International Journal of Innovative
Computing, Information and Control
Volume **11**, Number **6**, December **2015**

 ISSN 1349-4198
pp. **1947–1963**

A SEMANTICALLY HYBRID FRAMEWORK OF PERSONALIZING NEWS RECOMMENDATIONS

Cuong Dinh Hoa Nguyen, Ngamnij Arch-int* and Somjit Arch-int

Department of Computer Science
Faculty of Science
Khon Kaen University
123 Moo 16 Mittapap Rd., Nai-Muang, Muang District, Khon Kaen 40002, Thailand
ndhcuong@hce.edu.vn; *Corresponding author: ngamnij@kku.ac.th; somjit@kku.ac.th

Received April 2015; revised September 2015

Abstract. *Personalizing news recommendation, which explores different reading interests of different readers, has become an active research field in recent years. In this paper, we present a semantically hybrid framework of personalizing news recommendations which focuses on mining the relationships between named entities appearing in the articles and users' interests. Specifically, the recommendation engine aggregates the scores of collaborative filtering module and content-based module to produce the recommendation list. In the content-based module, reader's preferences are discovered at both term level and semantic level by k-NN classifier and associative classifier, respectively. The experimental results showed good performance in accordance to the real world data set extracted from well-categorized web news portals, and thus supported the use of the proposed framework.*

Keywords: Personalized news recommendation, News recommender system, k-NN, Associative classifier, Collaborative filtering

1. **Introduction.** With the dramatic development of the Internet, the electronic papers have become more popular in the daily life with millions of news articles which have been published everyday by thousands of news portals web-wide. Choosing what to read becomes a difficult task for readers due to the big amount of papers. This difficult task leads to the rising research topic of personalizing news recommendations in recent years [8, 13, 20, 21, 26, 37].

The approaches to personalizing news recommendations can be classified into three categories including collaborative filtering [31], content-based [2] and hybrid [28, 34] approaches. In the domain of personalizing news recommendations, analyzing news article content is one of the most important tasks for not only the content-based approach but also the hybrid approach. While frequently used techniques for representing article contents (e.g., Term Frequency - Inverse Document Frequency (TF-IDF) [2], Probabilistic Latent Semantic Indexing (PLSI), and Latent Dirichlet Allocation (LDA) [22]) can only capture the surface meaning of words, recent semantic studies have introduced to the literature different approaches to express article contents semantically (e.g., Concept Frequency - Inverse Document Frequency (CF-IDF) [14], Synset Frequency - Inverse Document Frequency (SF-IDF) [5] or named entities similarity [6]). For news articles, named entities play an important role in attracting readers [26]. Hence, many researchers have paid attention to named entities in their studies for better describing users' reading preferences [13, 22, 26, 28]. However, the relationships between named entities have not been explored. In our opinion, named entities and their relations could be used not only to describe users'

reading preferences but also to improve the performance of recommendation-making process.

To be more specific, named entities, their conceptualization levels and their relations can be utilized to express users' reading preferences. For the conceptualization of named entities, while most of the studies accept named entities as instances of general concepts (e.g., Person, Location, Organization), these entities actually can be described by specific concepts which are the sub-concepts of those general concepts (e.g., Student, Politician, Company). Hence, user's preferences can also be captured at different concept levels. For instance, through analyzing reading log of a user A, we can find out that user A likes to read articles about *Football Player*. In addition, the co-occurrence of named entities in each article of A's reading profile can be used to mine Class Association Rules (CAR) which reveals the user's preferences. For example, "*Lionel Messi*" $\wedge$ "*FC Barcelona*" $\rightarrow$ *Like, [support = 1%, confidence = 80%].* From this kind of association rule, we can combine with concepts of named entities to produce CAR of concepts which can be added to the user preference model. This kind of rule looks like this "*Football Player*" $\wedge$ "*Football Club*" $\rightarrow$ *Like, [support = 1%, confidence = 80%].* These CAR are used to build up the associative classifiers that are the key component to give personalized news recommendations. However, the CAR of named entities faces the problem of unseen named entity which has never appeared in the user's reading history before. This problem can be solved by substituting the unseen named entity with the most similar named entity in a common knowledge base [11] like DBpedia [3], YAGO [32] or YAGO2 [18].

Based on the above analysis, this research proposes a semantically hybrid framework of personalizing news recommendation whose novelty is the exploration of associative relationships between named entities and users' interests in a hybrid scenario for personalizing news recommendations. Specifically, article content is analyzed at both term level and semantic level, in which, the k-nearest neighbors (k-NN) algorithm and the associative classifier are used to predict user's reading preferences at term level and semantic level, respectively. In addition, an ontological inference algorithm is proposed to serve the process of mining CAR which are used to reveal user's preferences at different concept layers. We use DBpedia as a common knowledge base to support the associative classifier in solving the unseen named entity problem. Furthermore, the recommendation is decided by aggregating the decisions of k-NN classifier, associative classifier with the decision of collaborative filtering module. Based on the aggregated recommendation scores, a top-K personalized recommendation list is produced in descending order. We developed a system prototype to evaluate the proposed framework. An experiment was conducted with real world data set retrieved from well-categorized web news portals showing good performance which supported the use of the proposed framework.

The rest of this paper is structured as follows. Section 2 reviews the state-of-the-art studies of personalizing news recommendations. Section 3 presents the semantically hybrid news recommendation framework. Experimental results and discussions are shown in Section 4. Finally, conclusion and future work are given in Section 5.

2. **Related Work.**

The approaches to news recommender systems can be categorized as collaborative filtering, content-based and hybrid approaches. While the content-based approach analyzes article content to discover user's interests, the collaborative filtering approach uses article-related data, especially users' behavior data, to find like-minded readers' choices for recommending. The combination of content-based and collaborative-filtering approaches introduces the hybrid approach to the literature. In this section, we review some recently state-of-the-art studies of personalizing news recommendation for each kind of the above approaches.

The collaborative filtering approach gives personalized news recommendations based on the like-minded users' choices. Therefore, user's behavior data (e.g., user's ratings) are very important to this approach. Resnick et al. [31] presented GroupLens, which discovers the like-minded readers by analyzing readers' rating logs. And news items' rating data of like-minded readers were used to make recommendation for active reader. Another news recommender system using collaborative filtering was introduced by Das et al. [10] in which the recommendation score was produced by a linear model, which combined results of three methods including MinHash clustering, PLSI and covisitation counts. However, in case of new items do not receive enough users' rating data, which is known as cold-start problem, the collaborative filtering approach reveals its weakness that cannot efficiently produce recommendation in this situation.

In the line of content-based approach, the two popular techniques of representing article content are TF-IDF [2] and topic distributions generated by probabilistic models [8]. Based on the content representations, different classifiers were introduced to generate personalized news recommendations such as k-NN [2] or probabilistic model [8].

