HAL open science

Semantic-Based Recommendation Method for Sport News Aggregation System

Quang-Minh Nguyen, Thanh-Tam Nguyen, Tuan-Dung Cao

To cite this version:

Quang-Minh Nguyen, Thanh-Tam Nguyen, Tuan-Dung Cao. Semantic-Based Recommendation Method for Sport News Aggregation System. 10th International Conference on Research and Practical Issues of Enterprise Information Systems (CONFENIS), Dec 2016, Vienna, Austria. pp.32-47, 10.1007/978-3-319-49944-4_3 . hal-01630538

HAL Id: hal-01630538

https://inria.hal.science/hal-01630538v1

Submitted on 7 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Semantic-based Recommendation Method For Sport News Aggregation System

Quang-Minh Nguyen[1], Thanh-Tam Nguyen[1], Tuan-Dung Cao[1]

[1] Hanoi University of Science and Technology, 1 Dai Co Viet, Vietnam
minh.nguyenquang@hust.edu.vn, mrtamb9@gmail.com, dungct@soict.hust.edu.vn

Abstract. News on the Internet today plays an important role in helping people access daily information around the world. News aggregators are websites that collect and provide content from different sources in one location for easy viewing. However, the increasing number of news on the Internet makes it difficult for readers when they desire to access news they are concerned. One solution to this issue is based on employing recommender systems. In this research, we propose a novel method for news recommendation based on a combination of semantic similarity with content similarity between news and implement it as a feature of semantic-based news aggregators BKSport. Experimental results have shown that, a combination of both kind of similarity measures will result in better recommendation than when using either measure separately.

1. Introduction

The development of the Internet has brought a sharp increase in the number of news websites and the Web becomes a popular platform for broadcasting news. News aggregators are websites that collect news from various sources and provide an aggregated view of the events taking place in all over the world. Unfortunately, a critical issue of news aggregation systems is that large number of daily published news obstructs readers when they want to find the ones relevant to their particular interests. A possible solution to this problem is the use of recommender systems as they can traverse the space of choices and predict the potential usefulness of news for each reader.

There have been many researches on news recommendation methods which are based on a certain similarity measure, probably similarity between news with each other, known as *Global Recommendation System (GRS)*, or similarity between personal interests of readers and news, known as *Personal Recommendation System (PRS)* [2, 5]. In *GRS*, news recommended are news with the highest similarity with news that readers are reading. On the other hand in *PRS*, news recommended for readers are news with the highest similarity with personal interests of readers, which is modeled based on the history of posts that readers have read. Collaborative filtering (CF) is a widely applied technology in PRS development. With explosion of news on the Web, designing novel approach for effective new recommendation to suggest news closer and more relevant to readers is still a matter of concern. In this research, we focus on proposing a news recommendation method according to just *global recommendation system* model by enhancing results from existing works.

The most important task in developing *GRS* systems is to build a model to calculate similarity between news. Recent research works on news similarity measuring center on two prominent approaches: *content-based similarity* and *semantic-based similarity*. In content-based approach, similarity of news is calculated based on vocabulary statistics appeared in content of news and almost all recommended news only focus on a subject that target news is about. In contrast, in semantic-based approach [1], similarity of news is usually based on a knowledge base available to exploit semantic relationship between elements appeared in these news. Therefore, recommended news will likely expand the subjects than that of content-based approach. Both approaches have some weaknesses limit, which limit their effectiveness in news recommendation. Our approach is a hybrid one in the sense that it combines content-based recommendation and semantic-based recommendation. In concrete, similarity of news is a linear combination of content-based similarity and semantic-based similarity. The experimental results indicate that this combination brings news results suggest more effective than using either measure separately.

This work is in a part of development research of News Aggregation System BKSport [11] that is based on Semantic Web technology, aiming to effectively handle the amount of sports news gathered from various sources on the internet. Therefore, it inherits results obtained in our previous research such as ontology and knowledge base in the sport domain, methods for named entity recognition and semantic relationships extraction between entities in the news.

The rest of the paper is organized as follows. Section 2 describes previous works related to measuring semantic similarity between news. Section 3 presents more in details of our proposed method. In Section 4, we present the experiments and the evaluation we performed using the implementation of the proposed recommender. Subsequently, advantages and disadvantages of this method, as well as corrective measures and future research lines are concluded in Section 5.

2. Related work

Traditionally, many content-based recommenders [7, 9] use term extraction methods like TF-IDF (Term Frequency-Inverse Document Frequency [10]) in conjunction with the cosine similarity measure in order to compare the similarity between two documents. TF-IDF is used to measure the importance of a word in a document based on its frequency of occurrence in the entire document dataset (or corpus). After calculating TF-IDF value for each word in document, this metric is combined with Cosine measure or Jacard measure to calculate similarity between two documents.

