American Journal of Neural Networks and Applications
2019; 5(1): 28-35
http://www.sciencepublishinggroup.com/j/ajnna
doi: 10.11648/j.ajnna.20190501.15
ISSN: 2469-7400 (Print); ISSN: 2469-7419 (Online)

Random Walk-Based Semantic Annotation for On-demand Printing Products

Mingxi Zhang*, Guanying Su

College of Communication and Art Design, University of Shanghai for Science and Technology, Shanghai, China

Email address:
WAXL7461@aliyun.com (Mingxi Zhang), guanying_su123@163.com (Guanying Su)

*Corresponding author

To cite this article:
Mingxi Zhang, Guanying Su. Random Walk-Based Semantic Annotation for On-demand Printing Products. *American Journal of Neural Networks and Applications*. Vol. 5, No. 1, 2019, pp. 28-35. doi: 10.11648/j.ajnna.20190501.15

Received: April 29, 2019; **Accepted**: June 24, 2019; **Published**: July 4, 2019

Abstract: Nowadays, the scale of real network is increasing day by day, while also brings sparse problems. It is usually necessary to maintain a large number of product information. To organize this product information, a feasible way is to add semantic tags to the information. In this article, we aim to solve the problem of semantic annotation of on-demand printing products. Based on good properties of random walk in global networks, we deal with the sparsity problem by applying it, and then propose an efficient ProRWR algorithm. Firstly, it processes the text description dataset of printed products based on TF-IDF algorithm, and builds "product-term" bipartite network. Secondly, ProRWR builds square matrix using the TF-IDF weight matrix, rewrite the equation of random walk, and use the normalized square matrix as the input of rewrite ProRWR algorithm. By random walks, terms with the highest convergence probability in each product document are selected as the most relevant feature terms of the product. A large number of experiments have been done on Amazon dataset. The results show that the precision and recall of our algorithm are 73.5% and 60%, respectively, indicating that ProRWR has discovered the potential semantic association and implemented the semantic annotation of on-demand printed products.

Keywords: TF-IDF, Random Walk, Semantic Annotation

1. Introduction

Affected by the digitalization of the network, the traditional large-scale printing mode has been unable to adapt to the individualized market demand. Different user groups have significant differences in the demand for different products. On-demand printing refers to the sale of a reasonable number of products by pre-printing, and then timely supplementary printing according to the sales situation of the products and the needs of the public. However, faced with massive information of network products, users hope to search for a special product quickly and correctly. At the same time, the enterprise also hopes to print in time according to user's needs. How to efficiently and accurately obtain the required data from massive product data to improve the performance of information retrieval is one of the important issues that enterprises must focus on.

Semantic annotation [1] is an attractive method in machine learning and data mining, which is useful for indexing and organizing the product information. In the on-demand printing platform, semantic annotation is to add text tags to on-demand printing products, which facilitates the retrieval and management of products. The impact of semantic annotation on on-demand printing is mainly reflected in the proximity searches, recommendation, classification, clustering and other aspects of on-demand printing products. For example, by adding semantic tags to products, it is convenient to cluster or classify different categories of products. When users search for products, the system can accurately find related products based on semantic tags.

With the vigorous development of semantic annotation field, various kinds of research are deepening, and many novel tagging approaches are emerging. For example, content-based semantic annotation method and model-based semantic annotation method. Content-based approaches [2-4] mainly study how to combine the network metadata information, user comments, attention, clicks and other information during annotation stage. In contrast to the content-based algorithms, model-based algorithms [5-9] often use machine learning to

solve the problem of semantic annotation. Broadly speaking, machine learning gives machine learning ability, which plays an important role in the identification of human diseases [10], classification of products [11], feature selection [12] and image processing [13].

Aiming to establish microblog user interest model, [2] combined clustering and classification algorithm to extract user interest tags and [5] proposed an approach of automatic document annotation with data mining algorithms: classification, clustering and named-entity recognition. [3] used content-based filtering method and distance algorithm for journal Recommendation System. [4] applied context information to alleviate the negative impact of data sparsity, and uses hierarchical relationships among products to mine users' potential preferences, and then models users in a specific period of time. An automation framework is mentioned in [6], which extracts product adopter information from online reviews and incorporates the extracted information into feature-based matrix decomposition to more efficiently recommend products. [7] employed association rule mining and Apriori algorithm for product prediction and recommendation. [8] proposed a concept-based automatic semantic annotation method for online BIM product documents. [9] used K-nearest neighbor algorithm to propose annotation methods of the image from the semantic neighborhood propagation label. These methods provide important reference for semantic tag generation of on-demand printed products.

