IJE TRANSACTIONS B: Applications Vol. 32, No. 8, (August 2019) 1101-1116

International Journal of Engineering

Journal Homepage: www.ije.ir

Automatic Hashtag Recommendation in Social Networking and Microblogging Platforms Using a Knowledge-Intensive Content-based Approach

M. Jaderyan, H. Khotanlou*

Department of Computer Engineering, Bu-Ali Sina University, Hamedan, Iran

PAPER INFO

Paper history:
Received 23 April 2019
Received in revised form 23 May 2019
Accepted 05 July 2019

Keywords:
Content Enrichment
Hashtag Recommendation
Knowledge-Intensive
Ontology
Semantic Network Representation
Structured Knowledge Base

ABSTRACT

In social networking/microblogging environments, #tag is often used for categorizing messages and marking their key points. Also, since some social networks such as twitter apply restrictions on the number of characters in messages, #tags can serve as a useful tool for helping users express their messages. In this paper, a new knowledge-intensive content-based #tag recommendation system is introduced. The proposed system works by integrating structured knowledge in every core component. First, the relevant features, semantic structures and information-content are extracted from messages. Since little information can often be placed in a message, a content enrichment module is introduced to identify information structures that can improve the representation of message. The extracted features are represented by semantic network. Then, a hybrid and multi-layered similarity module identifies the commonalities and differences of the features, semantics and information-content in messages. At the end, #tags are recommended to users based on #tags in contextually similar messages. The system is evaluated on *Tweets2011* dataset. The results suggests that the proposed method can recommend suitable #tags in negligible operational time and when little content is available.

***doi**: 10.5829/ije.2019.32.08b.06*

1. INTRODUCTION

The microblogging environments in social networks have become the main source of preferred/favourite information and breaking news. They have also transformed into platforms for communicating with friends and celebrities for many people. These platforms are being employed to acquire and analyze information in applications such as public opinion mining [1, 2], prediction based on public opinions [3], individual reputation scoring and management [1], natural language processing [4], privacy protection [5], event detection [6] and interest and expertise mining [7]. Twitter is one of the most important and widely used social networking platforms [8]. What distinguishes twitter from other social networking platforms is the 280-character limit on the messages (tweets). This unique feature makes it difficult for users to convey their intended message by a single tweet. The adaptation of #tags for marking the key points of messages is another interesting feature of twitter. Since in most cases, the intended meaning of a tweet cannot be stated clearly by 280-character limit, using #tags will help users express their message better. Over the last couple of years, numerous approaches have been introduced for #tag recommendation in microblogging/social networking platforms. The majority of these approaches can be classified in three categories: (1) content-based [9–12], (2) collaborative filtering-based [10, 13] and (3) machine learning-based approaches [14]. The approaches in the first category calculate the similarity of a user's message to the stored messages in a database and recommend #tags to users based on the contextual similarity between them. However, since users' messages are usually short and they cannot convey the intended meaning clearly, it would be very difficult to find suitable #tags [10]. The approaches in the second category analyze the preferences of collaborative users and recommend #tags based on the opinion of users that share common interest with the active user. These approaches suffer from

*Corresponding Author Email: *khotanlou@basu.ac.ir* (H. Khotanlou)

Please cite this article as: M. Jaderyan, H. Khotanlou, Automatic Hashtag Recommendation in Social Networking and Microblogging Platforms Using a Knowledge-Intensive Content-Based Approach, International Journal of Engineering (IJE), IJE TRANSACTIONS B: Applications Vol. 32, No. 8, (August 2019) 1101-1116

serious drawbacks, which arise from challenges such as the cold start phenomenon, data sparsity, scalability and ambiguity (when dealing with synonyms). The approaches in the third category train a learning model using the microblogging data. The biggest drawback of these approaches is the vast amount of predefined classes in the training data, which results in heavy computational burden on the system.

The success of a #tag recommendation model depends on the generated representation of messages. Over the last decade, the vector space representation models have gone through fundamental changes. The shallow word embedding approaches can model the content of textual resources in a continuous space called word2vec [15]. In this space, the co-occurrence relation and semantic pattern between concepts are identified and modelled. These models are then used to develop semantically enhanced techniques for calculating similarity between word sequences. Methods such as skip-gram [16] and continuous bag-of-words (CBOW) [17] are among word2vec-based modelling techniques. Advanced methods such as Recurrent Neural Network and Convolutional Neural Networks are among the newly developed approaches that employ word-embedding techniques [18–20].

The main contributions of this paper are: (1) introducing a knowledge-intensive #tag recommendation system by integrating the structured knowledge of ontology, Wikipedia and WordNet in every component of the proposed method. (2) Introducing a novel content enrichment module for improving the representation of messages. (3) Introducing a novel weighting mechanism for optimized representation of content in semantic networks. (4) Using the graphical structure of semantic networks to model the semantics and information-content of messages. (5) Introducing a novel hybrid and multi-layer semantic similarity measure for identifying the shared information-content and commonalities/ differences in semantic features, structural features and semantic relations of two semantic networks.

The rest of the paper is structured as follows: in the second section, the related works are explored. In the third section, the research objectives are declared. In the fourth section, the structure of the Top-Level ontology and KBs is introduced. In the fifth section, the proposed method of #tag recommendation is introduced. In the sixth section, the evaluation results are presented and the conclusions are presented in the seventh section.

2. RELATED RESEARCHES

Many of the studies in this area analyze the information-content of the messages, extract prominent features of the content, and use the extracted features to recommend #tags [21]. Otsuka et al. [9] introduced a new ranking method called HF-IHU, which is a variant of the well-known TF-IDF method. This method considers data sparseness and #tag relevance as the deciding factors for #tag recommendation. Gong et al. [22] proposed bag-of-phrase model for better capturing of the underlying topics of posted microblog. Then, online Twitter-User LDA [22] is used to learn Twitter users' dynamic interests. This method uses incremental bi-term topic model (IBTM) to discover the latent topics of tweet content. These methods are then combined to generate #tag recommendations.

Most #tag recommendation approaches employ a representation schema to model the contents (tweets). The vector space models (VSMs) are among the most popular representation schemas. Zangerla et al. [23] used TF-IDF model for #tag recommendation. They used the TF-IDF vector model to find the most similar messages to user's tweet. The continuous space models (word embedding) were formed to address the intrinsic drawbacks of VSMs. Over the last couple of years, word-embedding techniques have been used successfully for #tag recommendation [15]. Mikolov et al. [24] in google proposed two novel models for computing continuous vector representations of words from very large data sets. The designed models enable the system to learn the distributed representations of words. Weston et al. [25] introduced a convolutional neural network for #tag recommendation. The proposed method learns the feature representations of short textual resources (posts, messages, sentences, …) by using #tags as a supervised signal. Gong and Zhang [26] have used the convolutional neural networks for #tag recommendation. The authors have proposed a novel architecture with an attention mechanism to calculate embedding for each word of a given message. The collective embedding is then used to model the whole message. A novel recurrent neural network model to learn vector-based tweet representations is proposed to recommend #tags [15]. A skip-gram model is used to generate distributed word representations and a convolutional neural network to learn semantic sentence vectors. Finally, the sentence vectors are used to train a LSTM-RNN network. The tweet vectors are used as features to classify #tags.

