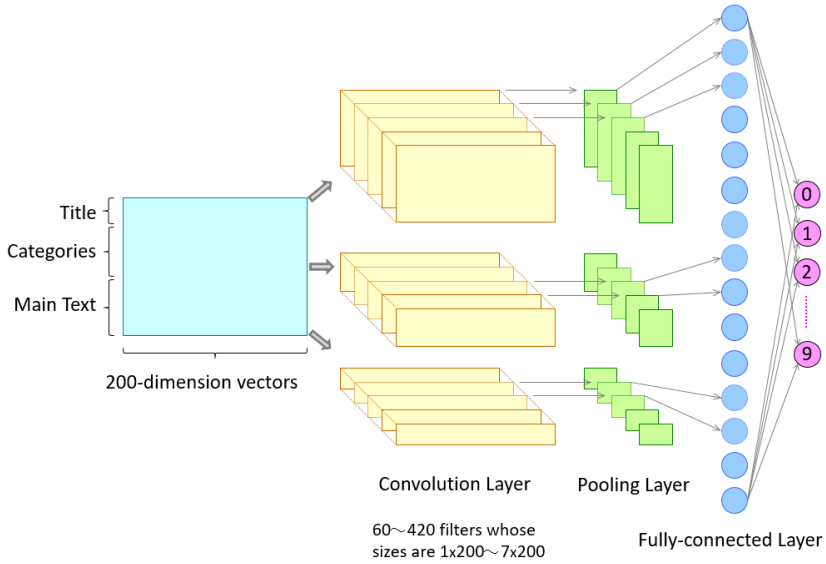

Figure 1. Data Flow of the Developed CNN

4. Results

The results are shown in Tables 1 and 2, in which "FS" represents the filter sizes. For instance, "1 to 5" in FS column indicates that the used filter sizes were 1, 2, 3, 4, and 5. While "NF" represents the number of filters. "—" represents that we could not apply filters whose heights were more than five to *tl* matrix because the number of row (height) of *tl* was just five.

As shown in Table 1, the results obtained by using the combination of both titles and categories (*tl+cg*) are better compared to those of the other combinations. Note that using all the titles, categories and main texts (*tl+cg+tx*) does not outperform using only titles and categories (*tl+cg*). Using the main texts of Wikipedia entries might have caused problem determining their NDC categories. However, considering that only ten nouns with the highest TF-IDFs were used, thus there might be other effective ways to utilize main texts.

Based on the preliminary experiment results of L2 set, it is found that better results were obtained by the filter size set {1, 2, 3, 4, 5} compared to the other sets. The results obtained by this filter size set are shown in Table 2. It can be seen in Table 2 that *tl+cg* produces higher accuracy compared to the other sets with an accuracy reaches 74.7% when the NF was set at 300.

Table 1. Accuracies of NDC Categories Assignment of L1 Set

FS	NF	*tl*	*cg*	*tx*	*tl+cg*	*tl+tx*	*tx+cg*	*tl+cg+tx*
1 to 3	20	70.7	80.7	79.3	84.3	78.3	82.3	82.3
	30	71.0	81.0	79.7	84.3	78.0	82.7	81.3
	40	70.3	81.0	78.3	84.3	79.0	81.7	81.3
3 to 5	20	70.7	81.3	77.7	83.7	78.0	80.3	80.7
	30	68.7	82.3	79.0	84.3	78.7	82.3	82.0
	40	69.0	81.7	79.3	86.7	77.3	82.0	82.0
1 to 5	20	68.7	82.0	78.0	84.7	78.0	82.3	83.0
	30	68.3	82.3	78.0	87.7	79.3	83.7	83.0
	40	68.7	84.3	80.0	87.3	81.7	83.7	83.0
3 to 7	20	—	81.3	78.7	84.7	78.7	83.3	81.3
	30	—	83.3	80.7	87.7	81.0	84.7	83.3
	40	—	84.0	80.3	86.3	81.0	83.7	82.3
1 to 7	20	—	82.0	79.7	87.0	79.0	82.3	83.3
	30	—	84.0	80.3	85.7	81.0	84.3	83.3
	40	—	84.3	80.7	86.0	80.7	83.3	82.3

Table 2. Accuracies of NDC Categories Assignment of L2

NF	*tl*	*cg*	*tx*	*tl+cg*	*tl+tx*	*tx+cg*	*tl+cg+tx*
100	52.7	67.0	65.0	72.7	64.0	70.7	70.3
150	53.7	66.7	65.0	73.7	65.3	72.3	71.7
200	54.3	68.0	64.7	74.0	66.0	72.7	71.7
250	54.0	72.3	65.0	74.0	66.7	72.3	72.3
300	53.3	68.3	65.7	74.7	66.7	70.3	72.0
350	53.7	68.3	68.7	74.0	71.0	73.0	71.0
400	53.3	68.7	69.7	73.7	68.3	71.0	71.3
450	53.3	68.3	68.0	74.0	67.7	72.0	71.7
500	53.3	68.3	65.3	73.3	68.0	70.0	72.3

The numbers of epochs and accuracies concerning *tl+cg* and *tl+cg+tx* with NF={30, 40, 60} for L1 are shown in Figure 2. We can see in Figure 2 that even if we increase the number of epochs (to 500), the accuracies do not increase and the accuracies of *tl+cg* are higher than those of *tl+cg+tx*.