Through the above analysis, we found that some existing machine learning methods [14-15] just only used to annotate the keywords that appear in the document, and cannot present terms that do not appear in the document. Among various annotation measures, random walk with restart (RWR) [16-17] provided useful node-to-node relevance scores by considering global network structure [18] and intricate edge relationships [19], which can discover potential semantic relationship between documents and terms. Moreover, RWR is a stable measurement standard and is not susceptible to noise and missing data. The traditional random walk-based algorithm has been applied to community detection [20], link prediction [21] disease detection [22], entity classification [23], image annotation [24] and other techniques. Based on above observations, this paper researches the semantic annotation of on-demand printing products based on RWR model.

2. Overview of Random Walk with Restart

Random walk with restart (RWR) [25] has become more attractive measure in the field of data mining and internet [26-27], which can discover potential entity tags and mine potential semantic relationships by calculating random walk distances that can be defined by relevance scores. The basic idea of RWR is to traverse a graph from one or a series of vertices, compared to some traditional approaches of calculating the distance on the graph, such as the shortest path method, maximum flow [28] and so on. RWR can capture the multi-faceted information between node pairs and obtain the overall structural relationship of the graph [29-32].

Figure 1 is an example of a traditional random walk model graph, denoted as $G =< V, E >$, where V is the set of non-empty nodes of the same type, E is the set of edges between nodes, and the weights of the set of edges are the relational weights between node pairs. In this case, all relationships between nodes can be mapped into the $n \times n$ matrix (n represents the total number of nodes), the elements of which represent whether a node has a link with other nodes.

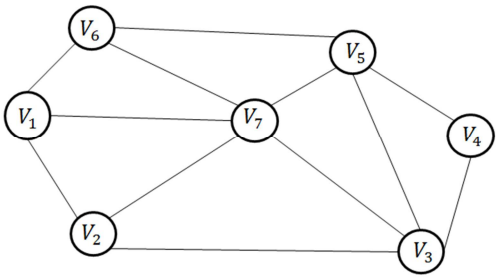

Figure 1. *Example of random walk graph.*

Eq. (1) is the calculation equation of the relational matrix in Figure 1. If there is a relationship between nodes, the P_{ij} is set to 1, otherwise, it is 0. In particular, there is no relationship between the nodes themselves, so the corresponding value is 0. In this way, we generate the 7×7 adjacency matrix P shown in Figure 2.

$$P_{ij} = \begin{cases} 0, i = j \\ 1, i \neq j \end{cases} i, j \in V \tag{1}$$

$$P = \begin{array}{c} \\ V_1 \\ V_2 \\ V_3 \\ V_4 \\ V_5 \\ V_6 \\ V_7 \end{array} \begin{array}{c} \begin{array}{ccccccc} V_1 & V_2 & V_3 & V_4 & V_5 & V_6 & V_7 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 \end{pmatrix} \end{array}$$

Figure 2. *7×7 adjacency matrix P.*

Let l be the maximum iteration step of random walk, $c \in (0, 1)$ is the restart probability, then the matrix of random walk distance from V_1 to V_7 [18] is:

$$R^l = \sum_k^l c(1-c)^k P^k \tag{2}$$

where R is the random walk distance matrix. According to this equation, the recurrence form of random walk distance matrix is derived as:

$$R^l = c(1-c)^l P^l + R^{l-1} \tag{3}$$

At any node, the traverser will walk to the neighbor node with probability $(1-c)$ and jump to any node in the graph with probability c. After each walk, we get a probability distribution, which characterizes the probability that each

node in the graph is visited. This probability distribution is used as the input of the next walk and iterates over and over again. When maximum iteration step is satisfied, a stable probability distribution [25] will be generated.

3. Algorithm Design

3.1. Bipartite Network Construction

The crucial question of product annotation in on-demand printing platform is how to construct a graph that can reflect the relevance between products and tags. The on-demand printing products and its descriptive terms can be regarded as the nodes in the graph, and the relationship between the nodes is edges. And then, the "product-term" bipartite graph is constructed. Figure 3 is a partial example of a bipartite network, denoted as $G' =< V', E' >$, $V' = (V_P \cup V_T)$ represents the set of products and terms, $P_1 \sim P_6$ are products, $T_1 \sim T_5$ are terms contained in these product descriptions. E' is the edges set between products and terms, and the weights between edges are the relevance scores between node pairs.