TF-IDF value of the word appeared in document is calculated by the following formula:

$$TF\text{-}IDF_{ij} = TF_{ij} \times IDF_i$$

In which:

$$TF_{ij} = \frac{n_{ij}}{\sum_k n_{kj}} \text{ and } IDF_i = log\frac{|D|}{|\{d:t_i \in d\}|}$$

n_{ij} is number of occurrences of the word i in document j and $|D|$ is total number of document in the dataset.

Then, document is represented as a vector V_i obtaining N dimensional vector (With N is the size of dictionary), value of each element of vector is TF-IDF value of the word. If the word in the dictionary does not belong to news, value of corresponding element in the vector is 0.

In semantic-based approach, previous studies have explored relationship between components between news with each other to calculate semantic similarity. In the study carried out by Batet *et al.* [4], a measure based on the exploitation of the taxonomical structure of a biomedical ontology is proposed for determining the semantic similarity between word pairs. Method proposed by Michel Capelle *et al.* [6] exploited element of similarity between components (words or named entities) in news thereby calculating similarity between two news. To measure the similarity between two components, their proposed method relies on:

- WordNet Dictionary tree when components are words - denoted by sim_{SS}
- PMI measure when components are named entities – denoted by sim_{Bing}. This measure relates to the statistical frequency of occurrence of components and co-occurrence between them

Final formula combines two sim_{SS} and sim_{Bing} measures to calculate semantic similarity between two news as follows (α is correction parameter):

$$sim_{BingSS} = \alpha \times sim_{Bing} + (1 - \alpha) \times sim_{SS}$$

Also exploiting the relationship between components in two news with each other, Frasincar *et al.* [8] presented a number of news recommendation methods in semantic-based approach. Similar to Capelle [6], their work aims to a personalized recommendation system. However user profile of the reader is also built based on the news that the reader has read and calculating similarity between user profile and a news is the same as calculating similarity between two news. Methods presented in this research used ontology and knowledge base to exploit semantic relationship between *concepts*, which are classes in the ontology. Experiment showed that *Ranked Semantic Recommendation 2* is the most effective among them. However, it remains certain limitations that we will show in the following parts and propose method to overcome.

3. Similarity between news items

There are two main approaches in calculating similarity between text news items as content-based and semantic-based. Each approach has its own advantages and disadvantages. We aim to combine these two approaches by combining content-based similarity measure and semantic-based similarity measure with the expectation to overcome limitations of each approach, making recommendation more effective.

3.1. Semantic-based similarity

To calculate semantic similarity, we exploit mutual semantic relations between components in news item. These relations are determined based on ontology and knowledge base that we have built. We extract and analyze components in the news items including: entities, types of entities and semantic annotations. The next sections

will present how to exploit these components in calculating semantic similarity between news items.

3.1.1. Semantic relation between entities

Specifically, in order to exploit relations between entities for calculating similarity between news items, we extend *Ranked Semantic Recommendation 2* method as approved by Frasincar *et al.* [8]. In this method, the authors also used ontology and knowledge base to exploit the relations between entities. However, the method remains some limitations such as:

- It only considers direct relations between entities without considering indirect relations.
- It does not consider the importance of entities as they appear in various positions in the news item (title, description, etc.)

To overcome these above limitations, in Section 3.1.1.1, we present a method to calculate the *relation weight* between entities based on ontology and knowledge base. In addition, we combine the statistical method of co-occurrence of entities in the same news items in determining relation weight between entities, which is presented in Section 3.1.1.2. Finally, we present the method in which uses relation weights between entities in determining semantic similarity between news items in Section 3.1.1.3.

3.1.1.1. Relation weight between entities based on ontology and knowledge base

Aleman-Meza *et al.* presented the methods to calculate the ranking of *Semantic Association* based on *Semantic Path* between the two entities in order to determine the relation weight between entities [3]. Specifically, they define Semantic Association and Semantic Path as follows:

Definition: if two entities e_1 and e_n can be connected together by one or more sequences $e_1, P_1, e_2, P_2, e_3, P_3, \dots, e_{n-1}, P_{n-1}, e_n$ in an RDF graph; here, e_i, $1 \le i \le n$, is entities and P_j, $1 \le j \le n$ is relations in ontology, then we say there exists *semantic relation* between e_1 and e_n.

Sequence $e_1, P_1, e_2, P_2, e_3, P_3, \dots, e_{n-1}, P_{n-1}, e_n$ is a *Semantic Path*.

