News Curation Service using Semantic Graph Matching

Ryohei Yokoo, Takahiro Kawamura, Yuichi Sei, Yasuyuki Tahara, Akihiko Ohsuga

Graduate School of Information Systems
University of Electro-Communications
Tokyo, Japan
Email: {r-yokoh,kawamura}@ohsuga.is.uec.ac.jp, {sei,tahara}@is.uec.ac.jp, ohsuga@uec.ac.jp

***Abstract*—In recent years, "News Curation Services" that recommend news articles on the Internet to users are getting attention. In this paper, we propose a news curation service that collects and recommends "news articles" that users feel interested by using semantic relationships between terms in the articles. We define "interested" news articles as articles that users have curiosity and serendipity. The semantic relations between events terms are represented by Linked Data. We create News Articles Linked Data (candidates for recommendation to users) and User's preferences Linked Data (users' preferences). In order to recommend news articles to users, we first search common subgraphs between two kinds of Linked Data. The experiment showed that the curiosity score is 3.30 (min:0, max:4), and the serendipity score is 2.93 in our approach, although a baseline method showed the curiosity score is 3.03, and the serendipity score is 2.79. Thus, we confirmed that our approach is more effective than the baseline method.**

Keywords*–*Semantic Relation; Linked Data; News Recommendation.

I. INTRODUCTION

Recently, web services, such as paper.li [1] and The Tweeted Times [2] that automatically gather news articles and recommend to users have been popular. The users can easily get interested information by those services called "News Curation Services". In this paper, we propose a semantic graph application for "News Curation Services", which recommends interested news articles according to users' preferences. We define "interested news articles" as articles that user has curiosity and serendipity. A lot of content-based recommendation approaches, such as tf-idf use only words or terms in news articles for features of recommendation. In contrast, our approach applies semantic relation between the terms as the features. Thus, our contribution is that we extract users' preferences more accurately than other approaches, and then recommend news articles to the users. The semantic relations between terms are represented in Linked Data.

We create two kinds of Linked Data in this paper. First, we create News Articles Linked Data, composed of sentences of news articles, which are candidates for recommendation to users. Next, we create users' preferences Linked Data, composed of sentences of news articles that users feel interested. In order to recommend news articles to users, we search news articles by finding common subgraphs, that is, triples like term-relation-term between two kinds of Linked Data. If there is a common subgraph. we recommend news articles, which are associated with the subgraph in News Articles Linked Data to the users.

The remainder of the paper is organized as follows. Section II describes related works, and Section III describes our approach. In Section IV, we show experiments and evaluation. Finally, we conclude this paper with discussion and the future work in Section V.

II. RELATED WORK

Most of previous studies for recommendation systems based on contents have applied terms in sentences [3][4]. These recommendation systems need Bag-of-Words vectors as features. They recommend contents with frequent terms in text that users feel interested.

Capelle et al. [5] studied content-based recommendation system, which focused on terms semantics. They developed a system by applying similar terms for news articles that users already read or not. The similarity of terms was calculated by WordNet and a search engine Bing.

There is also a study for constructing Linked Data from news articles. Radinsky et al. [6] extracted news topics from sentences in news article titles for 150 years, and then constructed News Linked Data with causal relationships. Then, they tried to expect future events by tracing the Linked Data.

Ohsawa et al. [7] proposed a method for expecting for the number of "Like" in Facebook pages. They applied the information in DBpedia and made the expectation model with words similarities between Facebook pages.

As recommendation systems by using Linked Data, Khrouf et al. [8] targeted event information. They converted meta information on the event news sites, such as location, time, genre and so on to Linked Data, and recommended the event information to users. The information is searched by a hybrid approach of similarities of events' structures and a collaborative filtering technique.

Moreover, Mirizzi et al. [9] have applied movie information in DBpedia to Vector-Space-model, and recommended movies, which users feel interested by similarity of movie information, such as genre, director and actor, etc.

Elahi et al. [10] proposed a picture recommendation system with DBpedia infomation.