In order to improve the recommendation performance, the hybrid approach, which combines the advantages of both content-based and collaborative filtering approaches, has been introduced to the literature. Especially, the recently hybrid approaches have often used TF-IDF to represent article contents [23, 26, 28, 34]. And the recommendation engines of these studies were built upon different methods such as information entropy [23], linear model [28] or Bayesian inference [26, 34] to deliver news recommendations.

Although TF-IDF has been used popularly, its major limitation is the ability of capturing word-meaning. Therefore, different semantic extensions of TF-IDF technique were presented such as CF-IDF [14], SF-IDF [5]. Besides, named entities, which are important semantic elements, were also used to express article content [6, 13]. With the advantages of Semantic Web technology, named entities were described clearly and semantically by using ontological concepts [4, 12, 19, 25]. Then different methods were applied to express article contents such as vector space model of concepts, weighted concept networks, etc. Based on the semantic representation of article contents, the recommendations were generated by using different solutions such as semantic relatedness between articles [4, 12], Markov chain [19] or semantic expansion networks [25].

In this study, we present a novel semantic method for describing user's reading preferences by exploring the relationships between named entities and user's interests. These relationships are expressed under the form of class association rules. In order to improve the recommendation performance, a semantically hybrid framework, which composes content-based and collaborative filtering modules, is introduced. The recommendation engine aggregates the scores of content-based module and collaborative filtering module to produce personalized recommendation lists.

3. **The Semantically Hybrid News Recommendation Framework.**

The architecture of the semantically hybrid framework of personalizing news recommendation is shown in Figure 1. User's rating logs, and articles are used as the inputs for the semantically hybrid recommendation system which composes three main components including the content-based module, the collaborative filtering module, and the recommendation engine. Specifically, the user's reading preferences are discovered directly by the content-based module, while the like-minded users are identified by the collaborative filtering module. In the recommendation engine, the results of the above two modules are aggregated by a linear model to produce final recommendation decision. Finally, the personalized recommendation items are sent to accordant individuals. In the rest of this section, we describe each module in detail.

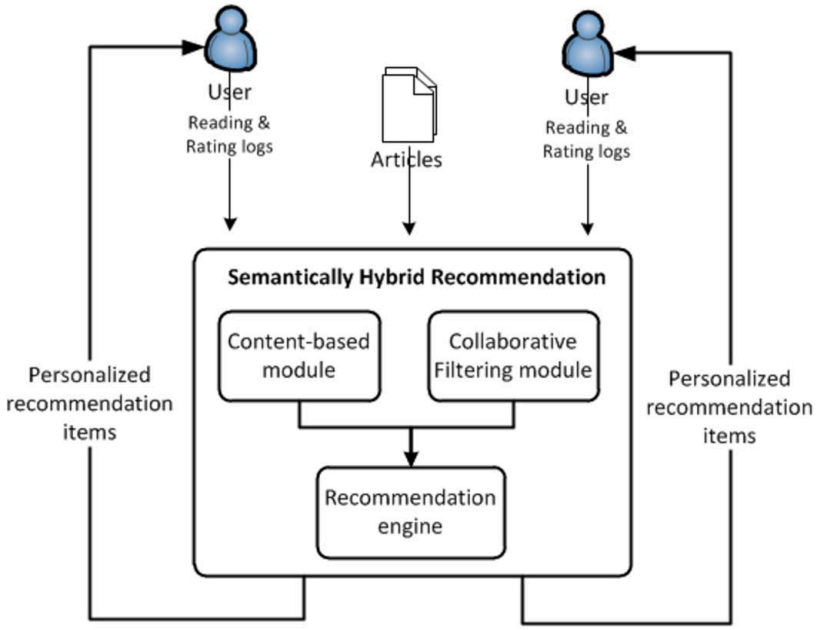

FIGURE 1. The architecture of the semantically hybrid framework of personalizing news recommendations

3.1. **The content-based module.** The content-based module discovers user's reading preferences directly through analyzing the article content. The considered contents of the articles include titles, abstracts, contents, and published times. This research uses text mining technology and Semantic Web technology to analyze the article content.

In a nutshell the operations of the content-based module are shown in Figure 2. Firstly, the content processing unit analyzes the users' logs and retrieves article content from downloaded html files. Secondly, the retrieved contents are processed in both term and semantic levels by term extraction unit and semantic annotation unit, respectively. The semantic annotation process is supported by the local ontology. Thirdly, the TF-IDF vectors of the papers are built upon the results of term extraction unit, while the entity and the concept sets of the papers are generated based on the results of the semantic annotation process with the support of the local ontology and DBpedia. Next, the CAR Miner unit discovers the relations between user's preferences and entity sets as well as concept sets by mining class association rules. These rules are then selected and populated into the rule base. Fifthly, the prediction at term level is produced by the k-NN classifier and the prediction at semantic level is produced by the associative classifier. Finally, these predicted results are combined to give the final decision of the content-based module.

3.1.1. *The local ontology.* News articles often talk about events associated with named entities of the four general major concepts including *Person*, *Organization*, *Location* and *Time*. These entities hold the main meaning of the paper. In other words, user's preferences can be described by the co-occurrences of these entities. Therefore, the more details we describe these entities, the deeper understanding we gain about user's interests. For example, assuming user likes a paper containing entities named "*Bob Wielinga*" and "*VU University Amsterdam*" that are normally annotated as *Person* and *Organization*, respectively. Moreover, we can deeply describe these entities by annotating "*Bob Wielinga*" as an instance of *Professor* concept and "*VU University Amsterdam*" as an instance of *University* concept. By this way, user's preferences can be described not only

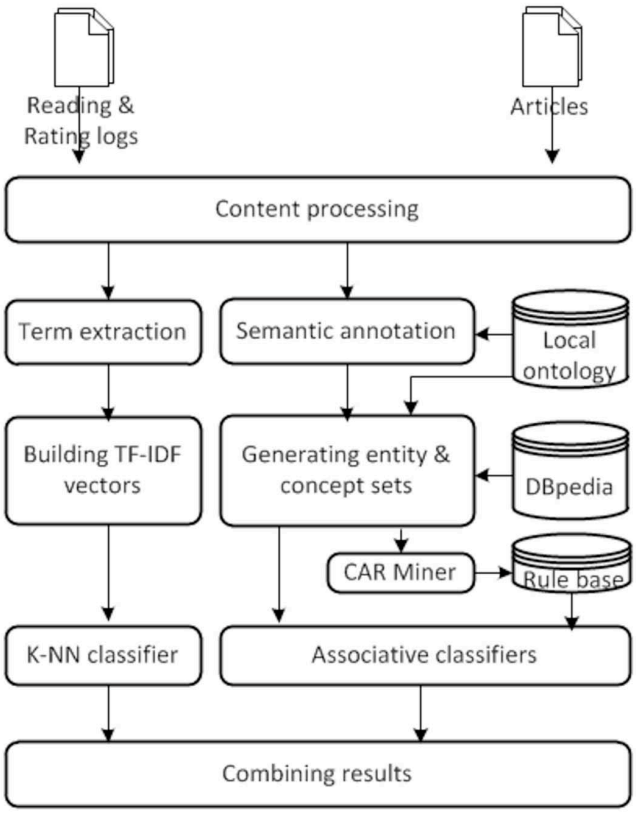

FIGURE 2. The content-based module

by the co-occurrence of entities like "*Bob Wielinga*" and "*VU University Amsterdam*" but also by the co-occurrence of concepts like *Professor* and *University*.