From Figure 3, we can find that: (1) P_1 is linked with T_1, T_2, T_4 but not with T_3, T_5, which indicates that the document of P_1 contains T_1, T_2 and T_4, and these three terms have great potential relevance to product P_1. (2) The term T_3 is associated with P_3 and P_5, indicating that the product P_3 and P_5 can be expressed simultaneously by term T_3, or that the two products may have similar features. (3) The product arrives at the term through odd steps, such as three steps from P_1 to T_5: $P_1 \rightarrow T_1 \rightarrow P_2 \rightarrow T_5$ or $P_1 \rightarrow T_2 \rightarrow P_4 \rightarrow T_5$. (4) If even steps are reached, product node P_2 or P_4 is reached.

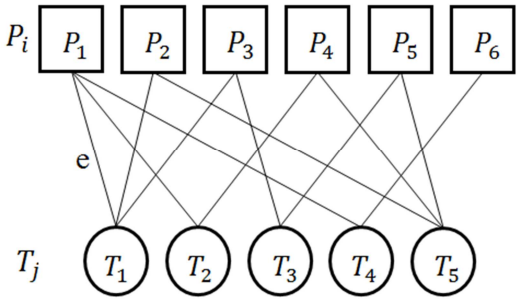

Figure 3. *Examples of product-term bipartite network.*

3.2. Product-Term Weight Computation

TF-IDF is a statistical method and a commonly used term weighting method for information retrieval, which usually used to specify the term weight of a class of documents and to evaluate the importance of a term to a document in corpus. The importance of the term increases proportionally with the number of times it appears in the document, but decreases inversely with the frequency it appears in the corpus.

The term frequency (TF) refers to the number of times the term appears in the document. Inverse document frequency (IDF) means that if the number of documents containing the term is smaller, the IDF is larger, indicating that the term has a good ability to distinguish categories. The main idea of TF-IDF is that if a term or phrase appears in a document with a high TF and rarely appears in other documents, then the term or phrase is considered to have good ability to distinguish categories and is suitable for classification [33]. TF-IDF is actually the product of TF and IDF. The larger the product, the more the term reflects the subject of this document. In [34], a keyword extraction algorithm based on TF-IDF is proposed, which combines the semantics and statistical weight of terms to extract keywords. [35] decomposed the eigenvectors generated by TF-IDF algorithm into singular values, and carried out emotional analysis of micro-blog combined with LSA. Based on above discussions, the TF-IDF algorithm has general applicability in the extraction of feature terms.

The equation for calculating TF-IDF is as follows:

$$tf_{i,j} = \frac{q_{i,j}}{\sum_m q_{m,j}} \tag{4}$$

$$idf_i = \frac{\log(|D|)}{\{j : i \in j\}} \tag{5}$$

$$tfidf_{i,j} = tf_{i,j} \times idf_i \tag{6}$$

where $q_{i,j}$ is the number of times the term j appears in the description of product i. $\sum_m q_{m,j}$ is the sum of occurrences of all terms in the description text of product i, m represents the number of terms contained in product i. $|D|$ is the total number of all products in the corpus.

Next, we construct "product-term" weight matrix A_{pt}, as shown in Figure 4. Here, each product description is represented by row vector, and each column corresponds to the terms. By Eq. (4-6), the $tfidf_{i,j}$ value of the matrix can be obtained, which represents the relevance between product P_i and term T_j.

$$A_{pt} = \begin{array}{c} \\ P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{array} \begin{array}{c} \begin{array}{ccccc} T_1 & T_2 & T_3 & T_4 & T_5 \end{array} \\ \begin{pmatrix} tfidf_{1,1} & tfidf_{1,2} & 0 & tfidf_{1,4} & 0 \\ tfidf_{2,1} & 0 & 0 & 0 & tfidf_{2,5} \\ tfidf_{3,1} & 0 & tfidf_{3,3} & 0 & 0 \\ 0 & tfidf_{4,2} & 0 & 0 & tfidf_{4,5} \\ 0 & 0 & tfidf_{5,3} & 0 & tfidf_{5,5} \\ 0 & 0 & 0 & tfidf_{6,4} & 0 \end{pmatrix} \end{array}$$

Figure 4. *"Product-Term" weight matrix.*

3.3. Initial Transition Probability Matrix Construction

From Eq. (2), it can be known that the basic idea of RWR is matrix multiplication calculation, which will involve the repeated transpose of the matrix, and lead to higher computational complexity. Therefore, in this section, we build a square matrix A to simplify the operation, as shown in Figure 5. The square matrix A can be regarded as a block matrix, which consists of A_{pt}, A_{tp} and two zero matrices. A_{tp} is a term-to-product matrix that is transposed with A_{pt}. It is worth noting that two zero matrices are placed on the main diagonal because the relationship between products and products, terms and terms is not considered.