Another form of message representation can be achieved using the collaborative filtering-based models. Gong and Zhang [26] used a collaborative filtering-based model to represent the messages and rank #tags. To this end, they used the message content and the embedded links to recommend #tags based on user preferences. Chen et al. [27] focussed on a neighborhood-based recommender system for recommending URLs to users in twitter. They used #tags as the topic representatives of URLs. Then, a correlation measurement is used to find the candidate #tags in similar tweets. A personalized method for #tag recommendation based on topical information and collaborative intelligence is introduced

by Tomar et al. [10]. The topic relevance of hashtags to posts are characterized based on content models. Active user's hashtag usage preference is predicted in a collaborative filtering manner.

Graph-based modelling of microblogging messages has also been used as a representation schema for #tag recommendation. Al-Dhelean et al. [28] introduced a heterogeneous social graph model that contains information about users, tweets, and #tags. The graph is then summarized to a #tag graph that shows the similarity between different #tags. At the end, the vertices in the graph are ranked using a random walk with restart and a content-based similarity measure. The researches in this field suggest that using a graph-based approach for content representation has positive impact on overall performance. A graph-coarsening approach that aims to speed-up the execution time of graph-based tag recommenders in large-scale folksonomies is introduced by Wang et al. [11]. A community detection algorithm in multiplex networks is applied for coarsening the hyper-graph depicting a folksonomy and #tag recommendation.

Learning-based approaches can be employed to model the tweets and corresponding #tags. Tomar et al. [10] used a skip-gram model to learn distributed word representations (word2vec) of tweets. They used the learned model to train a deep feed forward neural network for #tag recommendation. Probabilistic models [28], Topic Analysis [4] and RNN-LSTM networks [29] are also used for modelling and recommending #tags in microblogging platforms [30, 31].

Inspired by the researches in this field, we have exploited the structured knowledge of KBs and Top-Level Ontology to develop a content enrichment module, to represent messages in graphical structure of semantic network and to implement a hybrid semantic similarity measure for #tag recommendation.

3. THE STRUCTURE OF KBS

The KBs play a crucial role in optimal performance of the proposed method. Understanding their structural characteristics underlines what kind of semantic information/structures are identifiable and extractable.

3. 1. OntoWordNet The OntoWordNet ontology is an integral part of the system. Each concept in ontology is organized as synonym set so that the contextually similar (or equivalent) concepts can be identified. This facilitates content enrichment [13].

3. 2. WordNet WordNet models a semantically enhanced lexicon for English language. The structure of WordNet consists of synsets. The synset organizes a set of synonym concepts. Every synset consists of several senses. The senses are simply the different meanings of a concept. More details about WordNet is available in literature [13].

3. 3. Wikipedia Wikipedia data are available for academic use through D.I.S.C.O project [32]. Both data are structured the same way. The manner in which the data are created is described in literature [32, 33]. Wikipedia data structure consist of two sets of data: (1) first-order word vector: which contains words that occur together in Wikipedia and BNC corpus and (2) second-order word vector: which contains words that occur in similar contexts. more information is available in literature [32, 33].

4. PROPOSED METHOD

The proposed method utilizes the structured knowledge of KBs and ontology to extract the semantics and informative features from messages. The extracted features will be used for content enrichment, content representation and semantic similarity computation. The proposed system recommends #tags by analyzing the contents of user's message and measuring its semantic and contextual similarity to the messages in a database. Figure 1 illustrates the core components of the proposed method.

The structured knowledge of ontology and KBs are integrated in every component of the proposed system. The proposed method comprises of four components: (1) Data preparation and pre-processing module: this module is tasked with extraction of relevant features, semantic structures and information-content form messages. (2) Representation module: this module uses the extracted information and features about a message to model the information-content in graphical structure of semantic networks. (3) Semantic similarity module: this module is

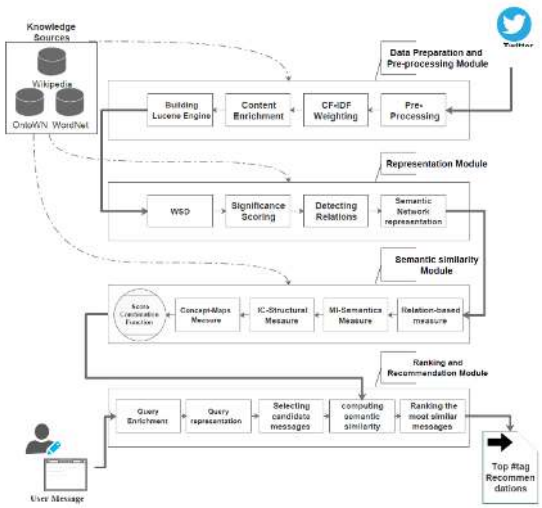

Figure 1. The core components of the proposed method

tasked with finding the most similar messages to an input message. (4) Ranking and recommendation module: this module ranks the messages according to their similarity to the user message and determines the most suitable #tags. As illustrated in Figure 1, the input of the proposed system is a database of social media messages. At first, the messages undergo a pre-processing operation and the semantic structures and information-content features are extracted. In the next step, the features (concepts/words) are weighted using CF-IDF weighting schema. The most challenging part of analyzing contents is the lack of sufficient content. A novel content-enrichment module is introduced to overcome this obstacle. The feature vectors are then enriched to enhance the representation of information-content. For accurate modelling of the relation between features, a Word Sense Disambiguation (WSD) technique is used to annotate features with their true (contextual) meanings. In the next step, the features and the enriched contents [34] are indexed using the Apache Lucene [35] to create a full-text indexing search engine. Since the extracted features are not equally important, a hybrid weighting mechanism is introduced to select the most important features (concepts/words) for content representation. Then, the semantic network structure is used to represent the features and the enriched contents. As the user enters a message into the system, Lucene search engine identifies and retrieves a number of similar messages. Then, a novel hybrid semantic similarity module is introduced for identifying the shared information-content and semantics between the semantic networks of the retrieved messages and semantic network representation of user input. The #tags in the contextually similar messages are marked as #tag recommendation candidates. Finally, the ranking mechanism ranks the computed #tag recommendation candidates and recommends suitable #tags to the user.

4. 1. Data Preparation and Pre-processing

Recommending #tags based on users' tweeting behavior and the content of tweets would require a database of pre-analyzed tweets, which would be the basis for #tag recommendation. We have used the "Tweets2011" dataset to generate this database. "Tweets2011" dataset contains identifiers for approximately 16 million tweets sampled between January 23rd and February 8th, 2011. The corpus is designed to be a reusable, representative sample of the twitter sphere [36]. In order to collect the tweets, tweet identifiers are retrieved from the dataset and the original tweets are fetched using third-party libraries [37]. From the nearly 16 million tweets in the dataset, 10,099,860 randomly selected tweets were retrieved and stored for further processing.