In order to satisfy the above usage scenario, we developed a local ontology which provides vocabulary for annotation process and serves as the foundation of reasoning process. For building the local ontology, we implemented a two-phase approach. Firstly, we conducted a survey to collect concepts relating to personal job titles, types of organizations, kinds of locations, and time. The results were then assessed by two experts who are journalists. Based on the expert's assessments, we enumerated the synonyms, hypernyms and hyponyms of each concept by using WordNet[1]. Secondly, we applied the top-down approach to construct the domain ontology based on the retrieved hierarchical concepts. Figure 3 shows an excerpt of the local ontology. This ontology was developed by using Protégé[2] editor and represented in OWL language.

3.1.2. *Term extraction and k-NN classifier.* In this work, the term extraction unit, which also removes stop words and reducing the number of words by stemming, is responsible for representing the article's terms in vector space model. We use TF-IDF method to measure the importance of terms in the corpus. The calculation of TF-IDF weight is showed in Equations (1), (2), and (3).

$$tf(t_i, d_j) = f(t_i, d_j) / \max(f(t_k, d_j)) \tag{1}$$

$$idf(t_i) = \log_e \left(\frac{\text{Total number of articles}}{\text{Number of articles contain term } t_i} \right) \tag{2}$$

$$w(t_i, d_j) = tf(t_i, d_j) * idf(t_i) \tag{3}$$

[1]http://wordnet.princeton.edu
[2]http://protege.stanford.edu

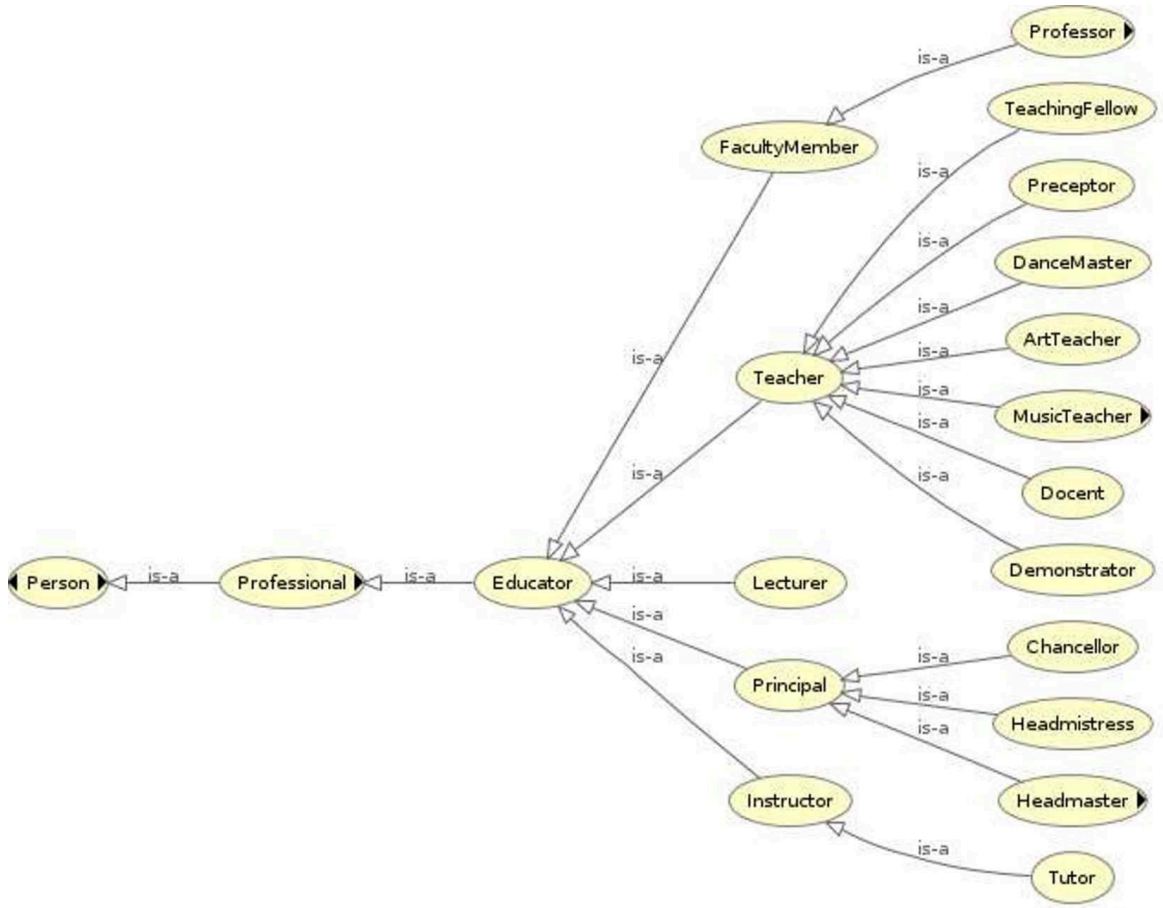

FIGURE 3. An excerpt of the local ontology

where $tf(t_i, d_j)$ is the term frequency of term t_i in article d_j, $f(t_i, d_j)$ is the frequency of term t_i in article d_j, $idf(t_i)$ is the inverse document frequency of term t_i, and $w(t_i, d_j)$ is the TF-IDF weight of term t_i in article d_j.

In order to apply the TF-IDF method, each article content is processed to reduce the number of words by stemming and is transferred into vector of stemmed words of the article. Then, we calculate TF-IDF weight for every word of the vector. For each user's rating profile, the TF-IDF vectors are combined with the relevant ratings to form the labeled training data set. The labeled training data set is then used by the k-NN classifier to reveal the user's preferences as well as to give personalized recommendation at term level. Due to the binary value of rating data (like or dislike), k-NN's classification result of an article p according to the interests of user u is expressed by Equation (4).

$$kNN_{decision}(u,p) = \begin{cases} +1, & \text{if kNN returns Like} \\ -1, & \text{if kNN returns Dislike} \end{cases} \tag{4}$$

The recommended score of the k-NN classifier for a given article p according to the interests of user u is calculated by Equation (5).

$$kNN_{score}(u,p) = kNN_{decision}(u,p) \cdot \frac{n}{k} \tag{5}$$

where k is the predefined parameter of the k-NN algorithm and $n \leq k$ is the major number of nearest neighbors which have the same label.

3.1.3. *Semantic annotation.* Annotating named entities is the task of identifying named entities in text and categorizing them into predefined categories. In this work, we use GATE framework [9], which has been used widely in many related studies, to annotate named entities in article content. The GATE framework not only provides general categories (e.g., Person, Location, Organization, or Time) for annotation but also allows developers to customize the category for their specific annotation purpose. This advantage is suitable for our purpose of annotating named entities according to the vocabulary provided by the local ontology.