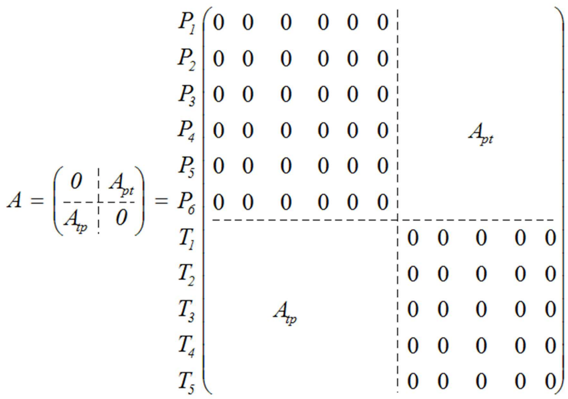

Figure 5. *Square matrix A.*

Next, for the convenience of calculation, A is normalized. The normalized equation is derived as:

$$\widetilde{A}_{ij} = \frac{a_{ij}}{\sum_{n=1}^{m} a_{ik}} \quad (7)$$

where $\widetilde{A}_{ij}$ represents the initial transition probability from product i to term j, a_{ik} represents the TF-IDF weight of the k term in product i row vector. In each product document, the total TF-IDF values of all terms are treated as denominators, and the TF-IDF weights of each term in product i are taken as molecules. In this case, the new values are obtained as the new elements of the $\widetilde{A}$. Since the main diagonal is two zero matrices, the normalization of the square matrix A can be regarded as the normalization of the A_{pt} and A_{tp} matrices.

3.4. ProRWR Algorithm

In this section, we first rewrite RWR equation into a more efficient and simple form, which is derived as:

$$R^{l+1} = c(1-c)\widetilde{A} + \widetilde{A}(1-c)R^{l} = c(1-c)\begin{pmatrix} 0 & \widetilde{A}_{pt} \\ \widetilde{A}_{tp} & 0 \end{pmatrix} + \begin{pmatrix} 0 & \widetilde{A}_{pt} \\ \widetilde{A}_{tp} & 0 \end{pmatrix}(1-c)R^{l} = (1-c)\begin{pmatrix} 0 & \widetilde{A}_{pt} \\ \widetilde{A}_{tp} & 0 \end{pmatrix}(c + R^{l}) \quad (8)$$

where $\widetilde{A}_{pt}$ and $\widetilde{A}_{tp}$ are normalized form by A_{pt} and A_{tp} respectively. When $l = 1$, R^1 is as follows:

$$R^{1} = c(1-c)\begin{pmatrix} 0 & \widetilde{A}_{pt} \\ \widetilde{A}_{tp} & 0 \end{pmatrix} \quad (9)$$

In order to further improve the accuracy of product annotation, we next introduced an improved algorithm, ProRWR algorithm, as shown in Algorithm 1. Before preforming random walk, we construct square matrix in line1-4 and use $\widetilde{A}$ as the initial transition probability matrix of ProRWR algorithm. Starting from a product node, random walk is performed on the "product-term" bipartite network in line 5-9. The speed of iterative convergence is determined by the restart probability c. R_{ij}^{l} is the relevance score between node i and node j, and is defined as the steady-state probability that particle i stays at node j after l steps. After the end of the walk, the larger the convergence probability after the walk, the more representative the term is to the product.

Algorithm 1 ProRWR Algorithm

Input: bipartite network G', restart probability c, iteration step l;

Output: *top-N* tags;

1 Construct matrix A_{pt} of G' by TF-IDF;

2 Calculate $\widetilde{A}_{pt}$;

3 Calculate $\widetilde{A}_{tp}$;

4 construct square matrix $\widetilde{A} = \begin{pmatrix} 0 & \widetilde{A}_{pt} \\ \widetilde{A}_{tp} & 0 \end{pmatrix}$;

5 for $t = 1$: l

6 $R^1 \leftarrow c(1-c)\widetilde{A}$;

7 $R^{t+1} \leftarrow c(1-c)\widetilde{A} + \widetilde{A}(1-c)R^{t}$;

8 until $R^{t} = R^{t+1}$;

9 end for

10 Sort and select top-N tags;

After l iteration, we obtained the convergence probability matrix. The matrix needed is the A_{pt} matrix in the upper right corner of the square matrix $\widetilde{A}$, elements of which are the convergence probability between products and terms, that is, the potential relevance score. By sorting these probability values, the top-N terms with larger convergence probability will be recommended as the tags of the product, that is, the semantic annotation of the product is realized.