For example, in the knowledge base, we have:

- <Lionel-Messi> <playFor> <Barcelona-FC> .
- < Luis-Suarez> <playFor> <Barcelona-FC>.

Then, there exists a semantic path between two entities *Lionel Messi* and *Luis Suarez* as follows:

<Lionel-Messi> → <playFor> → <Barcelona-FC> ← <playFor> ← <Luis-Suarez>

As a result, there exists a semantic relation between *Lionel Messi* and *Luis Suarez*.

Based on the properties of semantic path, we identify a *path rank* value to show the relation weight between two entities at both ends of the path. Because there might be multiple semantic paths between two entities, we get the highest *path rank* value to represent relation weight. Aleman-Meza *et al.* [3] used four characteristics of a semantic path to calculate *path rank*, corresponding to four following weights:

- *Subsumption Weight:* based on the structure of the ontology to determine *component weight* for each component (predicate and entity) in the path, thereby calculating weight for the whole path.
- *Path Length Weight:* based on length of the path.
- *Context Weight:* based on determining which region each component of the path belongs to in the ontology. Each region in the ontology has a separate weight depending on the user's interests.
- *Trust Weight:* based on weights of the properties in the ontology.

Applying in news recommendation in football, we found that *Path Length Weight* and *Trust Weight* are two meaningful and appropriate weights. For this reason, we only use these two weights to determine *path-rank* of a semantic path.

Path Length Weight

Length of a semantic path $e_1, P_1, e_2, P_2, e_3, P_3, \dots, e_{n-1}, P_{n-1}, e_n$ is the number of entities and relations in the path (exclude e_1 and e_n). We can see that, when two entities remain indirect relation with each other through which the more there are entities and relations, the lower similarity between these two entities is. Consequently, *path-rank* of a semantic path must be inversely proportional to the length of that path. The Path Length Weight is defined in [3] as below:

$$W_{length} = \frac{1}{length_{path}}$$

In which: $length_{path}$ is the length of semantic path.

For example, we have two semantic paths:

- P_1: <Lionel-Messi> → <playFor> → <Barcelona-FC> → <competeIn> → <La-Liga> ← <competeIn> ← <Real-Madrid> ← <playFor> ← <Karim-Benzema>
- P_2: <Lionel-Messi> → <playFor> → <Barcelona-FC> ← <playFor> ← <Luis-Suarez>

P_1 has length of 7, we obtain:

$$W_{length}(P_1) = \frac{1}{length_{path}} = \frac{1}{7}$$

P_2 has length of 3, we obtain:

$$W_{length}(P_2) = \frac{1}{length_{path}} = \frac{1}{3}$$

From there, we can see that similarity between *Lionel Messi* and *Luis Suarez* is higher than that between *Lionel Messi* and *Karim Benzema*.

Path Relation Weight

There are many different relations defined in the ontology. Every relation represents a different meaning therefore also represents a different relation weight between entities. Some relations show close association, some other relations express loose association. For example, we have two triplets in the knowledge base as below:

- <Luis-Enrique> <managerOf> <Barcelona-FC>.
- <Luis Suarez> <playFor> <Barcelona-FC>.

Here, there exist two relations which are relation <*managerOf*> and relation <*playFor*>. We can see that, relation <*managerOf*> shows more closer than relation <*playFor*>, because each team has only one single manager at a certain time; however, may have a lot of players. Therefore, we assign weight of <*managerOf*> higher than <*playFor*>. And for this reason, from above triplets, we conclude <*Barcelona-FC*> has higher similarity with <*Luis-Enrique*> than <*Luis Suarez*>.

Weight of relations is in the range $(0, 1]$. Path Relation Weight of an overall path P is defined in [3] as below:

$$W_{predicate} = \prod_{p \in path} w_p$$

Relation weight between two entities is based on ontology and knowledge base

Combining two weights W_{length} and $W_{predicate}$ by a pair of coefficients α_{wl} and α_{wp}, we define the path rank of a semantic path as below:

$$W_{path} = \frac{W_{length} \times \alpha_{wl} + W_{predicate} \times \alpha_{wp}}{\alpha_{wl} + \alpha_{wp}}$$

Value W_{path} in the above formula is also similarity value between two entities based on ontology and knowledge base.