For annotation purpose, GATE offers the built-in ANNIE (A Nearly New Information Extraction) system which can annotate named entities with general concepts in default mode. Furthermore, the ANNIE system can be extended by using JAPE (Java Annotation Pattern Engine) rules, which are special component allowing developer to annotate named entities according to their own vocabulary. The operations of ANNIE's components are expressed in Figure 4.

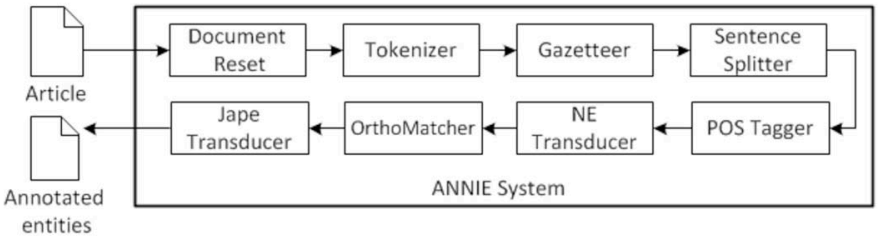

FIGURE 4. The semantic annotation process using ANNIE

The functionality of each ANNIE component is briefly described as follows: (i) Document Reset: resets the document to its original state; (ii) Tokenizer (ANNIE English Tokeniser): splits the text into very simple tokens such as numbers, punctuation and words; (iii) Gazetteer (ANNIE Gazetteer): identifies named entities in text; (iv) POS Tagger (ANNIE Part-Of-Speech Tagger): produces a part-of-speech tag as an annotation on each word or symbol; (v) NE Transducer: produces output of annotated entities; (vi) OrthoMatcher: adds identity relations between named entities; and (vii) JAPE Transducer: is a Java Annotation Patterns Engine allowing to recognize regular expression in annotations of documents.

Generally, the format of JAPE rule is $X \rightarrow Y$ where X is the left-hand side of the rule and Y is the right-hand side of the rule. The left-hand side of the rule consists of annotation patterns, while the right-hand side of the rule consists of annotation manipulation. To build JAPE rule, we apply a two-phases process: (i) figure out the text patterns usually associated with the ontological concepts and their synonyms, and (ii) define the rule template for each case. Following this process, it is clear that the more text patterns we can determine, the more JAPE rules we can build.

Following we present an example of building JAPE rules to annotate named entities of the concept *Professor* and *University*. Given a sentence like this "*Prof. Bob Wielinga works at VU University Amsterdam*". In this sentence, we find two named entities: "*Bob Wielinga*" and "*VU University Amsterdam*" that are originally annotated as *Person* and *Organization*, respectively. However, we know "*Bob Wielinga*" is a Professor because of the appearance of the title word "*Prof.*". Therefore, whenever the word "*Prof.*" or "*Professor*" appears before a *Person* entity, then we can annotate *Person* entity to be an instance of *Professor* concept. This text pattern is expressed in JAPE rule format as shown in Listing 1.

Listing 1. An example of JAPE rule

```
1  Rule: rule01
2  (
3    ({Token.string =~ ``[Pp]rofessor''}) |
4    (({Token.string =~ ``[Pp]rof.''})?)
5  ):temp
6  (
7    PERSON |
8    ({Token.kind==word, Token.category==NNP,
9      Token.orth==upperInitial}
10    ({Token.kind==word, Token.category==NNP,
11      Token.orth==upperInitial})*)
12 ):col
13 -->
14 :col.Professor = {rule = ``rule01''}
```

(i) Lines 1-12: while line 1 states the rule name, lines 2-12 express the pattern. The following lines from 2 to 5 find the title that can be *"Professor"*, *"professor"*, *"Prof."* or *"prof."*. Consequently, there could be a *PERSON* entity or proper noun which is encoded by the term *NNP* (lines 6-12).

(ii) Line 14: in case a given pattern matches the rule antecedent, the named entity will be annotated as an instance of *Professor* concept and the rule name is also recorded.

As for the entity *"VU University Amsterdam"*, which is originally annotated as an instance of *Organization* concept, we can categorize it as an instance of *University* concept based on the appearance of the word *"University"* inside the named entity's value. Based on this text pattern, the accordant JAPE rule using embedded Java language is shown in Listing 2.

Listing 2. An example of JAPE rule which is embedded Java language

```
1  Rule: rule02
2  (
3    {Organization}
4  ):temp
5  -->
6  {
7  AnnotationSet a=bindings.get(``temp'');
8  if(a!=null && a.size()>0){
9  int b=anno.firstNode().getOffset().intValue();
10 int c=anno.lastNode().getOffset.intValue();
11 String mydoc=doc.getContent().toString();
12 String s=mydoc.substring(b,c);
13 if(s.contains(``University'') || s.contains(``university'')){
14  FeatureMap f=Factory.newFeatureMap();
15   f.put(``rule'', ``rule02'');
16   outputAs.add(a.firstNode(),
17               a.lastNode(),
18         ``University'', f);
19    }
20     }
21 }
```

For any *Organization* entity which is matched in lines 1-4, the words of that entity are retrieved and stored in a string variable named *s* (line 12). If that string contains the word *"University"* or *"university"* (line 13), then the *Organization* entity is annotated as an instance of *University* concept (lines 14-19).

Based on this approach, we enumerated the text patterns associated with every concept in the local ontology. For each case, we built the relevant JAPE rule. As a result, we

constructed a JAPE rule base serving for our purpose of annotating named entities by the vocabulary of our local ontology.

3.1.4. *Generating entity sets and concept sets.* The unit of generating entity sets and concept sets is responsible for: (i) providing the entity sets and concept sets as inputs for mining class association rules, which are then used to build associative classifiers. Especially, the concept sets are inferred at different abstract layers to better describe user's preferences; (ii) solving the problem of unseen named entity in the using phase of the associative classifiers. This unit is in charge of analyzing article content at semantic level.

As for the first function of this unit, named entities and their associated concepts are used to express the relations between article content and user's preferences at different concept layers. We adopt the following assumptions:

(i) If user A likes an article containing entities $e_1, e_2, \ldots, e_n$, then he/she also likes the entity set $\{e_1, e_2, \ldots, e_n\}$;
(ii) Each entity is associated with a concept of the local ontology. If user A likes an entity set $\{e_1, e_2, \ldots, e_n\}$, then he/she also likes the relevant concept set $\{c_1, c_2, \ldots, c_m\}$ where $m \leq n$; and
(iii) If the concept c_k belongs to the concept set $C = \{c_1, c_2, \ldots, c_m\}$, which is interested by user A, then the user A also likes the following concept sets $C \setminus \{c_k\} \cup \{super(c_k) | depth(c_k) > 1\}$, where $1 \leq k \leq n$, $super(c_k)$ is the directed super class of c_k based on the hierarchy of the local ontology, and $depth(c_k) = 1$ when c_k is the general concept like *Person, Organization, Location* or *Time*.