Passant et al. [11] showed a musician recommendation system by information about musicians in DBpedia. They proposed a method for measuring semantic similarity between Linked Data as Linked Data Semantic Distance (LDSD), and then this method is applied to a lot of recommendation systems with Linked Data.

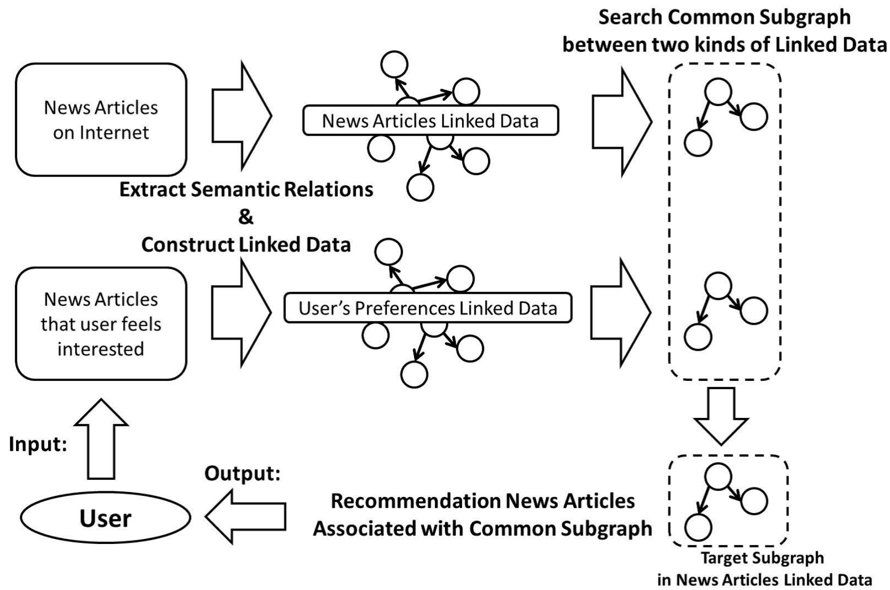

Figure 1. FLOW OF OUR APPROACH.

On the other hand, we put specific labels on terms in the text as Semantic Role Labeling [12] to extract semantic relations of the text, and then convert them to Linked Data. WordNet and VerbNet are used in Semantic Role Labeling.

There are many recommendation systems based on contents and Linked Data. However, to the best of our knowledge, there is no news recommendation system by using semantic relations in Linked Data.

III. Proposed Approach

We recommend news articles to users by using semantic relations of terms, since we assume that some news articles that users prefer, indicates the users' interest. Thus, we discuss how to extract the semantic relations and to recommend news articles to users in this section. Figure 1 indicates a flow of our approach.

First, we collect news articles, that users indicated obvious interest from social bookmark sites and others, and then extract the semantic relations from the articles. The semantic relations are combinations of terms with their relations in each sentence of the articles. We assume these semantic relations include users' preferences, and we construct User's Preferences Linked Data.

Next, we crawl a large amount of news articles on the Internet, and extract semantic relations as well, and then construct News Articles Linked Data.

In order to recommend the news articles to the users, we search common subgraphs between User's Preferences Linked Data and News Articles Linked Data. At this time, we also apply an "Entity Linking technique" for matching the terms (nodes of graph). Finally, we recommend the news articles associated with the subgraph in News Articles Linked Data to the users.

In details, the extraction of semantic relations of news articles is described in Section III-A. Section III-B describes how to find common subgraphs between two kinds of Linked Data. Then, we show the technique of Entity Linking in Section III-C.

A. Construction of Linked Data

1) Definition of Semantic Relation: Semantic relations are extracted from each sentence of news articles. In our previous work, Nguyen et al. [13] extracted behavioral properties from Web pages and Tweets to acquire users' behavioral information in a specific event like a disaster. They defined event's properties as Who, Action, What, When, Where, and so on. However, we aim to recommend news articles to users, and thus semantic relations must be simple in order to increase recommendation results. Therefore, we newly defined six new properties in this paper as follows.