4. Experimental Analysis

4.1. Setup

The program code for algorithm is written in the Java development tool Eclipse based on JDK 1.8.0 and JRE 1.8.0 and run under a computer with a Windows server 2012 (a) system environment. The computer configuration environment is 128GB memory, 2 core CPUs, and each core has a frequency of 1.70 GHz.

The experiment was conducted on the Amazon dataset (http://snap.stanford.edu/data/amazon-meta.html). We extract the label of Amazon Standard Identification Number (ASIN) and title in the dataset. Each ASIN number corresponds to a product, and the title is a basic text description of the product. The experiment selected the product's ID from 1 to 50,000 for analysis and processing.

4.2. Evaluation Criteria

In order to evaluate the effect of semantic annotation, the precision and recall are used to evaluate the quality of the results, which are the commonly used evaluation indicators of information retrieval and recommendation systems.

4.2.1. Precision

Precision is used to measure the proportion of elements in

the R(i) set that appear in the verification set T(i), that is, how many feature terms are retrieved accurately. The precision is calculated as follows:

$$Precision = \frac{1}{n}\frac{\sum_{i=1}^{n}|R(i)\cap T(i)|}{\sum_{i=1}^{n}|R(i)|} \quad (10)$$

where $R(i) \cap T(i)$ is the number of terms retrieved related to product i; $R(i)$ is the number of terms chosen to recommend product i, denoted as N; $T(i)$ is the number of all terms contained in product documents; n is the number of on-demand printed product selected for the experiment (n =50000), and the final precision is the average of the precision for all products.

4.2.2. Recall

The recall rate characterizes the proportion of the recommended feature term set to the verification set. The recall rate is calculated as shown in Eq. (11), and the final recall rate is the average of all product recall rates.

$$Recall = \frac{1}{n}\frac{\sum_{i=1}^{n}|R(i)\cap T(i)|}{\sum_{i=1}^{n}|T(i)|} \quad (11)$$

4.3. Experimental Results and Discussion

In Figure 3, regard product node as the starting node of the random walk, the term node can only be reached after odd steps, and the product node is reached after even steps. Therefore, to achieve semantic annotation of printed products, only odd steps can be taken to recommend the feature terms for the product. In this section, we discuss the setting of restart parameters and the influence of the number of selected feature terms on ProRWR algorithm performance.

4.3.1. Effect of Restart Probability

The ProRWR algorithm will converge in the process of iteration, and the rate of convergence is determined by the restart probability c. Figure 6 shows the precision on varying c. It is obvious that precision increases slowly and then decreases slowly with the increase of c. The algorithm is most effective when c = 0.8 since the closer c is to 0, the more the random walk process can reflect the network around the starting point. The closer the c is to 1, the more the local structure of the network can be reflected.

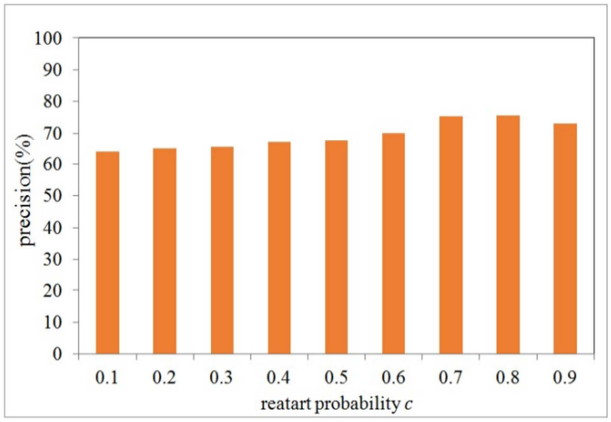

Figure 6. *Precision on varying restart probability* c.

4.3.2. Effect of Iteration Step

Figure 7 reports the precision and recall on varying iteration step l, where c=0.8. When the maximum iteration reaches, the probability that each term node is accessed tends to stable value, which suggests the convergence of ProRWR algorithm. In this article, the steady-state probability is used as the basis for recommending feature terms.