3.1.1.2. Relation weight between entities based on statistics of co-occurrence in the same news items

According to the idea of the Michel Capelle *et al.* on PMI measure [6], if two entities co-occur in the same news items many times; these two entities have high similarity to each other. We count co-occurrence of named entity pairs in a dataset on football news to calculate weights PMI. The formula is defined as below:

$$W_{PMI}(e_1, e_2) = log \frac{\frac{c(e_1, e_2)}{N}}{\frac{c(e_1)}{N} \times \frac{c(e_2)}{N}}$$

In which:

- N is the number of news items available in the dataset.
- $c(e_1, e_2)$ is the number of news items in the dataset that two entities u and r co-occur.
- $c(e_1)$ is the number of news items in the dataset containing entity e_1, and $c(r)$ l is the number of news items in the dataset containing entity e_2.

As such, for each any entity pair, we have two values to calculate relation weights: Weight W_{path}(calculated based on semantic path) and weight W_{PMI}(calculated based on statistics of co-occurrence of entity pairs). Before combining these two weights with each other, we normalize them as below:

$$w_{new} = \frac{w_{old} - MIN}{MAX - MIN}$$

In which: MAX and MIN corresponding are maximum value and minimum value in the value chain w.

Finally, we combine these two values together by a pair of coefficients β_{path} and β_{PMI} to calculate similarity of each entity pair as below:

$$Similarity_{entity}\ (e_1, e_2) = \frac{W_{path} \times \beta_{path} + W_{PMI} \times \beta_{PMI}}{\beta_{path} + \beta_{PMI}}$$

By convention, when $e_1 \equiv e_2$ then $Similarity_{entity}\ (e_1, e_2) = 1$.

3.1.1.3. Method for calculating similarity between news items based on relation between entities

First of all, we define set of entities related to entity r is a set containing entities that have similarity where r is greater than 0 and denoted as below:

$$R(r) = \{r_1, r_2, r_3, \dots, r_n\}$$

Suppose there is a news item A, set of recognizable named entities in news item A is denoted as below:

$$A = \{a_1, a_2, a_3, \dots, a_m\}$$

With each entity a_i in set A, we build a set of entities related to a_i corresponding to $R(a_i) = \{a_{i1}, a_{i2}, a_{i3}, \dots, a_{ik}\}$. Grouping all sets $R(a_i)$ together ($i: 1 \to m$), we obtain set of all entities not included in A, but related to A:

$$R = \bigcup_{i:1 \to m} R(a_i)$$

Finally, we group two sets A and R to obtain set A_R called as expansion set of news item A:

$$A_R = A \cup R$$

In the next step, we calculate ranking value for each entity in the set A_R. Each rating value will characterize the relevance of the entity corresponding to news item A. These ranking values should satisfy some properties:

- (1) If the more times an entity appears in the news item, the greater that entity's ranking value is.
- (2) If the greater of entities in the news item that an entity is relevant to, the greater that entity's ranking value is.
- (3) Ranking value also depends on appearance position of the entity in the news item.

Regarding property (3), we determine an entity that can appear in the different positions of the news item, as follows: title, description, bolder-text (bold text, image title, etc.) and content. We also identify importance weight for these positions respectively as below:

$$W_{title} > W_{description} > W_{boldertext} > W_{content}$$

To calculate the ranking value for each entity in the set A_R, based on *Ranked Semantic Recommendation 2* technique [8], we also represent entities in a matrix, in which the first row represents entities in the set A_R and the first column represents entities in the set A. Matrix takes the following form:

	$\boldsymbol{e_1}$	$\boldsymbol{e_2}$	$\boldsymbol{\dots}$	$\boldsymbol{e_q}$
$\boldsymbol{a_1}$	h_{11}	h_{12}	$\dots$	h_{1q}
$\boldsymbol{a_2}$	h_{21}	h_{22}	$\dots$	h_{2q}
$\boldsymbol{\dots}$	$\dots$	$\dots$	$\dots$	h_{3q}
$\boldsymbol{a_m}$	h_{m1}	h_{m2}	$\dots$	h_{4q}

In above matrix, we calculate the value h_{ij} as below:

$$h_{ij} = similarity(a_i, e_j) \times WE(a_i)$$

In which $WE(a_i)$ is importance weight of the entity a_i in the news. This weight is calculated as follows: Suppose a_i is an entity appeared in the news item, and N_{title}, $N_{description}$, $N_{boldertext}$, $N_{content}$ are respectively numbers of occurrences of a_i in the title, description, bolder-text and content of the news item. We define the importance weight of entity a_i as below:

$$WE(a_i) = N_{title} \times W_{title} + N_{descript\ ion} \times W_{description} + N_{boldertext} \times W_{boldertext} + N_{content} \times W_{content}$$

Finally, as the formula defined in [8], the ranking weight of each entity e_j in the set A_R is calculated by:

$$Rank(e_j) = \sum_{i=1}^{m} h_{ij}$$

Assume V_A is a vector containing above calculated $Rank(e_i)$ values. We normalize values of each element in V_A in the range [0, 1]. Normalization formula is expressed as follows:

$$v_i = \frac{v_i - MIN}{MAX - MIN}$$

In which MAX and MIN are maximum value and minimum value respectively of elements in vector V_A. If $MAX = MIN \neq 0$ then $v_i = 1$, with every value of i.