- Subject (subject of an event)
- Activity (activity of an event)
- Object (object of an activity)
- Date (date an event occurred)
- Time (time an event occurred)
- Location (where an event occurred)

For example, if a news article has a sentence "Keisuke Honda has been elected to the Worst Eleven in Serie A May 21, 2014", its semantic relations are represented in Linked Data like Figure 2. Our semantic relations are composed of multiple triples, which connect terms in the sentence. The triple is a meta-deta model, which represents the relationship between two resources with a property "resource $\rightarrow$ (property) $\rightarrow$ resource". In this case, triples are "elected $\rightarrow$ (Object) $\rightarrow$ Worst Eleven" and "Keisuke Honda $\rightarrow$ (Activity) $\rightarrow$ elected". Note that, a semantic relation between Subject and Activity is represented as "Subject's term $\rightarrow$ (Activity) $\rightarrow$ Activity's term", although other relations are represented as "Activity's term $\rightarrow$ (property) $\rightarrow$ term".

 ISBN: 978-1-61208-420-6

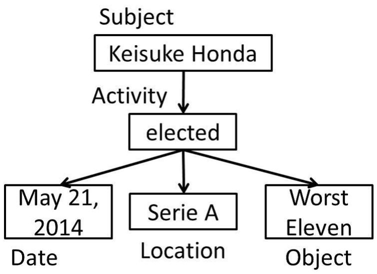

Figure 2. EXAMPLE OF SEMANTIC RELATION.

Terms	Dependency Info.	POS Info.	Labels
Keisuke	1	Noun	B-Subject
Honda	1	Noun	I-Subject
has	2	Auxiliary verb	
been	2	Verb	
elected	2	Verb	B-Activity
to	2	Adposition	
the	3	Article	
Worst	3	Noun	B-Object
Eleven	3	Noun	I-Object

Figure 3. EXAMPLE OF MANUAL LABELING.

2) Pre-processing: In this paper, we adopted Japanese news articles as sources. Then, parentheses frequently appear in news articles and dazzle someone's eyes. Also, they make semantic relations in a sentence more difficult. Therefore, we removed the parentheses in pre-process steps. We first split a sentence with the parentheses to string outside the parentheses and string inside parentheses to simplify the sentence. But, our previous work showed the string inside parentheses are often useless, and thus we deleted them all.

Also, we registered 7,572 locations in Japan and 150,90,897 titles of all Japanese Wikipedia articles as of December, 2014 in our dictionary.

3) Semantic Role Labeling with CRF: In order to extract the semantic relations from news articles, we apply Conditional Random Field (CRF) [14]. CRF is a machine learning technique to solve sequential labeling problems. CRF has been used in morphological analysis, part-of-speech (POS) tagging, named entity recognition [15], and group activity recognition [16], etc.

First, we extract dependency information between terms, and POS information of a sentence, and then convert them to a feature vector format for CRF. We get the dependency information from Cabocha [17], and the POS information from Mecab [18].

As a training dataset for CRF learning phase, we used sentences manually labeled in advance. Figure 3 shows an example of training data, "Keisuke Honda has been elected to the Worst Eleven". I is internal of a chunk, B shows beginning of the chunk. In estimation phase, we use a CRF's model constructed by the training data, and automatically put properties to each sentence.

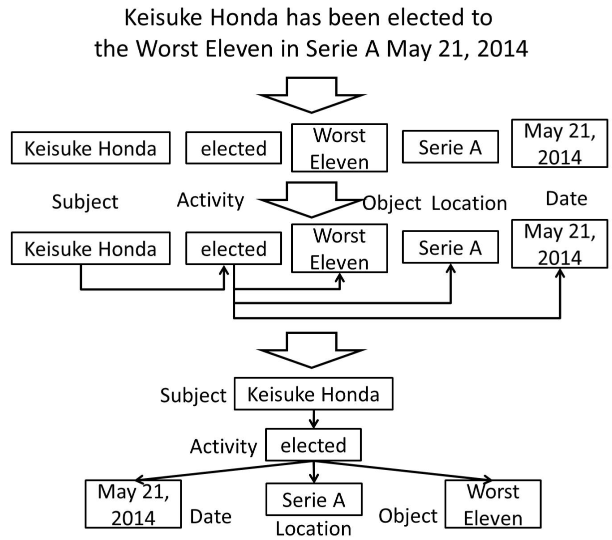

Figure 4. PROCEDURES OF SEMANTIC RELATION EXTRACTION.