The pre-processing steps include: (1) Removing tweets with no #tags, (2) Extracting and Storing #-tags, (3) Removing All Non-English Messages, (4) Removing User IDs, (5) Removing Re-Tweets, (6) Removing All URLs, (7) Removing Punctuations and non-alphanumerical symbols, (8) Removing Stop-Words, (9) Lower-case Transformation, (10) Lemmatization of textual entities and (11) uni-gram and Bi-gram identification. After pre-processing, 952,416 tweets out of 10,099,860 tweets were remained in the database.

Next, all the detected uni-gram and bi-gram concepts are weighted using CF-IDF weighting method (which is a variant of TF-IDF weighting method) [38].

4. 2. Enriching the Content of the Messages

4. 2. 1. Enriching the Content Using Wikipedia KB

Employing structured knowledge bases for improving the representation of textual resources can help the system understand the context and semantics in textual resources [39]. Wikipedia KB contains two sets of information namely, the co-occurring and the contextually similar concepts/words. We are proposing to use the first and second-order word vectors for enriching the content of messages. The enrichment module facilitates the identification of lexical/semantic features that are essential for system understanding of the context. For each concept/word in message vector, Apache Lucene Library [40] is used to retrieve the co-occurring and contextually similar concepts/words. The retrieved concepts/words will be appended to the corresponding message vector.

4. 2. 2. Enriching the Content Using WordNet KB

One of most important information structures in WordNet is the contextually similar sets. These sets determine which concepts are contextually similar/related to one another. Appending the contextually similar sets to the message vector helps system detect the messages with shared information-content. MIT's Java WordNet interface (JWI) [41] is used to retrieve the contextually similar set for each concept/word. Each retrieved set will be appended to the corresponding message vector.

4. 2. 3. Enriching the Content Using OntoWordNet Ontology

The semantic structure of "concept maps" is used to enrich the message content. The OntoWordNet ontology is organized in a way that each class (representing a concept) forms a synonym set (synset). After projecting the content of a message onto OntoWordNet ontology, the corresponding classes to each concept/word are identified. Next, a "concept map" represents each concept/word. A constructed concept map consists of a concept/word and a set of corresponding ontology classes. The links between the concept and the classes are the "equivalent" property and the "subclass" relation. The equivalent property is transitive and reversible. Concept maps are represented by a sub-ontology using OWL/XML schema. The superclass and the equivalent concepts are weighted and

appended to the message vector. An example of concept map for "news" is illustrated in Figure 2.

The "concept maps" are used to annotate the message semantic networks and to infer new links between concepts. Incorporating the enriched content for improving the representation model and system understanding of context is a novel idea presented here.

Definition 1. Let $d_i$be a message vector, then:

$d_i = \{c_i^1, c_i^2, c_i^3, \dots\dots, c_i^n\}\rightarrow$ a message vector

$c_i^{wiki,WN,OntoWN} = \{c_i^{e1}, c_i^{e2}, c_i^{e3}, \dots\dots, c_i^{en}\}\rightarrow$ enrichment vector for d_i

$d_i^{enriched} = \{d_i \cup c_i^{wiki,WN,OntoWN}\} = \{c_i^1, c_i^2, c_i^3, \dots\dots, c_i^n, c_i^{e1}, c_i^{e2}, c_i^{e3}, \dots\dots, c_i^{en}\}\rightarrow$ extended message vector for d_i

$w_i = \{w_i^1, w_i^2, w_i^3, \dots\dots, w_i^n\}\rightarrow$ a message weight vector

$w_i^{wiki,WN,OntoWN} = k_{extended} * w_i = \left\{w_{i_{i=1,,,n}}^{wiki,WN,OntoWN} \middle| w_i^{wiki,WN,OntoWN} = k_{extended} * w_{i_{i=1,,,n}}\right\} \rightarrow$ enriched message weight vector ($k_{extended} = 0.6$)

$w_i^{enriched} = \{w_i \cup w_i^{wiki,WN,OntoWN}\}\rightarrow$ extended message weight vector

4. 3. Word Sense Disambiguation of the Concepts

Before we can model the relations between concepts, we need to clear the concepts/words of ambiguity. Therefore, a word sense disambiguation (WSD) technique, inspired by the idea presented in literature [42], has been developed. The underlying assumption is that similar senses occur in similar contexts. In other words, by comparing the collective informational and contextual features of a concept with the information-content of each possible senses, we can induce its true contextual meaning. The developed WSD technique relies heavily on the structured knowledge of Wikipedia and WordNet. The following step are performed: (1) A ± 7 context window around the desired concepts in the message is created. In addition, the first-order word vector for each member of the context window is retrieved and appended to the context window. The window and the appending vectors create a "context vector" for each concept; (2) all possible senses of the concept, their usage example in a sentence and their brief definition (gloss) for each sense are extracted from WordNet. This will form a "sense vector" for each sense. The first-order word vector for each member of a sense vector is also retrieved and appended to corresponding sense vector. Finding the similarity between each sense vector and the context vector determines the contextual similarity between them; (3) a combination of cosine and Jaro-Winkler [43] measures is used to calculate the similarity score as follows.

$$Sim(Sense_{Vctor}, context_{Vector}) = \frac{1}{2}(Cosine_{Sim}(Sense_{Vctor}, context_{Vector}) + Jaro\_winkler_{Sim}(Sense_{Vctor}, context_{Vector})) \quad (1)$$

The sense vector with highest similarity score is selected as the correct sense vector and the corresponding sense is used to annotate the concept. The output of this module is a set of weighted concepts that are annotated by their contextual meaning.

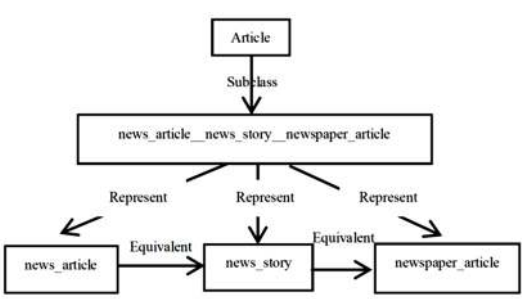

Figure 2. Representation of a generated Conceptual Map

4. 4. Building Full-Text Search Engine

Apache Lucene (Solr) Java library is used to index the messages and the enriched contents in the dataset and build a search engine. The generated Lucene's search indexes allows easy and fast access to hundreds of thousands of stored messages. When evaluating the proposed method, the indexed dataset will be split into training and testing subsets. These subsets will be used to evaluate the performance and estimate parameters.