As a result, taking all the steps above will obtain a vector for each news. Final step is calculating similarity between any two news based on their vectors.

Suppose we have two news A, B and two corresponding vectors V_A, V_B. Because these two vectors can have different number of dimensions, we define the similarity between two vectors V_A, V_B (also similarity between two news A and B) as a variation of cosine similarity as below:

$$similarity_{based\ -entity}\ (A, B) = cosine(V_A, V_B) = \frac{\sum_{e_a \equiv e_b}^{e_a \in A,\ e_b \in B} v_a \times v_b}{\sqrt[2]{\sum_a^{e_a \in A} v_a^2} \times \sqrt[2]{\sum_b^{e_b \in B} v_b^2}}$$

In which v_a, v_b corresponding are values $Rank(e_a), Rank(e_b)$ in vectors V_A, V_B.

3.1.2. Types of entities appeared in the news items

A reader who is interested in a subject is more likely to be also interested in other subjects of the same type. For example, if a reader is reading the news about football teams, then that reader tends to continue reading other news items about football teams rather than news items about players or stadiums. Therefore, if two news items have similarity in the types of entities, similarity of these two news items will be higher.

Fig. 1. An example of similarity between news based on types of entities in the news

In ontology, each named entity is defined in the knowledge base will belong to a certain object class defined. These classes can be regarded as the type of entity. For example, two entities *Lionel Messi* and *Luis Suarez* in the knowledge base have the same type, because they belong to class *FootballPlayer*; however, both are not the same type with entity *Barcelona-FC* because this entity belongs to *FootballTeam*.

Statistics of entity types appeared in the news items is similar to statistics of entities. Two different entities can be of the same type. Appearance position of entities also affects association weight between entity type and corresponding news item. These weights will be calculated based on appearance frequency and appearance position of entities of that type. Suppose, we calculate association weight for entity type C for a news item A. Given that c_i is entities of class C appeared in news item A, we define the association weight of entity type C with news item A as below:

$$WC(C) = \sum WE(c_i)$$

We build a vector for news item with elements as WC weights similar to building vector based on entity in section *3.1.1.3*. Elements in each vector will be normalized before using variations of the formula for calculating similarity between vectors used in section *3.1.1.3*. This value is denoted by $similartiy_{based\ -type}$.

3.1.3. Semantic annotations of the news items

Semantic annotations here are triplets in the form of <subject> <predicate> <object>. In which *subject* and *object* are two entities. These semantic annotations also play an important role because they represent somewhat content that news item is talking about.

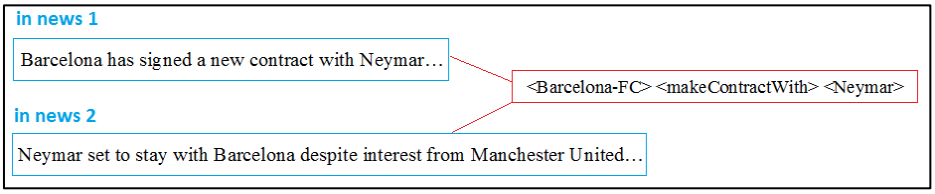

Fig. 2. An example of similarity between news items based on semantic annotations of news

A news item may contain many triplets and a triplet may appear several times. Triplets appeared several times in the news item will be important triplets, showing main contents that news item mentions. Moreover, appearance position of these triplets in the news item also expresses their importance. The importance of positions in the news item (title, description, bolder-text, content) is similar to that presented in the previous section. The more common triplets of two news items, the higher their similarity is.

With each triplet, we denote N_{title}, $N_{description}$, $N_{boldertext}$, $N_{content}$ are numbers respectively of occurrences of this triplet in title, description, bolder-text, content. We use the same formula as the one for calculating importance weight of the entities in Section *3.1.1.3.* to compute importance weight WT of each triplet in the news item. Then we represent these weights as elements of a vector then use vector normalization formula to put these weights in the range [0, 1]. To calculate similarity between news items based on semantic annotations, we use a variation of Cosine formula as described in Section *3.1.1.3.* to compute the distance between two vectors. This value is denoted by $similarity_{based\ -annotation}$.