As a preliminary experiment, we collected 98 sentences from 13 news articles in Japanese for our the training dataset. The articles are collected from Japanese news site, Asahi.co.jp [19] on Oct. 3, 2014. Details of the dataset are shown in Table I. We then tried to estimate properties (labels) in test sentences, but the accuracy by 10-fold cross-validations was not enough when applying CRF as it is. Especially, Subject, Time, and Location indicate low accuracies. Hence, we devised some heuristics for Time and Location. The heuristic rules are executed on the CRF results based on the dependency and POS information.

As a result, Table II shows the average accuracies for each labels become more than 80%. "Weighted Average" means the average accuracy for all labels.

4) Construction of Semantic Relation: In Figure 4, we show how to construct the semantic relations from the labeled sentences. The figure indicates a procedures of semantic relation construction from a sentence"Keisuke Honda has been elected to the Worst Eleven in Serie A May 21, 201". First, we extract the labeled terms in the sentence. Then, we gather the terms for each semantic relation using dependency information. Finally, we connect these terms with semantic relations, and then convert it to Linked Data in Resource Description Framework (RDF).

B. Recommendation of News Articles using Common Subgraph

In order to recommend news articles to users, we search "common subgraphs" between News Articles Linked Data and User's Preferences Linked Data. We define "subgraphs" in Linked Data as one or more linked triples. We find common subgraphs by finding at least a common triple between two kinds of Linked Data. Common triples need common Subject, Value and Property between two triples, and thus we first try to find them for searching common subgraphs. Then, we get news articles associated with the common subgraphs in News Articles Linked Data.

We show an example of the common subgraph in Figure 5. The subgraph "Keisuke Honda (Subject) ← Activity ←

TABLE I. SUMMARY OF TRAINING DATA

sentences	terms	all labels	Subject	Activity	Object	Date	Time	Location
98	2,479	1,888	265	718	754	79	37	35

TABLE II. ACCURACY OF LABELING

	Subject	Activity	Object	Date	Time	Location	Weighted Average
Precision	67.80%	91.22%	87.41%	81.23%	82.46%	97.77%	86.48%
Recall	85.61%	87.20%	82.22%	90.03%	87.50%	85.71%	86.59%
F-measure	75.67%	89.16%	84.74%	85.40%	84.90%	91.34%	86.53%

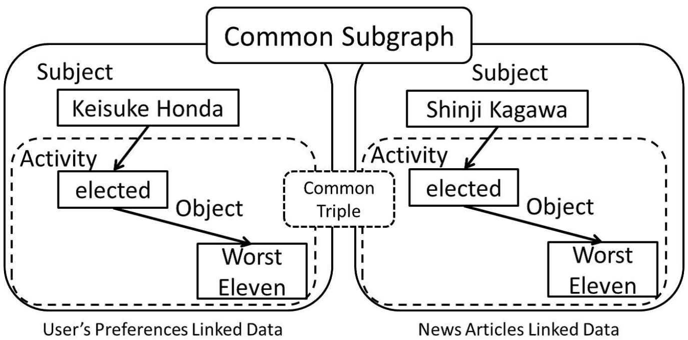

Figure 5. EXAMPLE OF COMMON SUBGRAPH.

elected (Activity) ← Object ← Worst Eleven (Object)" in User's Preferences Linked Data was extracted from a sentence "Keisuke Honda has been elected to the Worst Eleven". Similarly, the subgraph "Shinji Kagawa (Subject) ← Activity ← elected (Activity) ← Object ← Worst Eleven (Object)" in News Articles Linked Data was extracted from a sentence "Keisuke Honda has been elected to the Worst Eleven". These subgraphs have a common triple "elected (Activity) ← Object ← Worst Eleven (Object)", and so this corresponds to a common subgraph. Moreover, each subgraph has a partial match "x (Subject) ← Activity ← elected (Activity)" linked to the common triple. Therefore, we recommend a news article associated with the subgraph "Shinji Kagawa (Subject) ← Activity ← elected (Activity) ← Object ← Worst Eleven (Object)" to users.