Comparing TF-IDF value, the probability value at iteration 1 is equal to the TF-IDF value multiplied by the restart probability $c(1-c)$, so the precision and recall rate are both lower. The precision and the recall have increased with l increases from 1 to 3. Since then, with the iteration step increasing, there is no obvious change in precision and recall. Therefore, the experiment finally selects the experimental results at iteration 3 for semantic annotation of on-demand printed products, which has a high accuracy, and the annotation performance is much better.

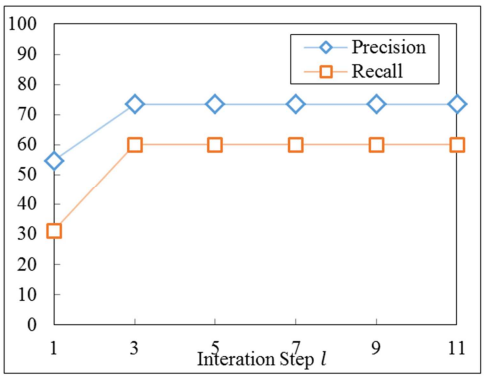

Figure 7. *Precision and recall on varying iteration step* l.

4.3.3. Effect on Number of Tags

Figure 8 shows the relationship between the precision and the recall when N takes different values, where l=3 and c=0.8. The experiment takes N= [1, 10]. Usually, we hope that the higher the precision of retrieval results, the better the recall, but in fact, the two are contradictory in some cases. For example, in Figure 8, only one term is selected, then precision is higher, but recall is very low. If all results are returned, recall is higher and precision is lower. Therefore, when the recommended number of selected tags is 4, we can get 73.5% precision and 60%recall.

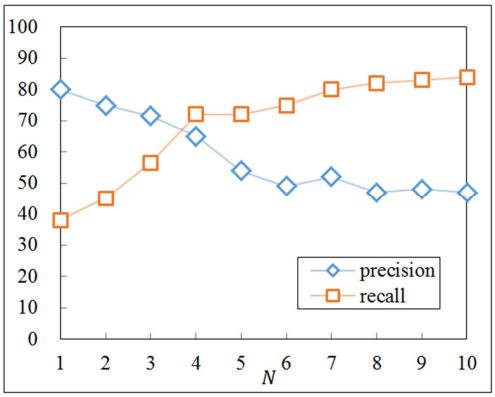

Figure 8. *Precision and recall on varying* N.

4.3.4. Case Study

ProRWR algorithm is proposed to solve the problem that users can't reasonably choose their own products because of the large number of on-demand printing products. Experimental results show that when the iteration step is 3, the restart probability is 0.8, the proposed algorithm can ensure better precision and recall, and thus maintain high precision of product annotation. Table 1 gives 3 examples of using this algorithm to annotate on-demand printed products in Amazon dataset, from which we found that most tags in the returned lists are highly relevant to the products. For example, "ballroom dancing" is the product title, the tags recommended by our algorithm is "dancing, ballroom, latin, fingerboard, slow, teach, kick, session, midnight, stars", which are related to "ballroom" and "dance" even though there is no common term in products' title. This indicates that ProRWR can discover potential terms and remove irrelevant terms. Some cases can be found in other products to prove the accuracy of our algorithm's semantic annotation. According to the feature terms recommended by ProRWR algorithm, we can see the main or similar descriptive information of the product, and then facilitate different users to find products that meet their needs.

Table 1. *Examples of on-demand printing product semantic annotation.*

Importance Ranking	swallowing the river ganges: a practice guide to the path of purification	ballroom dancing	funny teddy bear stickers dover little activity books
1	swallowing	dancing	stickers
2	river	ballroom	teddy
3	guide	latin	funny
4	path	fingerboard	bear
5	purification	slow	activity
6	practice	teach	dover
7	beaten	kick	rhyming
8	canongate	session	books
9	mood	midnight	nutcracker
10	ganges	stars	pinocchio

5. Conclusion and Future Work

This paper proposed a ProRWR algorithm based on RWR for semantic annotation of on-demand printed products. The method constructs a bipartite network based on TF-IDF algorithm to represent the relationships between products and terms, and combines RWR to discover the latent semantic association between them. Experimental results show that the proposed algorithm can solve the problem of semantic annotation in on-demand printing platform more accurately.