4. 5. Message Content Representation Using Graphical Structure of Semantic Network

The graphical structure of semantic network organizes the concepts/words and the relations between them, which helps the system understand the context and semantic structures in textual resources. OntoWordNet ontology is employed to establish semantic relations between concepts/words. The concepts/words in messages are projected onto the OntoWordNet ontology. In the next step, the following semantic relations are used to establish connections between concepts: (1 and 2) superclass/subclass relation: Assuming that two concepts x_i and x_j are given, if the concept x_i categorizes the concept x_j, then x_i is called superclass of x_j and x_j is called subclass of x_i. (3) Synonymy relation: Assuming that x_i is a concept/word in the document, if we can find a concept/word x_j and replace it with the x_i so that the informational context of the document does not change, it can be said that x_i and x_j are connected by Synonymy relation. (4) Part_of relation: The Part_of relation represents the part-whole relationship between the concepts/words. The Part_of relation is established between concepts x_i and x_j, if x_j is essentially part of x_i. In other word, the presence of x_j implies the existence of x_i. However, the presence of x_i does not indicate the

presence of x_j. This relation was not originally a part of OntoWordNet ontology, but it has been added by aligning DBpedia ontology [44] and related datasets with OntoWordNet ontology. The aforementioned relations are used to link the concepts/words in semantic networks.

Definition 2. Let $\boldsymbol{d_i^{enriched}}$ be an extended message vector for message d_i, then:

$E = \{Part_{of}, Superclass, subclass, synonym\} \rightarrow$ a set of relations

$V = d_i^{enriched} = \{d_i \cup c_i^{WN,Wiki,OntoWN}\} \rightarrow$ an extended message vector

$Graph\ G = \{(v_i, e_j, v_i) | v_i \in V,\ e_j \in E\} = \{\bigcup_{i=1,\dots n}^{j=1,2,3,4}(v_i, e_j, v_i)\}$

$Graph\ \bigcup_{i=1,\dots,n}^{j=1,2,3,4} G_{n=1\dots m} = \{(v_i, e_j, v_i)^1, (v_i, e_j, v_i)^2, \dots\dots, (v_i, e_j, v_i)^k\}_{k=number\ of\ trip}$

$Links = \{\forall (v_i, e_j, v_i) \in G_n\ \exists\ link_i | link_i \leftrightarrow (v_i, e_j, v_i)\} \rightarrow$ relation can be expressed as a triple

$Graph\ G_{msg}\ (message) = \{(v_i, e_j, v_i)_{msg} | v_i \in V,\ e_j \in E\} = \{\bigcup_{i=1,\dots n}^{j=1,2,3,4}(v_i, e_j, v_i)_{msg}\}$

$Graph\ G_{user}(user\ message) = \{(v_i, e_j, v_i)_{user} | v_i \in V,\ e_j \in E\} = \{\bigcup_{i=1,\dots n}^{j=1,2,3,4}(v_i, e_j, v_i)_{user}\}$

Not all the concepts/words contribute equally to the information-content of the message. Therefore, only concepts that make greater contribution to the information-content should participate in semantic network generation process. Thus, a novel measure is introduced to identify concepts that are irrelevant to the context.

4. 5. 1. Contribution Score for Semantic Networks

"Contribution scoring function" is a novel hybrid measure that calculates amount of "cohesion" each concept/word has with the "information-content" and "context" of a message. Four criteria are considered for calculating the contribution score: (1) lexical cohesion, (2) co-occurrences cohesion, (3) semantic cohesion and (4) Hierarchical path proximity.

Lexical Cohesion: calculates the lexical cohesion of each concept in a message with all the remaining concepts in the same message by measuring Jaro-Winkler [43] similarity between them. This will give us the idea of how much each lexical entity is compatible with the information-content of message.

$$Sim_{JaroWinkler}(a_i, b_i) = d_j + lp(1 - d_j) \quad a_i \neq b_i \tag{2}$$

$$d_j = \begin{cases} 0 & if\ m = 0 \\ \frac{1}{3}\left(\frac{m}{|a_i|} + \frac{m}{|b_i|} + \frac{m-t}{m}\right) & otherwise \end{cases} \tag{3}$$

where d_j is the computed Jaro similarity score for the concepts a_i and b_i. l_p is the length of common prefix between two concepts. m is the number of matched characters and t is half the number of characters displacements between two concepts.

Semantic Cohesion: calculates the semantic cohesion of each concept in the message with all the remaining concepts in the same message by measuring Lin [34] similarity between them.

$$Sim_{Lin}(a_i, b_i) = \frac{2*IC(LCS(a_i,b_i))}{(IC(a_i)+IC(b_i))} \quad a_i \neq b_i \tag{4}$$

where, IC is the information-content and is computed as $IC(a) = -\log p(a)$ using WordNet KB [34].

Hierarchical Path Proximity: calculates the shortest hierarchical path between two concepts in OntoWordNet ontology [34, 35]:

$$Sim_{WuPalmer}(a_i, b_i) = max\left[\frac{2*Depth(LCS(a_i,b_i))}{Length(a_i,b_i)+2*Depth(LCS(a_i,b_i))}\right] \quad a_i \neq b_i \tag{5}$$

Co-Occurrences Cohesion: calculates the degree of shared information-content between a concept in a message and all the remaining concepts in the same message.

$$Sim_{Co-occurrence}(a_i, b_i) = \frac{2*freq(a_i,b_i)}{freq(a_i)+freq(b_i)} + \mu * \frac{2*freq(b_i,a_i)}{freq(a_i)+freq(b_i)} \quad a_i \neq b_i \tag{6}$$

where, μ is a controlling parameter between [0, 1], $freq(a_i, b_i)$ is the frequency of b_i co-occurring with a_i and $freq(b_i, a_i)$ is the frequency of a_i co-occurring with b_i. Also, $freq(a_i)$ is the frequency of concept a_i.

At the end, a weighted linear combination of these measures determines the contribution score of each concept:

$$contribution\_score(a_i) = \begin{cases} 1 & if\ \sum_{j=1,\dots N-1; i\neq j} \frac{significance\_score(a_i,b_j)}{N} \\ 0 & O.W. \end{cases} \tag{7}$$

$$significance\_score(a_i, b_j) = k_1 * Sim_{JaroWinkler}(a_i, b_j) + k_2 * Sim_{lin}(a_i, b_j) + k_3 * Sim_{WuPalmer}(a_i, b_j) + k_4 * Sim_{Co-occurrence}(a_i, b_j) \tag{8}$$

where, N is the number of concepts and a_i is the respective concept. Also, k_1, k_2, k_3 and k_4 are weighting parameters. They are between [0, 1] and their sum is equal to 1. These parameters are estimated using a subset of evaluation data in the evaluation stage. In the next step, top-n% of the concepts (with highest contribution score) are used to generate the message semantic network. The top-80% seems to be the optimal percentage of concepts for generating message semantic network. This number is estimated using a subset of evaluation data.

4. 5. 2. Semantic Networks Generation Process

The first step is to project the top-n% contributing concepts/words onto OntoWordNet ontology so that

semantic relations are established between them. When only the original concepts (excluding enriched content) are projected onto the ontology, in most cases several separated clusters of concepts are formed. Two reason can explain this occurrence: (1) because of the lack of sufficient content in the messages, it is only natural that separated clusters of concepts/words are formed and (2) the concepts/words that can connect the separated clusters are left out. However, when the extended message vectors are projected onto OntoWordNet ontology, in most cases, a fully connected graphical representation of message is formed. The enriched concepts/words that can connect the separated clusters are called "liaison concepts". Therefore, the enrichment module is one of the most important component of the proposed method and integrating this module provides two major benefits for the system: (1) identifying liaison concepts, which connects the separated clusters and (2) improving the information-content and representation of the messages. The following steps are performed to generate a message semantic network: (1) the top-n% of the concepts/words are selected according to the contribution score. (2) The selected concepts/words are projected onto the ontology and the proposed algorithm (Figure 4) identifies the relations between them one by one and links them to one another. (3) The liaison concepts connect the separated concept clusters to form a fully connected semantic network. Figure 3 illustrates how the semantic network connects the concepts and how the liaison concepts connect the separated clusters.