For example, if user A likes the article containing named entities "*Bob Wielinga*" and "*VU University Amsterdam*", then he/she also likes the entity set {"*Bob Wielinga*", "*VU University Amsterdam*"}. The original concept set of this entity set, which is annotated by the semantic annotation unit, is {*Professor*, *University*}. From this original concept set, we can infer other concept sets at different abstractions to describe user's preferences at different abstract layers such as {*FacultyMember*, *University*} and {*Educator*, *University*}. More generally, the process of inferring concept sets at different abstract layers from the original one based on the hierarchy of the local ontology is shown in Algorithm 1.

Hence, given a user rating profile, each article is seen as a transaction of entity set and transactions of concept sets. The interesting relationships between entities or concepts can be discovered through the process of mining association rules. Theoretically, association rules can be described as follows. Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of items (in this case i_j can be either an entity or a concept) and let $D = \{t_1, t_2, \ldots, t_n\}$ be a set of database transactions where each transaction t_i owns unique identifier and is also a set of items, $t_i \subseteq I$. An association rule is expressed in the form $A \rightarrow B$, where $A \subset I$, $B \subset I$, and $A \cap B = \varnothing$. The three popular measurements for association rules are expressed in Equations (6), (7), and (8).

$$support(A \rightarrow B) = P(A \cup B) = \mid A \cup B \mid / \mid D \mid \tag{6}$$

$$confidence(A \rightarrow B) = P(B \mid A) = \mid A \cup B \mid / \mid A \mid \tag{7}$$

$$lift(A \rightarrow B) = \frac{P(B \mid A)}{P(B)} = \frac{\mid A \cup B \mid \cdot \mid D \mid}{\mid A \mid \cdot \mid B \mid} \tag{8}$$

where $\mid A \cup B \mid$ is the number of transactions in D containing both A and B item sets. In order to mine association rules, many algorithms have been introduced to the literature such as Apriori [1], ECLAT [36], FP-Growth [17] to name a few.

Input: C - the original concept set
O - the local ontology
Output: result - inferred concept sets
$superSet \leftarrow \emptyset$;
$result \leftarrow \emptyset$;
$superSet \leftarrow superSet \cup C$;
while $superSet \neq \emptyset$ **do**
  $set \leftarrow pop(superSet)$;
  foreach c_i *in* set **do**
    $newSet \leftarrow \emptyset$;
    if $depth(c_i) > 1$ **then**
      $newSet \leftarrow (set \setminus \{c_i\}) \cup (super(c_i))$;
    end
    $superSet \leftarrow superSet \cup newSet$;
  end
  $result \leftarrow result \cup set$;
end
return $result$;

Algorithm 1: The inference algorithm of generating concept sets at different abstract layers

Association rules not only express the associative relationships between item sets but can be applied to the classification problem also. The association rules are used to characterize the relations between item sets and class labels. Therefore, the kind of these rules are called Class Association Rule (CAR) and these rules are expressed in the form of $A \rightarrow c$ where $A \subset I$, C is a set of class labels, and $c \in C$. The classifiers using CAR to classify input item sets are called associative classifiers. There are some studies proposing algorithms to mine CAR and constructing associative classifiers such as CBA [27], CMAR [24], and CPAR [35]. By comparing the performances of CBA, CMAR and CPAR classifiers, Pinho Lucas et al. [29] suggested that the CBA algorithm is more suitable to be employed for recommendation systems than the other two classifiers. Therefore, we adopt CBA algorithm to generate class-association rules.

As for the second function of solving unseen named entity, we adopt the following assumptions:

(i) Linking named entities onto a common knowledge base of entities for disambiguation which is often called entity linking;
(ii) Within a common knowledge base, an entity e can be substituted by its most similar entity, which is measured by the semantic similarity between e and the candidate entities in the knowledge base;
(iii) The user, who likes an entity e, will also like the most similar entity of e.

For entity linking task, we adopted the method of Varma et al. [33] which showed good performance in the comparison study of Hachey et al. [15]. The common knowledge base in this study is DBpedia because this Linked Open Data source holds approximately 1,445,000 persons, 735,000 places, 241,000 organizations, and over 410,000 creative works. This rich information source can satisfy the two important tasks of this unit which include linking entity and find the most similar entity of the unseen named entity.

In order to find the most similar entity of the unseen named entity, we apply the two-phases process including: (i) linking named entities onto the common knowledge base – DBpedia, and (ii) finding the most similar entity of the unseen named entity among

the candidates of DBpedia's entities based on the TF-IDF approach and cosine similarity measurement which was introduced in the work of Di Noia et al. [11].

In the first phase, we apply the method of Varma et al. [33] to compute the cosine similarity between the paragraph context and the candidate entities' Wikipedia pages. Then, the results are ranked and the DBpedia entity, which has closest distance, is determined to be the entity for linking (called linked entity).

In the second phase, we use TF-IDF approach to express linked entity in the vector space model. First, the set of properties associated with the linked entity of the unseen named entity is figured out. Then, for each property p, the TF-IDF vectors of the linked entity of unseen named entity and the linked entities of candidate entities are built. Specifically, each vector attribute is an object in the object set retrieved from attended entities in property p. The cosine similarity is applied to measure the distance between two different linked entities e_i and e_j on property p as shown in Equation (9).

$$sim^p(e_i, e_j) = \frac{\sum_{n=1}^{t} w_{n,i,p} \times w_{n,j,p}}{\sqrt{\sum_{n=1}^{t} w_{n,i,p}^2} \times \sqrt{\sum_{n=1}^{t} w_{n,j,p}^2}} \tag{9}$$

where $w_{n,k,p}$ is the TF-IDF weight of entity e_k on property p and n^{th} attribute. And the distance between the two entities e_i and e_j is calculated by Equation (10).

$$sim(e_i, e_j) = \left(\sum_{p \in P} sim^p(e_i, e_j) \right) / \mid P \mid \tag{10}$$

where P is the set of properties of the unseen named entity. In summary, the steps of solving the problem of unseen named entity are as follows.

- Step 1: Based on user's reading history, link named entities of the articles and the unseen named entity onto the common knowledge base – DBpedia. If there is an equivalent entity of the unseen named entity in DBpedia, move to Step 2. In case there is no equivalent entity in DBpedia, return null.
- Step 2: Retrieve the number of properties of the linked entity of the unseen named entity in DBpedia, called dimensions.
- Step 3: For each dimension of the linked entity of the unseen named entity, compute the cosine similarity between every linked entity of named entity and linked entity of the unseen named entity by applying Equation (9).
- Step 4: For each linked entity of named entity, compute the distance between it and the linked entity of the unseen named entity by applying Equation (10).
- Step 5: Return the most similar entity among the linked entities of the named entities that has the smallest distance with the linked entity of the unseen named entity.