The contributions presented in this paper are different from existing approaches, such as keyword extraction, text categorization and TextRank. This work based on RWR has both theoretical and practical implications, which is also due to its following advantages: (1) RWR is a stable measurement standard and is not susceptible to noise and missing data. (2) It can recursively capture the multi-faceted information between two nodes in the graph. When the data is extremely sparse, it can effectively capture the transition probability between nodes and obtain the overall structural relationship of the graph. (3) RWR can effectively discover potential product tags and reduce the redundancy of tags.

There are still some limitations in our researches. (a) ProRWR can only be used in smaller networks due to memory constraints. When the number of network nodes increases sharply, the RWR-based algorithm requires a large amount of computational cost and cannot deal with more documents. (b)users want instant feedback when they get information, which does not take into account the efficiency of real-time recommendation. (c) ProRWR focuses on only the networks that are static. When network changes, ProRWR need to recalculate the relevance matrix, which has high complexity, which leads to huge computational and storage challenges.

In future work, the following aspects need to be studied to overcome the above limitations. Firstly, we will explore how to improve the RWR algorithm's ability to process large-scale data in order to cope with the expanding social network scale. Secondly, we need to speed up user query to meet user real-time response needs. In addition, the researches of dynamic graph should also be considered. In future research, the method in this paper can also be applied to commodity recommendation in e-commerce platforms, video push, and image annotation, intelligent question and answer, etc.

Acknowledgements

This work was supported by Natural Science Foundation of Shanghai grant 16ZR1422800; The 2018 Training Project for Outstanding Teaching Achievement Award of University of Shanghai for Science and Technology grant JXCGPY2018012; Bidding Program of The State Administration of Press and Publication Key Laboratory of "Green Plate Making and Standardization of Flexographic Printing" grant ZBKT201809; and The 2019 Teachers' Teaching Development Project of University of Shanghai for Science and Technology grant CFTD193006.

References

[1] Kiryakov A, Popov B, Terziev I, et al. "Semantic annotation, indexing, and retrieval". Web Semantics: Science, Services and Agents on the World Wide Web, Vol.2, No. 1, pp. 49-79, 2004.

[2] Yu M, Han X, Gou X, et al. “Content-based social network user interest tag extraction”. International Journal of Database Theory and Application, Vol. 8, No. 2, pp. 107-118, 2015.

[3] Jain S, Khangarot H, Singh S. “Journal Recommendation System Using Content-Based Filtering”. Recent Developments in Machine Learning and Data Analytics. Springer, Singapore, 2019: 99-108.

[4] Lu Kai, Zhang Guanyuan, Wan Bin. “CICF: a context information based collaborative filtering algorithm”. Journal of Chinese Information Processing, Vol. 28, No. 2, pp. 122-128, 2014.

[5] Canito A, Marreiros G, Corchado J M. “Automatic Document Annotation with Data Mining Algorithms”. World Conference on Information Systems and Technologies. Springer, Cham, 2019, pp. 68-76.

[6] Zhao W X, Wang J, He Y, et al. “Mining product adopter information from online reviews for improving product recommendation”. ACM Transactions on Knowledge Discovery from Data (TKDD), Vol. 10, No. 3, pp. 29, 2016.

[7] Bandyopadhyay S, Thakur S S, Mandal J K. “Product Recommendation for E-Commerce Data Using Association Rule and Apriori Algorithm”. International Conference on Modelling and Simulation, Springer, Cham, pp. 585-593, 2017.

[8] Gao G, Liu Y S, Lin P, et al. “BIMTag: Concept-based automatic semantic annotation of online BIM product resources”. Advanced Engineering Informatics, Vol. 31, pp. 48-61, 2017.

[9] Verma Y, Jawahar C V. “Image annotation by propagating labels from semantic neighbourhoods”. International Journal of Computer Vision, Vol. 121, No. 1, pp. 126-148, 2017.

[10] Halder A, Dobe O.“Detection of tumor in brain MRI using fuzzy feature selection and support vector machine”. 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, 2016, pp. 1919-1923.

[11] Gupta V, Karnick H, Bansal A, et al. “Product classification in e-commerce using distributional semantics”. arXiv preprint arXiv:1606.06083, 2016.

[12] Ravale U, Marathe N, Padiya P. “Feature selection based hybrid anomaly intrusion detection system using K means and RBF kernel function”. Procedia Computer Science, Vol. 45, pp. 428-435, 2015.