As shown in Figure 3, "info", "story", "television_news" and "newscast" act as the "liaison" concepts for connecting the separated clusters and connecting the semantic network to concepts in the deeper hierarchical structure of ontology. The semantic network will be represented as a sub-ontology using the OWL/XML schema. Therefore, the semantic network is machine-readable. The algorithm for generating Semantic network is depicted in Figure 4.

4. 6. Semantic Similarity Module The proposed

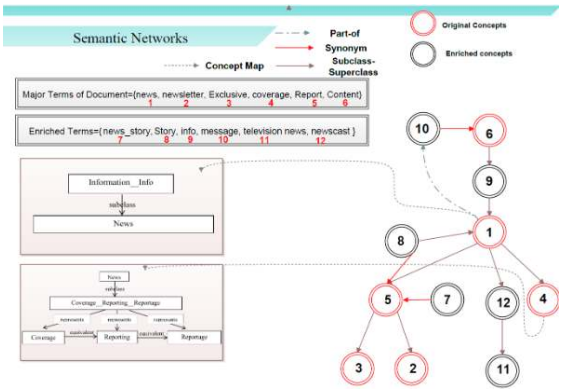

Figure 3. Semantic network

Input: a set of messages $D = \{d_1, d_2, d_3, \ldots\ldots, d_n\}$, each document $d_i^{enriched} = \{c_i^1, c_i^2, c_i^3, \ldots\ldots, c_i^n\}$

- Loop: for each concept in $d_i^{enriched}$
  - Loop: until $d_i^{enriched}$ is empty
    - Condition: if semantic network is empty
      - Append the first concept to the semantic network and Delete the first Concept from $d_i^{enriched}$.
    - End of Condition
    - Relation-Path set = a set indicating concepts with relations to c_i^n and their path length; relative to c_i^n in ontology
    - Relation={concepts with relation to c_i^n}, Path={path length of each Relation}
    - Loop: for each c_i^n that already exists in the semantic network
      - Loop: for each c_j^n in the $d_i^{enriched}$
        - Condition: if the relation between c_i^n and c_i^m is in Relation-location set & path-length>1
          - Source= c_i^n, Destination=c_i^m
          - Find every concept in a path from source t Destination
          - Disregard if the path already exist in the semantic network
          - Connect the Destination concept of the respective relation to Source concept in the semantic network
          - Add "Destination" to the semantic network and Remove the "Destination" from $d_i^{enriched}$
          - Delete Relation and its Path-length from sets Relation and Path
        - End of Condition
      - End of Loop
    - End of Loop
    - Condition: if there are relations in set Relation with Path-Length=1
      - Connect c_i^n and c_i^n via superclass/subclass relation
    - End of Condition
  - End of loop
- End of Loop

Output: the generated semantic network for the $d_i^{enriched}$,

Figure 4. Algorithm for generating Semantic network

Semantic similarity module relies heavily on the structured knowledge of ontology, Wikipedia and WordNet. As mentioned earlier, accurate modelling of semantics and information-content is vital to the optimal performance of the proposed method. The proposed semantic similarity module is a hybrid and multi-layer one. It is designed to consider the similarities and differences in semantics, information-content, relations and other related features for computing similarity between two semantic networks. The proposed semantic similarity computation method consists of **I**. Relation-based measures**, II.** Semantics-based measures**, III.** IC-Structural measures and **IV.** Concept Maps-based Measure.

4. 6. 1. Relation-based Measure

Explicit Measure: calculates the amount of shared information-content between two semantic networks. This measure determines which message semantic network is similar to semantic network representation of user's message.

$$Score_{Explicit}\left(\bigcup(\boldsymbol{v_i}, \boldsymbol{e_j}, \boldsymbol{v_i})_{\boldsymbol{msg}}\right) = \frac{\sum_{all\ the\ triplets} Sim_{\exp(v_i, e_j, v_i)_{msg}}}{|all\ triplets\ in\ the\ msg\ semantic\ net.|} \tag{9}$$

$$Sim_{\exp(v_i, e_j, v_i)} = \begin{cases} \delta_{exp}, & if\ \boldsymbol{v_i}\ and\ \boldsymbol{v_j}\ are\ in\ \bigcup(\boldsymbol{v_i}, \boldsymbol{e_j}, \boldsymbol{v_i})_{\boldsymbol{user}} \\ 1 - \delta_{exp} & o.w. \end{cases}$$

Implicit Measure: calculates the similarity between two semantic networks by measuring the commonalities in the established relations. In other words, this measure determines how much a message semantic network resembles a user message semantic network.

$$Score_{implicit}\left(\bigcup(\boldsymbol{v_i}, \boldsymbol{e_j}, \boldsymbol{v_i})_{\boldsymbol{msg}}\right) = \frac{\sum_{all\ the\ triplets} Sim_{\mathrm{imp}(v_i, e_j, v_i)_{msg}}}{|all\ triplets\ in\ the\ msg\ semantic\ net.|} \tag{10}$$

$$Sim_{\exp(v_i, e_j, v_i)} = \begin{cases} \delta_{imp}, & if\ (\boldsymbol{v_i}, \boldsymbol{e_j}, \boldsymbol{v_i}) \in \bigcup(\boldsymbol{v_i}, \boldsymbol{e_j}, \boldsymbol{v_i})_{\boldsymbol{user}} \\ 1 - \delta_{imp} & o.w. \end{cases}$$

where, δ_{exp} and δ_{imp} are thresholds between [0, 1] and $\boldsymbol{v_i}$and $\boldsymbol{v_j}$ are concepts in user message and a message in the database respectively. This measure generates a number between [0, 1] indicating the similarity score.

4. 6. 2. Semantics-based Measures

WordNet-based Measure: This method is based on the notion of information-content (IC) of the Least Common Subsumer (LCS) [34]. The notion of IC and the level of shared IC between concepts can be considered as a measure for calculating semantic similarity between two messages. Higher level of shared IC indicates high level of semantic similarity and higher similarity score. To this end, the normalized Jiang and Conrath measure is employed [32, 35].

$$WN\_sim(\boldsymbol{v_i}, \boldsymbol{v_j}) = j\&c(\boldsymbol{v_i}, \boldsymbol{v_j}) = 1 - \left(\frac{\left[IC_{nrm}(\boldsymbol{v_i}) + IC_{nrm}(\boldsymbol{v_j}) - 2 * IC_{nrm}\left(LCS(\boldsymbol{v_i}, \boldsymbol{v_j})\right)\right]}{2}\right) \tag{11}$$

where, $\boldsymbol{v_i}$and $\boldsymbol{v_j}$ are concepts in user message and a message in the database respectively. The IC of $\boldsymbol{v_i}$ is $IC(\boldsymbol{v_i}) = -\log p(\boldsymbol{v_i})$ and is calculated using WordNet.