3.1.5. *Associative classifiers.* To produce class association rules, we employ the CBA algorithm for the CAR Miner unit with the predefined support threshold being set to low value (1%). The satisfied rules and their corresponding confidence values are populated into the CAR rule base for building associative classifiers.

Given a user u, an article p, its entity set and concept set are E_p and C_p, respectively. The associative classifier will find all rules, which were mined from profile of user u, in the rule base that fully and partially match the entity set and the concept set of article p. Then, the associative classifier sums up the confident values of all matched rules to

produce its recommended score as shown in Equation (13).

$$dE(u, E_p) = \sum_{r \in LRE_u} conf(r) - \sum_{r \in DRE_u} conf(r) \tag{11}$$

$$dC(u, C_p) = \sum_{r \in LRC_u} conf(r) - \sum_{r \in DRC_u} conf(r) \tag{12}$$

$$AC_{score}(u, E_p, C_p) = dE(u, E_p) + dC(u, C_p) \tag{13}$$

where $conf(r)$ is the confidence value of the matched rule r; LRE_u and DRE_u are the sets of matched CARs of entities with *Like* and *Dislike* labels, respectively; LRC_u and DRC_u are the sets of matched CARs of concepts with *Like* and *Dislike* labels, respectively.

3.2. **The collaborative filtering module.** To improve the recommendation efficiency, we deploy the user-based collaborative filtering module which aims at finding the like-minded readers of the current user and give recommendation based on their choices. The operation of this module composes two steps: (i) finding the like-minded readers who have the same rating patterns of the active users, and (ii) predicting the active user's interest about the new item based on the like-minded users' ratings.

In the first step, the users' rating patterns are used to find like-minded users. Because the rating value is binary (like and dislike), the Jaccard coefficient is used to measure the similarity between two different users u and v as shown in Equation (14).

$$similar(u, v) = \mid R_u \cap R_v \mid / \mid R_u \cup R_v \mid \tag{14}$$

where R_u is the set of articles liked by user u.

In the second step, given an articles p, the predicted rating value of user u on p is defined in Equation (15).

$$r_{u,p} = k \cdot \sum_{v \in U} similar(u, v) r_{v,p} \tag{15}$$

where $k = \left(\sum_{v \in U} |similar(u, v)| \right)^{-1}$ and U denotes the set of top N like-minded users of u who rated paper p.

3.3. **Recommendation engine.** The recommendation engine module uses a linear model to combine the decisions of the content-based module and the collaborative filtering module into a final decision. The linear model is shown in Equation (16).

$$predict(u, p) = kNN_{score}(u, p) + AC_{score}(u, E_p, C_p) + r_{u,p} \tag{16}$$

where u is an active user, p is an article, and $kNN_{score}(u, p)$, $AC_{score}(u, E_p, C_p)$ and $r_{u,p}$ are calculated by Equations (5), (13), and (15), respectively.

In addition, given an active user u and a candidate item set I which has $predict(u, p) > 0$, $\forall p \in I$, we rank the recommended items in descending order by incorporating the time feature of each article p. Equation (17) is used to produce the recommended list.

$$rank_{score}(u, p) = predict(u, p) + \lambda / (cur - pub) \tag{17}$$

where $\forall p \in I, predict(u, p) > 0$; λ is a predefined parameter; *pub* and *cur* are the published time of the paper p and the current time, respectively.

4. **Experiment.** Due to the intrinsic features of recommender systems, it is difficult to apply statistical-method to compare different news recommender systems with each other [34]. Hence, the state-of-the-art studies of news recommendation mainly compared their proposed methods with other traditional content-representation methods (e.g., TF-IDF, PLSI or LDA) [5, 22, 26, 37], revealed the advantages of the proposed method in different using scenarios [21], or compared the performances of different components to the proposed hybrid method [34]. In this section, in order to evaluate the performance of the semantically hybrid news recommender system, we adopted the comparison strategy of [34] and used TF-IDF as the benchmark. In this section we describe the data sets, the evaluation metrics, the experimental implementation and the results.

4.1. **Experiment settings.** The data sets used in this study were collected from well-categorized sources including Reuters (www.reuters.com), Yahoo News (news.yahoo.com), and CNN (www.cnn.com). Firstly, the articles were downloaded in html format. Then, the title, the abstract, the content, the published time and the author of the articles were retrieved while the other parts like advertisements, pictures and html tags, were removed. Thirdly, the retrieved contents were saved in text files. Finally, we obtained a result corpus containing 10,047 files.

Based on the original corpus, we processed and analyzed every text file to obtain the secondary data sets serving the content-based module at both term level and semantic level. First, to represent articles as bag-of-words model, we removed stop words and reduced the number of words by stemming. This task was done by applying Lucene[3] framework with its implementation of Porter algorithm [30] for stemming. The results were used to build TF-IDF vectors which will be utilized by the k-NN classifier. Second, the obtained text files were also used as input for the semantic annotation process which produces the entity set and concept sets of every article. The Weka framework [16], which is a collection of machine learning algorithms for data mining tasks, is used to build k-NN classifier and mine class association rules.

In order to give personalized news recommendations, the users' preferences were discovered in users' rating profiles which were used by both the content-based module and the collaborative filtering module. Because giving personalized recommendation is subjective to individual, we adopted the method of [34] which requires a number of users interacting with the system in order to collect users' rating data for the purpose of evaluating the news recommender system. Hence, we invited 12 students participating into the experiments, called users. The users used the proposed news recommender system in a period of 30 days. During this time, they were asked to read and rate the papers according to their personal interests. It is difficult to ask the users to read the whole contents of the articles, so we suggested them to read at least the title and the abstract of the paper before rating it. The rating data were collected to build the users' rating profiles which are in the form of user – item matrix where articles play the role of items.

The data sets were randomly divided into train sets and test sets with the proportion of 70% and 30%, respectively. While the content-based module used the train sets to build its classifiers, the collaborative filtering module used train sets to find like-minded users. The test sets were used to evaluate the system performance.

4.2. **Evaluation metrics.** With the purpose of generating top-K item-list from a given set of articles for personalizing news recommendation, we computed the recommended scores of every candidate articles by Equation (17). The higher recommended score of an article means that it is more relevant to the user's preferences. Then, the items were

[3]http://lucene.apache.org

ranked in the descending order of their recommended scores. Finally, the top-K items were selected to recommend.

According to [7], for evaluating the top-K recommended item-lists, the top 10, 20, and 30 news items are more valuable than the average evaluation of the overall news items. Therefore, to evaluate the performance of the proposed system, we considered the top *10, 20, 30, 40, 50* and *60* item-lists. These item-lists were generated from the candidate sets which were randomly selected from the test sets.

Although the precision and the recall metrics are often used to measure the accuracy of the information retrieval process, in this context, we have to evaluate different ranked item-lists. Therefore, an alternative evaluation metric should be taken into consideration. By concentrating on precision, we used the precision at K (P@K) metric to evaluate the ranked recommendation lists. The P@K of user u is calculated by Equation (18).

$$P@K_u = m/k \tag{18}$$

where k is the length of the recommended list and m is the number of true relevant items in the recommended list ($m \leqslant k$). There were different users in the experiment, so the average precisions at K were computed to measure the mean performance of the system.