Figure 6 shows an algorithm for searching common subgraphs between two kinds of Linked Data. Inputs are *UserGraph* and *NewsGraph*. *UserGraph* is a set of triples in User's Preferences Linked Data. Similarly, *NewsGraph* indicates a set of triples in News Articles Linked Data. First, we check whether *user_triple* and *news_triple* have a common triple or not by using *SIMTRIPLE*. Details of *SIMTRIPLE* are shown in Figure 7. If these triple are determined as a common triple, we get other triples include terms (Subject or Value) of each triple. Then, we search triples that have a common Property between *u_graph* and *n_graph* by *PartialMatch*. In addition, we gather them to *X*. Outputs are common subgraphs in News Articles Linked Data that are linked to *n_triple* and *x*. Finally, we recommend news articles associated with the common subgraphs.

In our approach, we can collect common subgraphs not only in the case that we were able to entirely extract semantic relations in a news articles but also the cases that we partially extracted the semantic relations. We use subgraphs for matching, which have at least two triples with two properties and three nodes. Therefore, common subgraph search in our approach works with news article if the extracted semantic relations have at least two linked triples.

However, the defined schema for Linked Data has Activity as a hub as shown in Figure 2. Therefore, the common subgraph search cannot work if the semantic relations do not include Activity's terms.

C. Entity Linking

In order to search common subgraphs between User's Preferences Linked Data and News Articles Linked data, the common subgraphs need the same Subject, Property, and Value. However, the number of common subgraphs is very little if we search the common subgraphs with exact matching of terms. Also, this causes to miss an opportunity to find similar subgraphs, and thus leads to a matter of no recommendation.

Therefore, we apply "Entity Linking" for common subgraphs search. Entity Linking is a task for searching common terms by applying synonyms of Entity (terms) in sentences. Entity Linking usually needs an expression dictionary, and a similarity measure between terms. For example, if there is a sentence includes "be elected to the Worst Eleven", "be elected" has the same meaning as "be chosen", and"elected", etc. We get much more common subgraphs than the exact matching by applying such an Entity Linking technique. Study of Bunnescu et al. [20] is a pioneer of Entity Linking. Bunnescu has proposed a method for resolving the word-sense disambiguation by using hyperlink structure between articles of Wikipedia. Also, Hoffart [21] developed an Entity Linking framework AIDA for named entity extraction and word-sense disambiguation. Hoffart's Entity Linking is similarity calculation for terms by using contexts in sentences.

In this paper, we applied Jaccard index and Japanese WordNet for similarity calculation. Jaccard index is a string matching techniques. Equation (1) indicates a formula for Jaccard index, which represents a ratio of common elements of the two sets: A and B. Here, we calculate a similarity score between the two terms by using their surfaces. Inputs are two terms and output indicates a similarity score between [0-1]. If the score is 1, A and B are matched exactly. We a set threshold score of Jaccard index as 0.5.

$$Jaccard(A, B) = \frac{A \cap B}{A \cup B} \tag{1}$$

Algorithm 1 SEARCH COMMON SUBGRAPH

```
Input: UserGraph, NewsGraph
Output: All_Subgraph
function PARTIALSEARCH(u_graph, n_graph)
    for all u_triple ∈ u_graph do
        for all n_triple ∈ n_graph do
            if PARTIALMATCH(u_triple, n_triple) then
                Push n_triple into array X
            end if
        end for
    end for
    return X
end function

function COLLECTSUBGRAPH(news_triple, X)
    for all x ∈ X do
        Push news_triple + x
                into array Subgraphs
    end for
    return Subgraphs
end function

for all user_triple ∈ UserGraph do
    for all news_triple ∈ NewsGraph do
        if SIMTRIPLE(user_triple, news_triple) then
            u_graph ← CollectGraph(user_triple)
            n_graph ← CollectGraph(news_triple)
            X ← PARTIALSEARCH(u_graph,n_graph)
            Push COLLECTSUBGRAPH(news_triple, X)
                    into array All_Subgraphs
        end if
    end for
end for
return All_Subgraphs
```

Figure 6. SEARCH COMMON SUBGRAPH ALGORITHM.