Wikipedia-based Measure: The semantic similarity between two concepts can be viewed as function of similarities/differences between their respective Wikipedia's first-order and second-order word vectors. To this end, a measure inspired by Lin's information theoretic measure [45, 46] has been implemented. This measure is called Collocation-Contextual similarity measure. At first, the first-order and second-order vectors corresponding to two concepts are retrieved. Then the following equations are used to calculate the similarity:

$$Wiki\_sim(v_i, v_j) = \frac{\sum_{(v_M, rel_j, v_N)}^{j=1,2;N=1,\dots n;M=1,\dots,m} (I(v_M, rel_j, *) + I(v_N, rel_j, *))}{\sum_{(v_M, rel_j, v_N)}^{j=1,2;N=1,\dots n;M=1,\dots,m} (I(v_M, rel_j, v_N)) + \sum_{(v_M, rel_j, v_N)}^{j=1,2;N=1,\dots n;M=1,\dots,m} (I(v_N, rel_j, v_M))}$$

$$rel_j = \{rel_1, rel_2\} = \{co - occurrence\_rlation, Contextual\_similarity\_Rel \tag{12}$$

$$I(v_M, rel_j, v_N) = log \frac{freq(v_M, rel_j, v_N) * freq(v_N, rel_j, v_M)}{freq(v_M, rel_j, *) * freq(*, rel_j, v_N)}$$

where, $\boldsymbol{v_i}$ and $\boldsymbol{v_j}$ are concepts in user message and a message in the database respectively. $I(v_M, rel_j, v_N)$ is equal to the mutual information between v_M and v_M. The $freq()$ function calculates the frequency of $\boldsymbol{v_M}$ and $\boldsymbol{v_M}$ in co-occurrence/contextually_similar relations based on Wikipedia KB.

4. 6. 3. IC-Semantics-Structural Measures

IC-Structural Measure: measures commonalities in structural and IC-based features of two concepts in two semantic networks:

$$sim\_IC\_Stc(v_i, v_j) = \frac{IC(LCS(v_i, v_j))}{IC\left(LCS(v_i, v_j)\right) + \left(\lambda(v_i, v_j) * \left(IC(v_i) - IC\left(LCS(v_i, v_j)\right)\right)\right) + ((1 - \lambda(v_i, v_j)) * (IC(v_j) - IC(LCS(v_i, v_j))))}$$

$$\lambda(v_i, v_j) = \begin{cases} \frac{depth(v_i)}{depth(v_i) + depth(v_j)} & depth(v_i) \le depth(v_j) \\ 1 - \frac{depth(v_i)}{depth(v_i) + depth(v_j)} & O.W \end{cases} \tag{13}$$

where, $\boldsymbol{v_i}$ and $\boldsymbol{v_j}$ are concepts in user message and a message in database respectively and $depth(\boldsymbol{v_i})$ calculates the depth of $\boldsymbol{v_i}$ in hierarchical structure of ontology. IC of $\boldsymbol{v_i}$ is $IC(\boldsymbol{v_i}) = -\log p(\boldsymbol{v_i})$ and is calculated using WordNet. $\lambda(\boldsymbol{v_i}, \boldsymbol{v_j})$ is a normalizing factor and is a function of depth of v_i and v_j in ontology.

Semantics Measure: measures commonalities in corresponding first-order (co-occurrence) and second-order (contextually similar) vectors of two concepts in two semantic networks:

$$sim\_semantis(\boldsymbol{v_i}, \boldsymbol{v_j}) = \frac{1}{3}\left(\frac{|co-ocr(\boldsymbol{v_i}) \cap co-ocr(\boldsymbol{v_j})|}{|co-ocr(\boldsymbol{v_i}) \cup co-ocr(\boldsymbol{v_j})|} + \frac{|cntx(\boldsymbol{v_i}) \cap ccntx(\boldsymbol{v_j})|}{|cntx(\boldsymbol{v_i}) \cup cntx(\boldsymbol{v_j})|} + \frac{|syns(\boldsymbol{v_i}) \cap syns(\boldsymbol{v_j})|}{|syns(\boldsymbol{v_i}) \cup syns(\boldsymbol{v_j})|}\right) \tag{14}$$

where, $co - ocr(\boldsymbol{v_i}), cntx(\boldsymbol{v_i}), syns(\boldsymbol{v_i})$ are the corresponding first-order (Wikipedia), second-order (Wikipedia) and synonym vector (OntoWordNet Ontology) of concept $\boldsymbol{v_i}$.

In the semantics-based and the IC-Semantics-Structural-based measures, the notion of semantic similarity between two concepts is used to compute the similarity between two semantic networks. These methods compute the semantic similarity between all the possible pairs of concepts in user message and a message in database and generate a number between [0, 1].

4. 6. 4. Concept Maps-based Measure This measure calculates the similarities between concepts maps of two different semantic networks. Each concept in a semantic network is annotated by a concept map, which provides useful insight about the concepts. Finding the commonalities between concept maps gives us an idea of how much two semantic networks are similar. The following equations are used to calculate the similarity:

$$sim\_concept\_maps(\boldsymbol{G_{user}}, \boldsymbol{G_{msg}}) = \frac{\sum_{v_i \in G_{msg_4}} \frac{1}{4}(Hypo(v_i)+Hyper(v_i)+equi(v_i)+Partof(v_i))}{|v_i \in \boldsymbol{G_{msg}}|}$$

$$Hypo(\boldsymbol{v_i}) = \begin{cases} \propto & if\ (\boldsymbol{v_i}, Hyponym, *) \in \boldsymbol{G_{user}} \\ 1-\propto & if\ (*, Hyponym, \boldsymbol{v_i}) \in \boldsymbol{G_{user}} \\ 0 & O.W \end{cases}$$

$$Hyper(\boldsymbol{v_i}) = \begin{cases} \propto & if\ (\boldsymbol{v_i}, Hypernym, *) \in \boldsymbol{G_{user}} \\ 1-\propto & if\ (*, Hypernym, \boldsymbol{v_i}) \in \boldsymbol{G_{user}} \\ 0 & O.W \end{cases} \quad (15)$$

$$equi(\boldsymbol{v_i}) = \begin{cases} \propto & if\ (\boldsymbol{v_i}, equivalen, *) \in \boldsymbol{G_{user}} \\ 1-\propto & if\ (*, equivalen, \boldsymbol{v_i}) \in \boldsymbol{G_{user}} \\ 0 & O.W \end{cases}$$

$$partof(\boldsymbol{v_i}) = \begin{cases} \propto & if\ (\boldsymbol{v_i}, partof, *) \in \boldsymbol{G_{user}} \\ 1-\propto & if\ (*, partof, \boldsymbol{v_i}) \in \boldsymbol{G_{user}} \\ 0 & O.W \end{cases}$$

where $\propto$ is thresholds between [0, 1]. This method generate a number between [0, 1] indicating the similarity score.