4.3. **Experimental results and discussion.** For the target of generating top-K recommended lists, we randomly selected candidate set from all users' test sets. For each K value, the top-K recommended list was individually generated for each user from the same candidate set. Based on the obtained top-K recommendations, the P@K of each user was calculated by using Equation (18) and the average precisions were also computed.

Figure 5 shows the average P@K curves of the k-NN classifier, the content-based module which combines the prediction results of the k-NN classifier and the associative classifier, and the semantically hybrid approach which combines the prediction results of the content-based module and the collaborative filtering module. As can be seen from the graph, the prediction results of the only k-NN classifier, which also plays the role of benchmark

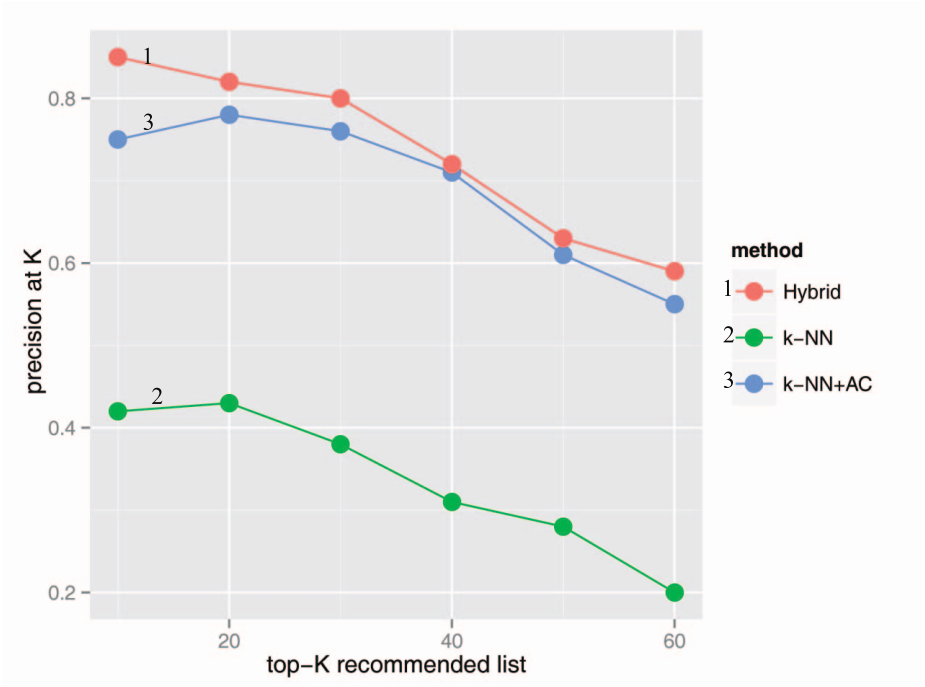

FIGURE 5. The average precision curves

line, are quite low with the highest value being 43% and the lowest value being 20% of the top-20 and top-60 recommended lists, respectively. However, the combination of the k-NN classifier and the associative classifier makes a dramatic change which raises the performance of the content-based module to the range $[55\%, 75\%]$. The comparison between the k-NN curve and the k-NN+AC curve reveals that for content-based approach, analyzing user's preferences at both term level and semantic level can discover user's interests better than analyzing user's preferences at term level only. Although there are several intersection points between the hybrid curve and the k-NN+AC curve, the hybrid curve, which presents the average precisions of the semantically hybrid approach, is still the dominant one. This implies that the combination of the collaborative filtering and the content-based approach can improve the recommendation performance.

For more details, the P@K values of every user obtained in the experiments, which measure the performance of the semantically hybrid approach, and the corresponding average P@K are shown in Table 1. The results are quite promising with the average P@K values of the top-10, top-20 and top-30 recommended lists being not less than 80%. For instance, in the best case, the proposed system delivered the top-10 recommended lists to the users A and C with the precision of 100%. The average P@10 reached the peak of 84% while the average P@20 was 82%. For P@30 values, although in the worst case the proposed system was only able to reach the precision of 73%, the average P@30 still maintained a quite high precision of 80%. From P@40 to P@60, the average P@K values steadily decreased from 72% to 59%. These results show that the true relevant items are likely to appear at the top of the recommended list.

Table 1. P@K values of the semantically hybrid approach

User	P@10	P@20	P@30	P@40	P@50	P@60
A	1	0.9	0.87	0.83	0.66	0.6
B	0.8	0.8	0.8	0.7	0.64	0.6
C	1	0.85	0.83	0.83	0.66	0.58
D	0.8	0.8	0.73	0.68	0.64	0.58
E	0.9	0.8	0.73	0.78	0.62	0.58
F	0.9	0.85	0.83	0.7	0.62	0.6
G	0.8	0.8	0.8	0.68	0.62	0.6
H	0.8	0.8	0.8	0.65	0.62	0.58
I	0.8	0.7	0.73	0.78	0.64	0.6
J	0.8	0.85	0.8	0.68	0.62	0.58
K	0.7	0.8	0.8	0.65	0.64	0.58
L	0.8	0.85	0.83	0.7	0.62	0.58
Average	0.84	0.82	0.80	0.72	0.63	0.59

5. **Conclusion.** In this paper, a semantically hybrid framework of personalizing news recommendation has been presented. The recommendation making is based on a linear model which combines the recommended scores of the content-based module and the collaborative filtering module. For the content-based module, the user's preferences are discovered at both term level and semantic level by the k-NN classifier and the associative classifier, respectively. At the semantic level, the prediction process is supported by the common knowledge base – DBpedia – in solving the problem of unseen named entity. The framework prototype has been validated with promising results which supported the deployment of the proposed framework to real-life application. To the application target,

the future work of this study is to deploy the proposed framework as the backbone of a news recommender system in mobile environment, in which, the technical improvements will include annotating named entity relationships in text and utilize users' behavior data in mobile application.

REFERENCES

[1] R. Agrawal and R. Srikant, Fast algorithms for mining association rules in large databases, *Proc. of the 20th International Conference on Very Large Data Bases*, San Francisco, CA, USA, pp.487-499, 1994.

[2] D. Billsus and M. J. Pazzani, A personal news agent that talks, learns and explains, *Proc. of the 3rd Annual Conference on Autonomous Agents*, New York, NY, USA, pp.268-275, 1999.

[3] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak and S. Hellmann, DBpedia – A crystallization point for the web of data, *Web Semantics: Science, Services and Agents on the World Wide Web*, vol.7, pp.154-165, 2009.

[4] I. Cantador and P. Castells, Semantic contextualisation in a news recommender system, *Workshop on Context-Aware Recommender Systems (CARS 2009)*, 2009.