By applying Jaccard index, we can determine that “elected” and “elect” are identical. However, “elected” and “chosen” are not solved only by Jaccard index. Therefore, we also applied WordNet, and search similar terms for covering a string matching’s weak point.

We show a method for searching common triples with Entity Liking in Figure 7 (*SIMTRIPLE* in Figure 6). Inputs are a triple in User’s Preferences Linked Data *u_triple* and a triple in News Articles Linked Data *n_triple*. Note that the triples must have the same Property. Thus, we calculate terms similarity of Subject terms (*u_triple.subject* and *n_triple.subject*) and Value terms (*u_triple.value* and *n_triple.value*) in the order of exact match, WordNet, and Jaccard index.

IV. EXPERIMENT

We recommended “interested” news articles to test users with our approach. We define “interested” means curiosity and serendipity. Therefore, we set as metrics “curiosity”, “serendipity”, and “relevance” (similarity) as reference information.

A. Dataset

In order to construct News Articles Linked Data, we applied 21,105 news articles from Oct. 4, 2014 to Jan. 10, 2015. It took about an hour to construct the Linked Data with

Algorithm 2 SEARCH TRIPLE

```
Input: u_triple, n_triple
Output: Bool
function SIMWORDS(u_word, n_word)
    if u_word == n_word then
        return True
    end if
    if WORDNET(u_word, n_word) then
        return True
    end if
    if JACCARD(u_word, n_word) ≥ 0.5 then
        return True
    end if
    return False
end function

function SIMTRIPLE(u_triple, n_triple)
    if u_triple.property != n_triple.property then
        return False
    end if
    if SIMWORDS(u_triple.subject, n_triple.subject)
then
        if SIMWORDS(u_triple.value, n_triple.value)
then
            return True
        end if
    end if
    return False
end function
```

Figure 7. SEARCH TRIPLE ALGORITHM.

the articles. Similarly, we applied 1,471 news articles from Jan. 11, 2015 to Jan. 19, 2015 for User’s Preferences Linked Data, which was constructed in a few minutes. The news articles that construct both Linked Data are collected from Japanese news site, Asahi.co.jp. A summary of our dataset for News Articles Linked Data is shown in Table III, and a summary for User’s Preferences Linked Data is shown in Table IV.

B. Experimental Setting

We found 978 common subgraphs from the two datasets. These subgraphs were found between 142 news articles in User’s Preferences Linked Data and 578 news articles in News Articles Linked Data. Thus, 578 news articles associated with the common subgraphs could be recommended to test users. The calculation time is about 3,577 sec. However, checking a large number of articles is almost impractical for test users. Therefore, in order to reduce the news articles, we excluded the following common subgraphs.

- Properties in common triples are Date and Time.
- Number of terms in common triples is 2 or less.
- Common triple’s terms indicate tense alone.
- Common triple includes only short length terms.

The number of the reduced common subgraph was 166. These common subgraphs are composed of 62 news articles in User’s Preferences Linked Data and 126 news articles in News Articles Linked Data. Thus, we used 62 news articles in User’s Preferences Linked Data for evaluation. There were some news

TABLE III. DATASET FOR NEWS ARTICLES LINKED DATA

Articles	Nodes	Labels	Subject	Activity	Object	Date	Time	Location
21,105	42,890	44,869	10,892	12,040	17,994	1,761	749	1,433

TABLE IV. DATASET FOR USER'S PREFERENCES LINKED DATA

Articles	Nodes	Labels	Subject	Activity	Object	Date	Time	Location
1,471	4,526	4,617	1,612	1,612	1,548	172	84	117

TABLE V. EXPERIMENTAL RESULT

	relevance	curiosity	serendipity
Our Approach	3.06	3.30	2.93
Baseline	3.22	3.03	2.79

TABLE VI. PERFORMANCE OF RECOMMEND NEWS

	good	excellent
Our Approach	2.55	1.15
Baseline	2.40	0.65

articles, which can recommend multiple news articles to users. But we recommended a news article from a news article that users feel interested. A news article is selected based on the similarities of triples. If the similarities are the same, we randomly chose a news article.