At the end, a linear weighted combination of the introduced measures is used to compute the overall semantic similarity.

$$overall\_sim\_score(\boldsymbol{G_{user}}, \boldsymbol{G_{msg}}) = k_1 * \frac{\sum_{v_i \in G_{user}}^{v_j \in G_{msg}, w_j \in w_j^{enriched}} \boldsymbol{w_j} * WN\_sim(v_i, v_j)}{\sum_{w_j \in w_j^{enriched}} \boldsymbol{w_j}} + k_2 * \frac{\sum_{v_i \in G_{user}}^{v_j \in G_{msg}, w_j \in w_j^{enriched}} \boldsymbol{w_j} * Wiki\_sim(v_i, v_j)}{\sum_{w_j \in w_j^{enriched}} \boldsymbol{w_j}} +$$

$$+ k_3 * \frac{\sum_{v_i \in G_{user}}^{v_j \in G_{msg}, w_j \in w_j^{enriched}} \boldsymbol{w_j} * sim\_semantis(v_i, v_j)}{\sum_{w_j \in w_j^{enriched}} \boldsymbol{w_j}} + k_4 * \frac{\sum_{v_i \in G_{user}}^{v_j \in G_{msg}, w_j \in w_j^{enriched}} \boldsymbol{w_j} * sim\_IC\_Stc(v_i, v_j)}{\sum_{w_j \in w_j^{enriched}} \boldsymbol{w_j}} + + k_5 * Score_{Explicit}\left(\bigcup(\boldsymbol{v_i}, \boldsymbol{e_j}, \boldsymbol{v_i}) \in \boldsymbol{G_{msg}}\right)$$

$$+ k_6 * Score_{implicit}\left(\bigcup(\boldsymbol{v_i}, \boldsymbol{e_j}, \boldsymbol{v_i}) \in \boldsymbol{G_{msg}}\right) + k_7 * sim\_concept\_maps(\boldsymbol{G_{user}}, \boldsymbol{G_{msg}}) \quad (16)$$

where, $k_1, k_2, k_3, k_4, k_5, k_6$ and k_7 are the weighting parameters between [0, 1] and their sum is equal to 1. These parameters are estimated using a subset of evaluation data in evaluation stage. So far, the result is a set of ranked message and a set of #tag candidates. The next step is to rank the #tag candidates and recommend the top-N #tags to users.

4.7. #Tag Ranking and Recommendation Module In order to rank the top-N #tag candidates, a suitable ranking method needs to be introduced to rank, select and recommend the best #tags to user.

Ranking based on the similarity score: the score of similarity between messages can be used to rank the #tags for recommendation. It is logical to assume that if the user message and a message in database are contextually similar, the message in database probably contains #tags that are appropriate for recommendation to user. In circumstances where several messages may contain the same #tag, the #tags in messages with higher scores will be ranked higher. Figure 5. Illustrates the overall #tag recommendation algorithm.

5. EVALUATION

5. 1. Evaluation Setup *"Tweets2011"* dataset (described in section 5.1) is the basis for creating the message database, implementing the search and recommendation engine and evaluating the proposed method. The Leave-One-Out method is used to evaluate the proposed #tag recommendation system. A total of test runs are performed and the average of test results is used

Input: the sets of messaged $D = \{d_1, d_2, d_3, \ldots\ldots, d_n\}$, Lucen search engine, user message, KBs

- Construct the user message semantic network based on the entered user message
- Loop: for the constructed user message semantic networ
  - Retrieve a number of messages comprising similar content.
- End of Loop
- Loop: for each retrieved message
  - Construct the semantic network
- End of Loop.
- Compute the semantic similarity between user message and semantic network representation of other messages in DB.
- Rank most similar messages according to their similarity to user message.
- Rank the #tag recommendation candidates in the most similar messages based on scores generated by hybrid measure.

Output: #tag recommendation for social networ users/microblog users.

Figure 5. The overall algorithm for #tag recommendation

to evaluate the performance. In each test run, 95,240 input messages (nearly 10% of the indexed messages) containing maximum of eight #tags are retrieved randomly (from Lucene search index) for evaluation. From the left-out data, ten thousand messages are selected randomly for parameter optimization. These messages are different from the evaluation data. Obviously, we remove #tags for every input message. In addition, the input message is removed from the Lucene index so that it does not distort the evaluation results. Since nearly a million recommendations are computed and evaluated; therefore, the evaluation setup is comprehensive and it generates a good assessment of system performance.

5. 1. 1. Similar Methods for Comparison

I. The method introduced in literature [27] is considered to compare the system performance with similar methods. It is a straightforward method based on term frequency-inverse document frequency (tf/idf). The following equations are used to compute the similarity:

$$tf\_idf_{t,d} = tf_{t,d} * idf_t \tag{17}$$

$$tf_{t,d} = n_{t,d} \tag{18}$$

$$idf_t = \log \frac{|D|}{|\{d:t \in d\}|} \tag{19}$$

This equation consists of two parts. The *term_frequency* part counts the number of occurrences of term *t* in a given message *d* and *inverse_document _frequency* determines the significance of term *t* in all the messages.

II. Another method has been developed to assess the performance of proposed method. This method exploits lexical, semantics-based and structural-based features of messages to determine the similarity between two different messages. This method consists of three measures: (1) cosine similarity (2) Information-content - based similarity (3) Hierarchical path-based similarity**.** The following equations are used to calculate the similarity between two messages:

$$Cosine(d_m^{enriched}, d_n^{user}) = \frac{d_m^{enriched}.d_n^{user}}{\|a_m^{enriched}\| \|Bd_n^{user}\|} = \frac{\sum_{v_i \in d_m^{user}}^{v_j \in d_m^{enriched}} v_i x v_j}{\sqrt{\sum_{v_i \in d_m^{enriched}} (v_i)^2 x} \sqrt{\sum_{v_j \in d_m^{user}} (v_j)^2}} \tag{20}$$

$$Ic - based(v_i, v_j) = IC_{seco}(LCS(v_i, v_j))$$

$$IC_{seco}(v_i) = \left(1 - \frac{\log(hypo(v_i)+1)}{\log(\text{Hypo\_root})}\right) \tag{21}$$

$$Path\_based(v_i, v_j) = \frac{1}{\min_{\forall p} Hierarchical\_Path_p(v_i, v_j)} \tag{22}$$

where $d_m^{enriched}$ represents an extended message vector, d_n^{user} represents extended user message vector, $hypo(v_i)$ is the number of hyponyms for v_i and Hypo_root is the number of hyponyms for the root node of ontology. The final similarity score is calculated as follows:

$$Hybrid\ (sim(d_m^{enriched}, d_n^{user})) = k_1 * Cosine(d_m^{enriched}, d_n^{user}) + k_2 * \frac{\sum_{v_t \in d_n^{user}}^{v_j \in d_m^{enriched}, w_j \in w_j^{enriched}} w_j * Ic-based_{sim(v_t, v_j)}}{\sum_{w_j \in w_j^{enriched}} w_j} + k_3 * \frac{\sum_{v_t \in d_n^{user}}^{v_j \in d_m^{enriched}, w_j \in w_j^{enriched}} w_j * Path\_based_{sim(v_t, v_j)}}{\sum_{w_j \in w_j^{enriched}} w_j} \tag{23}$$

where k_1, k_2 and k_3 are the weighting parameters between [0, 1] and their sum is equal to 1. Also, $w_j^{enriched}$ represents extended message weight vector.