[5] M. Capelle, F. Frasincar, M. Moerland and F. Hogenboom, Semantics-based news recommendation, *Proc. of the 2nd International Conference on Web Intelligence, Mining and Semantics*, New York, NY, USA, pp.27:1-27:9, 2012.

[6] M. Capelle, F. Hogenboom, A. Hogenboom and F. Frasincar, Semantic news recommendation using wordnet and bing similarities, *Proc. of the 28th Annual ACM Symposium on Applied Computing*, New York, NY, USA, pp.296-302, 2013.

[7] W. Chen, L. J. Zhang, C. Chen and J. J. Bu, A hybrid phonic web news recommender system for pervasive access, *WRI International Conference on Communications and Mobile Computing*, pp.122-126, 2009.

[8] S. Cleger-Tamayo, J. M. Fernández-Luna and J. F. Huete, Top-n news recommendations in digital newspapers, *Knowledge-Based System*, vol.27, pp.180-189, 2012.

[9] H. Cunningham, D. Maynard, K. Bontcheva and V. Tablan, GATE: A framework and graphical development environment for robust NLP tools and applications, *Proc. of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, 2002.

[10] A. S. Das, M. Datar, A. Garg and S. Rajaram, Google news personalization: Scalable online collaborative filtering, *Proc. of the 16th International Conference on World Wide Web*, New York, NY, USA, pp.271-280, 2007.

[11] T. Di Noia, R. Mirizzi, V. C. Ostuni, D. Romito and M. Zanker, Linked open data to support content-based recommender systems, *Proc. of the 8th International Conference on Semantic Systems*, New York, NY, USA, pp.1-8, 2012.

[12] F. Frasincar, W. I. Jntema, F. Goossen and F. Hogenboom, A semantic approach for news recommendation, *Business Intelligence Applications and the Web: Models, Systems and Technologies*, 2011.

[13] E. Gabrilovich, S. Dumais and E. Horvitz, Newsjunkie: Providing personalized newsfeeds via analysis of information novelty, *Proc. of the 13th International Conference on World Wide Web*, New York, NY, USA, pp.482-490, 2004.

[14] F. Goossen, W. IJntema, F. Frasincar, F. Hogenboom and U. Kaymak, News personalization using the CF-IDF semantic recommender, *Proc. of the International Conference on Web Intelligence, Mining and Semantics*, New York, NY, USA, pp.10:1-10:12, 2011.

[15] B. Hachey, W. Radford, J. Nothman, M. Honnibal and J. R. Curran, Evaluating entity linking with wikipedia, *Artificial Intelligence*, vol.194, pp.130-150, 2013.

[16] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten, The WEKA data mining software: An update, *SIGKDD Explor. Newsl.*, vol.11, pp.10-18, 2009.

[17] J. Han, J. Pei and Y. Yin, Mining frequent patterns without candidate generation, *Proc. of the 2000 ACM SIGMOD International Conference on Management of Data*, New York, NY, USA, pp.1-12, 2000.

[18] J. Hoffart, F. M. Suchanek, K. Berberich and G. Weikum, YAGO2: A spatially and temporally enhanced knowledge base from wikipedia, *Artificial Intelligence*, vol.194, pp.28-61, 2013.

[19] F. Hopfgartner and J. M. Jose, Semantic user profiling techniques for personalised multimedia recommendation, *Multimedia Systems*, vol.16, pp.255-274, 2010.

[20] T. Lavie, M. Sela, I. Oppenheim, O. Inbar and J. Meyer, User attitudes towards news content personalization, *International Journal of Human-Computer Studies*, vol.68, pp.483-495, 2010.
[21] H. J. Lee and S. J. Park, MONERS: A news recommender for the mobile web, *Expert Systems with Applications*, vol.32, pp.143-150, 2007.
[22] L. Li, D. Wang, T. Li, D. Knox and B. Padmanabhan, SCENE: A scalable two-stage personalized news recommendation system, *Proc. of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA, pp.125-134, 2011.
[23] Q. Li, J. Wang, Y. P. Chen and Z. Lin, User comments for news recommendation in forum-based social media, *Information Sciences*, vol.180, pp.4929-4939, 2010.
[24] W. Li, J. Han and J. Pei, CMAR: Accurate and efficient classification based on multiple class-association rules, *Proc. of IEEE International Conference on Data Mining*, pp.369-376, 2001.
[25] T. P. Liang, Y. F. Yang, D. N. Chen and Y. C. Ku, A semantic-expansion approach to personalized knowledge recommendation, *Decision Support Systems*, vol.45, pp.401-412, 2008.
[26] C. Lin, R. Xie, X. Guan, L. Li and T. Li, Personalized news recommendation via implicit social experts, *Information Sciences*, vol.254, pp.1-18, 2014.
[27] B. Liu, W. Hsu and Y. Ma, Integrating classification and association rule mining, *Proc. of the 4th KDD*, pp.80-86, 1998.
[28] A. Montes-García, J. M. Álvarez Rodríguez, J. E. Labra-Gayo and M. Martínez-Merino, Towards a journalist-based news recommendation system: The wesomender approach, *Expert Systems with Applications*, vol.40, pp.6735-6741, 2013.
[29] J. Pinho Lucas, S. Segrera and M. N. Moreno, Making use of associative classifiers in order to alleviate typical drawbacks in recommender systems, *Expert Systems with Applications*, vol.39, pp.1273-1283, 2012.
[30] M. F. Porter, An algorithm for suffix stripping, *Program: Electronic Library and Information Systems*, vol.14, pp.130-137, 1980.
[31] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom and J. Riedl, GroupLens: An open architecture for collaborative filtering of netnews, *Proc. of the 1994 ACM Conference on Computer Supported Cooperative Work*, New York, NY, USA, pp.175-186, 1994.
[32] F. M. Suchanek, G. Kasneci and G. Weikum, YAGO: A large ontology from wikipedia and WordNet, *Web Semantics: Science, Services and Agents on the World Wide Web*, vol.6, pp.203-217, 2008.
[33] V. Varma, P. Bysani, V. B. Kranthi Reddy, K. K. Santosh GSK, S. Kovelamudi, N. Kiran Kumar and N. Maganti, IIIT hyderabad at TAC 2009, *Proc. of Text Analysis Conference*, 2009.
[34] H. Wen, L. Fang and L. Guan, A hybrid approach for personalized recommendation of news on the web, *Expert Systems with Applications*, vol.39, pp.5806-5814, 2012.
[35] X. Yin and J. Han, CPAR: Classification based on predictive association rules, *Proc. of the 2003 SIAM International Conference on Data Mining*, SIAM, pp.331-335, 2003.
[36] M. J. Zaki, S. Parthasarathy, M. Ogihara and W. Li, New algorithms for fast discovery of association rules, *Technical Report*, University of Rochester, Rochester, NY, USA, 1997.
[37] L. Zheng, L. Li, W. Hong and T. Li, PENETRATE: Personalized news recommendation using ensemble hierarchical clustering, *Expert Systems with Applications*, vol.40, pp.2127-2136, 2013.