Thus, we use three parameters to determine semantic similarity between news items, based on the following factors:

- Relations between named entities,
- Types of entity in the news items,
- Semantic annotations of the news items.

Each of these three parameters has different meanings in determining semantic similarity between news items. We combine these three parameters together to determine the final value showing semantic similarity between news items. To combine these three parameters, we use a set of three parameters including θ_{entity}, $\theta_{annotation}$, θ_{type} to express the level of importance of each of the above parameters. We define the final formula for calculating semantic similarity between two news items is as below:

$$\begin{aligned} Similarity_{sema\,ntic}(A,B) \\ &= similarity_{based\ -entity}(A,B) \times \theta_{entity} \\ &+ similarity_{based\ -annotation}(A,B) \times \theta_{annotation} \\ &+ similarity_{based\ -type}(A,B) \times \theta_{type} \end{aligned}$$

3.2. Content-based similarity

With news recommendation method in which only uses semantic similarity as proposed above, we may encounter some problems as:

- Insufficient or incorrect identification of named entities that appear in the news item.
- Insufficient semantic annotations of the news item.

Occurrence of above limitations is caused by limited information in the ontology and knowledge base. This is unavoidable since the construction of ontology and knowledge base must be done manually or semi-automatically, so a lot of efforts need to be made. Furthermore, the evolution of real world knowledge, for example when new players come or players change their clubs, makes it difficult to timely update.

To overpass these limitations, we combine the proposed semantic similarity and content similarity of two news items.

In this section we describe the content-based similarity which is computed using TF-IDF weight of words in the news item combined with cosine measure.

Words with high TF-IDF weight are often important words, showing main contents of the news item. So, we are only interested in words with high TF-IDF weight. Steps to build a set of important words of the news item include:

- *Step 1*: Eliminate stop words. Stop words are words that do not make sense in the representation of contents of the news, such as: “a”, “an”, “the”, etc.
- *Step 2*: Standardize words into infinitive form. Verbs or nouns often exist in many different forms depending on the context, although they still express the same meanings. For example, "make", "makes" and "made". So, we will change them into infinitive form.
- *Step 3*: Calculate TF-IDF for each word in the news (After being standardized in Step 2).
- *Step 4*: Sort and select top words with the highest TF-IDF based on defined threshold.

After above steps, we obtain a set of words with the highest TF-IDF. We represent news item in the form of a vector containing values v_k as TF-IDF value of words in the above set. Similarity measure between two news A and B with two important word sets S_A, S_B and two corresponding vectors V_A, V_B will be calculated based on variation of Cosine formula as below:

$$Similarity_{TF-IDF}(A,B) = \frac{\sum_{t_a \equiv t_b}^{t_a \in S_A,\, t_b \in S_B} v_a \times v_b}{\sqrt[2]{\sum_a^{t_a \in S_A} v_a^2} \times \sqrt[2]{\sum_b^{t_b \in S_B} v_b^2}}$$

In which:

- t_a, t_b are corresponding words in two sets S_A, S_B.
- v_a, v_b are TF-IDF values of words t_a, t_b.

3.3. News recommendation algorithm with combined similarity

To combine semantic similarity $Similarity_{semantic}$ with content similarity $Similarity_{TF-IDF}$ of two news items, we use pair of weights $\gamma_{semantic}$ and $\gamma_{content}$. We define the combination formula as below:

$$Similarity_{combined}\ (A, B) = Similarity_{semantic}\ (A, B) \times \gamma_{semantic}\ + Similarity_{TF-IDF} \times \gamma_{content}$$

News recommendation algorithm as below:
Input: Target news item A and set N candidate news items C.
Output: set of K news items with the highest semantic similarity with A.

- *Step 1*: Identify named entities, make semantic annotations for news item A and candidate news items in set C.
- *Step 2*: Build set of words with the highest TF-IDF weight for news item A and news items in set C.
- *Step 3*: With each news C_i in set C, take the following steps:
    - *Step 3.1*: Calculate $Similarity_{based\ -entity}\ (A, C_i)$
    - *Step 3.2*: Calculate $Similarity_{based\ -annotation}\ (A, C_i)$
    - *Step 3.3*: Calculate $Similarity_{based\ -type}\ (A, C_i)$
    - *Step 3.4*: Calculate $Similarity_{semantic}\ (A, C_i)$ based on the results of steps *3.1*, *3.2* and *3.3*.
    - *Step 3.4:* Calculate $Similarity_{TF-IDF}(A, C_i)$
    - *Step 3.5:* Calculate $Similarity_{combined}\ (A, C_i)$ based on the results of steps *3.4* and *3.5*.
- *Step 4*: Sort news items C_i in descending order according to value $Similarity_{combined}\ (A, C_i)$.