C. Experiment Procedure

We asked the test user to determine whether or not the recommended articles are relevant (similar to), an article that the user feels interested, and has curiosity, serendipity.

We defined the interesting articles, which users that users get attracted to and make discovery from. We then regarded the articles, which users get highly attracted as the curiosity articles, and the articles, which users make an important discovery as the serendipity articles. There are 20 test users, in which 13 test users are our university students. The test users answered in 4 levels: "I think so", "I think so a little", "I don't think so a little", and "I don't think so". We also conducted comparison with a baseline method using tf-idf. The method is the most famous approach for extracting feature words of sentences and it has been used for a lot of studies [22][23]. It needs term (word) frequency as tf and inverse Document Frequency as idf for calculating weights of the words. We extracted top three weighted words from an article that test users feel interested. All three words are nouns. The baseline method searches a news article contains those three words from dataset for News Articles Linked Data, and then recommends the news articles to each test user.

D. Evaluation

Table V indicates the average scores of our approach and the baseline method. Our approach showed relevance:3.06, curiosity:3.30, and serendipity:2.93 in average. In contrast, the baseline method showed relevance:3.22, curiosity:3.03, serendipity:2.79. As a result, the curiosity and the serendipity score of our approach were higher than the baseline method, although the relevance is lower than the baseline.

The reason why the baseline had a high relevance score was that the baseline method recommended news articles, which include three frequent nouns. However, semantic relations in our approach include terms of noun, verb and adjective, and so on. As a result, the baseline method directly retrieved topics represented in nouns of news articles. Our common subgraphs include several terms, which are not directly relevant to the news articles that users feel interested. However, these terms have the same semantic relations from a certain topic terms as in the news articles the users feel interested. In a sense, we believe that these *variables* contributed to raise the curiosity and the serendipity score, decreasing the relevance score.

Finally, we checked how many "interested" news articles were recommended to the users. If a test user determined that the curiosity and serendipity score are more than 3, we counted the news article as "good". Then, if a test user answered that both scores are 4 , we counted the new article as "excellent". We show the result in Table VI. Our approach and the baseline method are almost the same in "good", but our approach recommended more "excellent" news articles than the baseline method. We thus confirmed that, our approach is superior to the conventional content-based method.

V. Conclusion and Future Work

In this paper, we proposed a new "News Curation Service" by using semantic relations in news articles. Semantic relations are represented as Linked Data. We proposed an approach for constructing Linked Data from news articles and recommend news articles to users based on common subgraphs between User's Preferences Linked Data and News Articles Linked Data. Through the experiments, we confirmed our approach can incorporate more users' interest than the existing approach.

In the future work, we will improve accuracy of the CRF labeling and Entity Linking. In addition, we will examine patterns of common subgraphs for news recommendation. Also, we reconstruct our Linked Data schema to find more common subgraphs between two kinds of Linked Data.

Acknowledgment

This work was supported by JSPS KAKENHI grant Numbers 24300005, 26330081, 26870201.

References

[1] Paper.li team: "Paper.li – Be a publisher", http://papper.li, 2015.06.11.

[2] Tweeted Times team: "The Tweeted Times | Content curation and publishing", http://tweetedtimes.com, 2015.06.11.

[3] W. Lee, K. Oh, C. Lim, and H. Choi: "User profile extraction from Twitter for personalized news recommendation", Proceedings of the 16th Advanced Communication Technology, 2014, pp. 779-783.