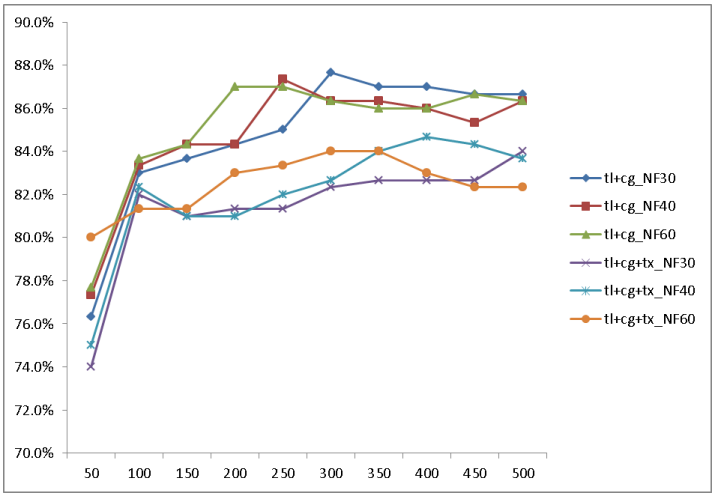

Figure 2. Numbers of Epochs and Accuracies

The numbers of NDCs which should be assigned (i.e. label) and those of NDCs actually assigned by CNN with *tl+cg*, NF=30 and FS={1, 2, 3, 4, 5} for L1 are shown in Table 3. Table 3 also shows the accuracy for each NDC label. We can see in Table 3 that accuracies for NDC labels (i.e. which should be assigned) which appeared no less than 20 times in our testing data were no less than 90.0% (i.e. "2" (History), "4" (Natural Sciences), "5" (Technology), and "7" (The Arts)). On the other hand, the accuracy for "3" (Social Sciences) was relatively low (50%) although the number of their samples was not extremely small (i.e. 16). Among the eight articles which the CNN failed to assign "3" were "Western costume (16th century)," "... (15th century)," "... (14th century)," "... (13th century)," and "Western costume (11th and 12th centuries)." The last one was assigned "7" (The Arts) and the rests were assigned "2" (History). The articles on Western costume in these centuries are related to history and probably the arts. It might not cause a serious problem if we recommended books whose NDCs were "2" or "7" to these Wikipedia articles.

Table 3. Assigned NDCs and Accuracies

		NDCs Assigned by CNN with tl+cg, NF=30 and FS={1, 2, 3, 4, 5}										Total	Accu-racy
		0	1	2	3	4	5	6	7	8	9		
NDCs which should be Assigned	0 (General Works)	0	0	1	1	1	0	0	0	0	0	3	0.0
	1 (Philosophy)	0	8	3	1	0	0	0	0	0	0	12	66.7
	2 (History)	0	1	153	5	2	3	0	1	0	0	165	92.7
	3 (Social Sciences)	0	0	6	8	1	0	0	1	0	0	16	50.0
	4 (Natural Sciences)	0	0	2	0	37	0	0	0	0	0	39	94.9
	5 (Technology)	0	0	0	1	1	18	0	0	0	0	20	90.0
	6 (Industry)	0	0	0	1	0	1	0	0	0	0	2	0.0
	7 (The Arts)	0	0	0	0	0	0	0	39	0	0	39	100.0
	8 (Language)	0	0	0	0	0	0	0	0	0	0	0	—
	9 (Literature)	0	1	2	0	0	0	0	1	0	0	4	0.0
Total		0	10	167	17	42	22	0	42	0	0	300	87.7

5. Conclusions

Based on the assumption that recommending books to Wikipedia reader in libraries is an effective method to encourage students read books, we developed a method to automatically assign NDC categories to Wikipedia articles. The study results showed that using a combination of titles and categories was more effective than using other combinations such as titles, categories and main texts. The accuracy of L1 set (i.e. Main Class of NDC) reached 87.7%, which is considered significant to help determine which books should be recommended. Accordingly, our future research will focus on developing a method to actually determine the books to be recommended using the present study's results.

References

Anbiru, T. et al. (2010). Information Seeking Behavior. *Proceedings of the Spring Meeting of the Japan Society of Library and Information Science*, 87–90. (text in Japanese).

Arai, S. and Tsuji, K. (2015). Automatically Assigning NDC Categories to Reference Service Records by Using Machine Learning Methods. *Journal of Japan Society of Information and Knowledge*, Vol. 25, No. 1, 23–40. (text in Japanese).

Johnson, R. and Zhang, T. (2015). Effective Use of Word Order for Text Categorization with Convolutional Neural Networks. *Proceedings of Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL*, 103–112.

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1746–1751.

Mecab. <http://taku910.github.io/mecab/>.

Tensorflow. <https://www.tensorflow.org/>.

Tsuji, K. et al. (2013). Book Recommendation based on Library Loan Records and Bibliographic Information. *Proceedings of the 3rd International Conference on Integrated Information (IC-ININFO 2013)*, 8p. (No Pagination).

Tsuji, K. et al. (2014). Book Recommendation Using Machine Learning Methods Based on Library Loan Records and Bibliographic Information. *Proceedings of the 5th International Conference on E-Service and Knowledge Management (ESKM 2014)*, 76–79.

Tsuji, K. et al. (2015). Book Recommendation Using Machine Learning Methods Based on Library Loan Records and Bibliographic Information. *International Journal of Academic Library and Information Science*, Vol. 3, No.1, 7–23.

Tsuji, K. (2016). Books Cited in Wikipedia: Possibility to Use their Nippon Decimal Classification Categories for Book Recommendation. *Proceedings of the 7th International Conference on E-Service and Knowledge Management (ESKM 2016)*, 1196-1197.

Wang, P. et al. (2015). Semantic Clustering and Convolutional Neural Network for Short Text Categorization. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 352–357.

Word2vec. <https://radimrehurek.com/gensim/models/word2vec.html>.