Step 5: Get k news items in the top of the list sorted in Step 4 to recommend for news item A.

Assume that n_t is the average number of tokens in a news item and n is the number of news items in dataset C. We see that, in step 1, the complexity of named entity recognition and semantic annotation of a news item is $O(n_c n_t)$, where n_c is the total number of classes, entities and properties in ontology and knowledge base. Therefore, for n news items in the set C and a news item A, the time complexity of step 1 is $O(nn_c n_t)$. Step 2 transfers n+1 news items into vector TF-IDF. As we had computed the IDF for all tokens in the dictionary before running the algorithm, the time complexity of transferring a news item into a vector TF-IDF equal to the time complexity of calculate TF values for all tokens in that news item, $O(n_t)$. Consequently the complexity of step 2 is $O(nn_t)$. On the other hand, step 3 is repeated n times for each element in C. The steps from 3.1 to 3.4 are the multiplication of the pair of vectors TF-IDF, therefore, the time complexity of each iteration is $O(n_t)$ and the time complexity of step 3 is $O(nn_t)$. The time complexity of the sort algorithm in step 4 is O(nlogn). As a result, the time complexity of the proposed algorithm is $O(nn_c n_t + nlogn)$.

4. Experiment and evaluation

4.1. Experiment scenario

The goal of this chapter is to evaluate and compare the effectiveness of three news recommendation methods:

- Only use semantic similarity between news items.

- Only use content similarity between news items.
- Combine both above similarities.

The evaluation of the different methods is performed by measuring precision. Because we did not build an online system yet, so we use offline evaluation method for evaluation. For offline evaluation, we choose N=100 news items (symbolized as set A) from a number of famous sports websites such as http://www.skysports.com/ , http://www.espnfcasia.com/ , http://sports.yahoo.com/ and then we ask collaborators to rate that a news item as relevant or non-relevant with another one. After that, we have an experiment dataset in which each news item A_i will have K_{A_i} $(0 \leq K_{A_i} \leq N - 1)$ related news items and $(N - 1 - K_{A_i})$ unrelated news items. We separately run methods above for each news item A_i in set A and also generate K_{A_i} news items with the highest similarity with it, then compared with K_{A_i} news items that collaborators have identified in experiment dataset. For example, consider the news item A_1, collaborators discover 5 news items in the remaining 99 news items related to A_1, then algorithm automatically run also generated 5 corresponding news items, then compared them with 5 news items that collaborators have identified.

Symbol:

- TP_{A_i} is the number of news items that the algorithm precisely recommends for news item A_i .
- FP_{A_i} is the number of news items that the algorithm imprecisely recommends for news item A_i.
- FN_{A_i} is the number of related news items that the algorithm not recommend for news item A_i.

We define *precision* for a news item A_i, using the following formula:

$$precision(A_i) = \frac{TP_{A_i}}{TP_{A_i} + FP_{A_i}} = \frac{TP_{A_i}}{K_{A_i}}$$

Follow the way that we implement, we obtain $FP_{A_i} = FN_{A_i}$, then $precision(A_i) = recall(A_i)$. There for we only concern about $precision$ to evaluate these above methods. Finally, we define the final precision of the method as the average of precisions for the entire N news items in the experiment dataset.

$$Precision(A) = \frac{\sum_{A_i \in A} precision(A_i)}{N}$$

4.2. Experiment parameters

Certain parameters are employed to determine the importance of the components when these components are combined together. In this experiment, we set the value of parameters totally based on our point of view. For instance:

- Weights w_p of relations in the ontology to calculate W_{path} was assigned based on our perception on the relevance of each relation: $w_{managerOf} = 0.8$, $w_{playFor} = 0.6$, $w_{stadiumOf} = 0.5, \ldots$
- $\gamma_{semantic}$ and $\gamma_{content}$ are two parameters used when combining semantic similarity measure and content similarity measure between news items. As we consider the importance of content similarity is higher than the one of

semantic similarity in news recommendation, we choose $\gamma_{semantic} = 1$, $\gamma_{content} = 2$.

4.3. Experiment results and evaluation

After running three separate methods for set A containing 100 news items as experiment scenario as presented in section 4.1, we obtain precision result of each method shown in Table 1.