[4] W. IJntema, F. Goossen, F. Frasincar, and F. Hogenboom: "Ontology-based News Recommendation", Proceedings of the 2010 EDBT/ICDT Workshops, 2010, pp. 16:1-16:6.

[5] M. Capelle, F. Hogenboom, and A. Hogenboom: "Semantic News Recommendation Using WordNet and Bing Similarities", Proceedings of the 28th Annual ACM Symposium on Applied Computing, 2013, pp. 296-302.

[6] K. Radinsky, S. Davidovich, and S. Markovitch: "Learning causality for news events prediction". Proceedings of the 15th international conference on World Wide Web, 2012, pp. 909-918.

[7] S. Ohsawa and Y. Matsuo: Like Prediction: "Modeling Like Counts by Bridging Facebook Pages with Linked Data". Proceedings of the 22Nd International Conference on World Wide Web Companion, 2013, pp. 541-548.

[8] H. Khrouf and R. Troncy: "Hybrid event recommendation using linked data and user diversity", Proceedings of the 7th ACM conference on Recommender systems, 2013, pp. 185-192.

[9] R. Mirizzi, T. D. Noia, A. Ragone, V. C. Ostuni, and E. D. Sciascio: "Movie Recommendations with DBpedia", IIR, volume 835 of CEUR Workshop Proceedings, 2012, pp. 101-112.

[10] N. Elahi, R. Karlsen, and E. J. Holsb?: "Personalized Photo Recommendation By Leveraging User Modeling On Social Network". Proceedings of International Conference on Information Integration and Web-based Applications, 2013, pp. 68-71.

[11] A. Passant: "dbrec: music recommendations using DBpedia", ISWC'10 Proceedings of the 9th International Semantic Web Conference on The Semantic Web - Volume Part II, 2010, pp. 209-224.

[12] Y. Matsubayashi, N. Okazaki, and J. Tsujii: "Generalization of Semantic Roles in Automatic Semantic Role Labeling", Information and Media Technologies, 2014, pp. 736-770.

[13] T. M. Nguyen, T. Kawamura, Y. Tahara, and A. Ohsuga: "Self-supervised capturing of users´ activities from weblogs", International Journal of Intelligent Information and Database Systems, Vol.6, No.1, 2012, pp. 61-76.

[14] J. Lafferty, A. McCallum, and F. C. N. Pereira: "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data", ICML '01 Proceedings of the Eighteenth International Conference on Machine Learning, 2001, pp. 282-289.

[15] G. Zhu, T. J. Bethea, and V. Krishna: "Extracting Relevant Named Entities for Automated Expense Reimbursement", Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2007, pp. 1004-1012.

[16] T. Kaneko, M. Shimosaka, S. Odashima, R. Fukui, and T. Sato: "Consistent collective activity recognition with fully connected CRFs", Proceedings of the 21st International Conference on Pattern Recognition, 2012, pp. 2792-2795.

[17] Taku Kudo: "CaboCha: Yet Another Japanese Dependency Structure Analyzer", http://taku910.github.io/cabocha/, 2015.06.11.

[18] Taku Kudo: "MeCab: Yet Another Part-of-Speech and Morphological Analyzer", http://taku910.github.io/mecab/, 2015.06.11.

[19] The Asahi Shimbun Company: "Asahi Shinbun Degital: The news cite of Asahi", http://asahi.com, 2015.06.11.

[20] R. Bunnescu and M. Pasca: "Using Encyclopedic Knowledge for Named Entity Disambiguation". Proceesings of the 11th Conference of the European Chapter of the Association for Computational Linguistics, 2006, pp. 9-16.

[21] J. Hoffart et al.: "Robust disambiguation of named entities in text". Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2011, pp. 782-792.

[22] J. H. Paik: "A novel tf-idf weighting scheme for effective ranking". In Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '13, 2012, pp. 343-352.

[23] L. F. S. Teixeira, G. P. Lopes, and R. A. Ribeiro: "An extensive comparison of metrics for automatic extraction of key terms". In Joaquim Filipe and Ana L. N. Fred, editors, ICAART 2012, 2012, pp. 55-63.

 ISBN: 978-1-61208-420-6