[13] Pang L, Lan Y, Guo J, et al. “Text matching as image recognition”. Thirtieth AAAI Conference on Artificial Intelligence, 2016.

[14] Yih W T, Goodman J, Carvalho V R. “Finding advertising keywords on web pages”. International Conference on World Wide Web, DBLP, pp. 213, 2006.

[15] Matsuo Y, Ishizuka M. “Keyword Extraction from a Single Document using Word Co-occurrence Statistical Information”. International Journal on Artificial Intelligence Tools, Vol. 13, No. 01, pp. 157-169, 2008.

[16] Jung J. “Random walk with restart on large graphs using block elimination”. ACM Transactions on Database Systems, Vol. 41, No. 2, pp. 1-43, 2016.

[17] Jung J, Park N, Lee S, et al. “BepI: Fast and memory-efficient method for billion-scale random walk with restart”. Proceedings of the 2017 ACM International Conference on Management of Data, ACM, pp. 789-804, 2017.

[18] Zhou Y, Cheng H, Yu J X. “Graph clustering based on structural/attribute similarities”. Proceedings of the VLDB Endowment, Vol. 2, No. 1, pp. 718-729, 2009.

[19] Zhang M L, Zhou Z H. “A k-nearest neighbor based algorithm for multi-label classification”. GrC, Vol. 5, pp. 718-721, 2005.

[20] Hollocou A, Bonald T, Lelarge M. “Multiple local community detection”. ACM SIGMETRICS Performance Evaluation Review, Vol. 45, No. 3, pp. 76-83, 2018.

[21] Zhiyuli A, Liang X, Chen Y. “HSEM: highly scalable node embedding for link prediction in very large-scale social networks”. World Wide Web, 2018, pp. 1-26.

[22] Ahmed R, Baali I, Erten C, et al. “MEXCOWalk: Mutual Exclusion and Coverage Based Random Walk to Identify Cancer Modules”. bioRxiv, 2019, pp. 547653.

[23] Han C, Luo Z, Gu W, et al. “A Random Walk Tensor Model for Heterogeneous Network Entity Classification”. IEEE Access, 2019.

[24] Zhang J, Tao T, Mu Y, et al. “Web image annotation based on Tri-relational Graph and semantic context analysis”. Engineering Applications of Artificial Intelligence, Vol. 81, pp. 313-322, 2019.

[25] Tong, Hanghang, Faloutsos, et al. “Fast Random Walk with Restart and Its Applications”. International Conference on Data Mining, pp. 613-622, 2006.

[26] Yu W. “Reverse Top-k Search Using Random Walk with Restart”. PVLDB, Vol. 7, No. 5, pp. 401–412, 2014.

[27] Zhou Y, Cheng H, Yu J X. “Graph Clustering Based on Structural/Attribute Similarities”. PVLDB, Vol. 2, No. 1, pp. 718-729, 2009.

[28] Tong H, Faloutsos C. “Center-piece subgraphs: problem definition and fast solutions”. Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, ACM, pp.404-413, 2006.

[29] Jung J, Jin W, Sael L, et al. “Personalized ranking in signed networks using signed random walk with restart”. 2016 IEEE 16th International Conference on Data Mining (ICDM), IEEE, pp. 973-978, 2016.

[30] Wang S, Tang Y, Xiao X, et al. “HubPPR: effective indexing for approximate personalized pagerank”. Proceedings of the VLDB Endowment, Vol. 10, No. 3, pp. 205-216, 2016.

[31] Yu W, McCann J. “Random walk with restart over dynamic graphs”. 2016 IEEE 16th International Conference on Data Mining (ICDM), IEEE, pp. 589-598, 2016.

[32] Yoon M, Jin W, Kang U. “Fast and accurate random walk with restart on dynamic graphs with guarantees”. Proceedings of the 2018 World Wide Web Conference on World Wide Web, International World Wide Web Conferences Steering Committee, pp. 409-418, 2018.

[33] Guo A, Yang T. “Research and improvement of feature words weight based on TFIDF algorithm”. 2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference, IEEE, pp. 415-419, 2016.

[34] Yin L. “Chinese Keyword Extraction Based on Weighted Complex Network”. International Conference on Intelligent Systems and Knowledge Engineering, Vol. 12, pp. 1-5, 2017.

[35] Li Y, Shen B. “Research on sentiment analysis of microblogging based on LSA and TF-IDF”. 2017 3rd IEEE International Conference on Computer and Communications (ICCC), IEEE, pp. 2584-2588, 2017.