Table 1. News recommendation precision in circumstances

	Precision
Only use semantic similarity (*semantic-based)*	75.8 %
Only use content similarity (*content-based)*	82.2 %
Combine both similarities (*combined*)	85.6 %

Assessment of experiment results

Table 1 indicated that, for the experiment data A containing 100 news items, the *semantic-based recommendation method* is not as precise as the *content-based recommendation method.* Meanwhile, if combining the content-based similarity method and semantic-based similarity method, it will bring the best results. This can be explained as follows:

- When using only the semantic-based similarity (*semantic-based approach)*, it is mainly dependent on the entities in the news items. Therefore, in some case, the algorithm recommends correct news items about the relevant entities but the completely different topic. For some collaborators, they will seem as irrelevant.
- Following the *content-based approach*, the recommended news item's topic is usually quite close to the target news item. However, this method does not have the ability to expand the topic. If we have two news items about Barcelona club in which the first news item is about the play of the Club and the second one is about the transfer of the Club's players, *the content-based approach* will determines that the similarity of these news items is low.
- When combining the content-based similarity and semantic-based similarity, the recommended news will overcome the limitations of each separated measure, leading to more efficient recommendation.

5. Conclusions and future work

In this research, we presented a recommendation method based on the combination of the content-based similarity and semantic-based similarity of the news items. The semantic-based measure is calculated based on the semantic relation among objects. It enables the recommendation not only stopping at the suggestion of the similar topic news items or news items rounding a key object of the target news item, but also being able to recommend the news items of other objects that these objects have a semantic relation with other ones in the target news item. However, the similarity measure is mainly focused on the entities and not considered the context mentioned in the news item. The content-based measure will overcome the weakness of semantic-

based measure by extracting from the news item the words having the highest TF-IDF value and these words are characterized the main context mentioned in the news item.

We evaluated and compared the precision of the proposed method and the recommendation method when using only either measure separately. The experimental results showed that the combination of the two similarities helps to promote the effectiveness of both and overcome the weaknesses of each other method, ultimately increasing the better recommendation. However the proposed method remains some limitations such as its dependency on the adequacy of the knowledge base and ontology. Determining the weights in such a way so that the combination of the measures achieves the highest efficiency is also a difficult problem to be solved of the method.

References

1. Abdelrahman, A., Kayed, A.: A Survey on Semantic Similarity Measures between Concepts in Health Domain. In: American Journal of Computational Mathematics, 5, 204-214 (2015).
2. Ahn, J.W., Brusilovsky, P., Grady, J., He, D., Syn, S.Y.: Open User Profiles for Adaptive News Systems: Help or Harm?. In: 16th International Conference on World Wide Web (WWW 2007), pp. 11-20. ACM (2007)
3. Aleman-Meza, B., Halaschek, C., Arpinar, I.B., Sheth, A.: Context-Aware Semantic Association Ranking. In Proceedings of the Semantic Web and Database Workshop, Berlin, pp. 33-50.
4. Batet, M., Sánchez, D., Valls, A.: An Ontology-Based Measure to Compute Semantic Similarity in Biomedicine. Journal of Biomedical Informatics, 44, 118-125 (2011).
5. Billsus, D., Pazzani, M.J.: A Personal News Agent that Talks, Learns and Explains. In: 3rd Annual Conference on Autonomous Agents (AGENTS 1999), pp. 268-275, ACM (1999)
6. Capelle, M., Hogenboom, F., Hogenboom, A., Frasincar, F.: Semantic News Recommendation Using WordNet and Bing Similarities. Proceedings of the 28th Annual ACM Symposium on Applied Computing, pp. 296-302.
7. Elahi, A., Javanmard Alitappeh, R., Shokohi Rostami, A.: Improvement TFIDF for News Document using efficient similarity", Research Journal of Applied Sciences, Engineering and Technology 4(19): 3592-3600, (2012).
8. Frasincar, F., IJntema, W., Goossen, F., Hogenboom, F.: Ontology-based news recommendation. Proceedings of the 2010 EDBT/ICDT Workshops, Lausanne, Switzerland, March 22-26, (2010).
9. Huang, A.: Similarity Measures for Text Document Clustering. In Proceedings of the 6th New Zealand Computer Science Research Student Conference, Christchurch, New Zealand, 14–18 April 2008; pp. 49–56.
10. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing and Management: an International Journal archive. Volume 24 Issue 5, 1988, pp. 513-523.
11. Tuan-Dung, C., Quang-Minh, N., Hoang-Cong, N., Hagino, T.: Towards efficient sport data integration through semantic annotation. Proceeding of The Fourth International Conference on Knowledge and Systems Engineering KSE 2012, pp. 99-106, ISBN 978-1-4673-2171-6, Da Nang Viet Nam, August, 2012.