5. 2. Overcoming High Dimensionality of Messages

Because of the huge number of messages in the dataset, it is not computationally efficient to calculate the similarity between user message and all the messages in the database. Thus, we used the embedded "more_like_this" module in the Apache Lucene library. This module allows users to query and retrieve documents in the database that are similar to an arbitrary document. For each user message, this module retrieves two hundred similar message from the dataset and their similarity to the user message is calculated. The following settings are used to evaluate the performance of the proposed method and its components.

As mentioned in section 5.6, for each input message, similar tweets are identified and the top-N #tags are recommended. In addition, the performance of the proposed method in recommending top-N #tags (where N=1, N=2... and N=10) is evaluated.

5. 3. Evaluation Metrics

The *Precision* and *Recall* metrics are used to evaluate the proposed method:

$$Precision(Hashtag_{recommendation}) = \frac{|Recommended_{Hashtags} \cap Original_{Hashtags}|}{|Recommended_{Hashtags}|} \tag{24}$$

$$Recall(Hashtag_{recommendation}) = \frac{|Recommended_{Hashtags} \cap Original_{Hashtags}|}{|Original_{Hashtags}|} \tag{25}$$

where, $Recommended_{Hashtags}$ are the Top-N recommended #tags and $Original_{Hashtags}$ are the original #tags, which were removed from the input message.

5. 4. Evaluation Results

5. 4. 1. Evaluating the Performance of Proposed Method in Recommending #Tags to Users

The recall and precision values for top-N recommendations are illustrated in Figures 6 and 7. The *similarity_function* is used as the basis for #tag ranking. The evaluation results suggest that the proposed similarity measure

outperforms the tf/idf and hybrid similarity method. It also performs significantly better even when only small number of #tag recommendations are computed. The proposed method yields higher recall and precision. Two possible reasons can explain the results: (1) the representation module is comprehensive. It incorporates the lexical, semantic, syntactical, information-content and structural features in the proposed model. (2) The proposed semantic similarity module exploits the extracted features for analyzing the content, finding the commonalities/differences between messages and finally computing the similarity between them.

As depicted in Figure 7, the precision value decreases as the number of recommended #tags is increased. The proposed method is evaluated on a portion of database that contain maximum eight #tags. Therefore, even if system performs perfectly in terms of recall, the maximum performance in terms of precision will be 80%. Therefore, 2 out of 10 #tag recommendations are not admissible. In addition, two other issues contribute to the issue of low precision values: (1) the messages are shorts and the number of features in messages are low (Max. 9 features). (2) Some messages do not contain any features and contain only #tags.

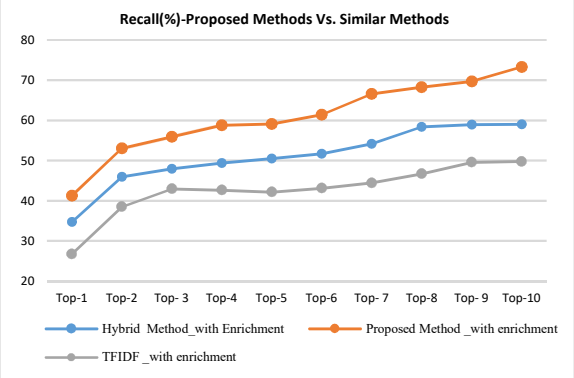

Figure 6. The recall values for the proposed #tag recommendation method compared to other similar methods

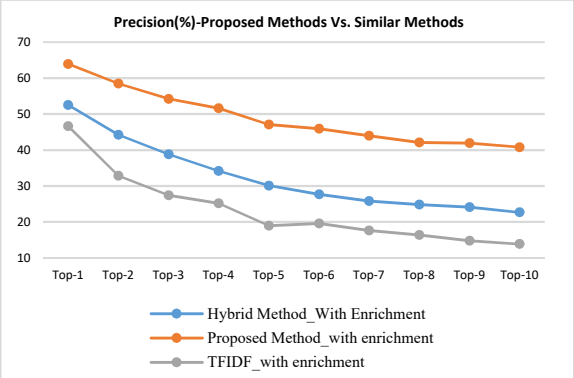

Figure 7. The Precision values for the proposed #tag recommendation method compared to other similar methods

5. 4. 2. Evaluating the Enrichment Module and Its Effect on the Overall Performance of the Proposed Method and Similar Methods In this section, we are trying to determine the overall effect of this module on the performance of the proposed method. The results are illustrated in Figures 8 and 9. As depicted, coupling the proposed method with content enrichment module results in higher precision and recall values. According to results, adding a content enrichment module to the #tag recommendation systems has a positive effect on the performance and accuracy. In addition, the results indicate that coupling a semantic similarity measure with content enrichment modules will results in better understanding of the context and further improvement in the quality of generated #tags. It should be noted that even when the N (the number of recommended #tags) is small, the proposed method coupled with enrichment modules scores promising recall and precision values. Also, the proposed method is capable of recommending suitable #tags even when little content is available.

5. 4. 3. Evaluating the Semantic Similarity Module and Its Effect on Overall Performance of the Proposed Method The accuracy of the proposed method heavily depends on the representation technique and the manner in which the commonalities/difference between messages are measured.

The proposed similarity module considers all available information (lexical, semantic, syntactical, information-content-based, latent and structural features) for computing similarity. Therefore, it is essential to determine the efficacy of the module and its performance. As illustrated in Figures 10 and 11, the proposed method performs significantly better in terms of precision and recall when the proposed similarity module is applied. Also, the precision values, especially when the N is small, indicates that top recommendations by the proposed method are significantly better compared to the situation when the hybrid similarity measure is applied.

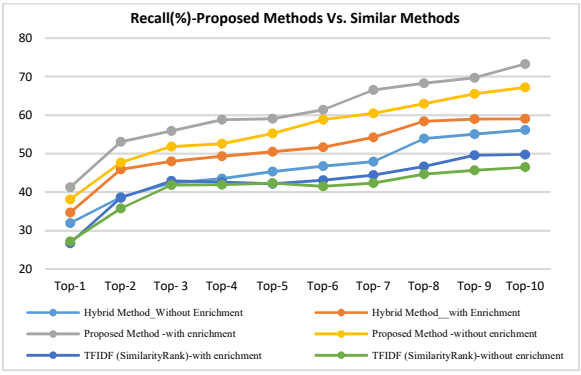

Figure 8. The Recall values for evaluating the overall effect of the enrichment module on the proposed and similar methods

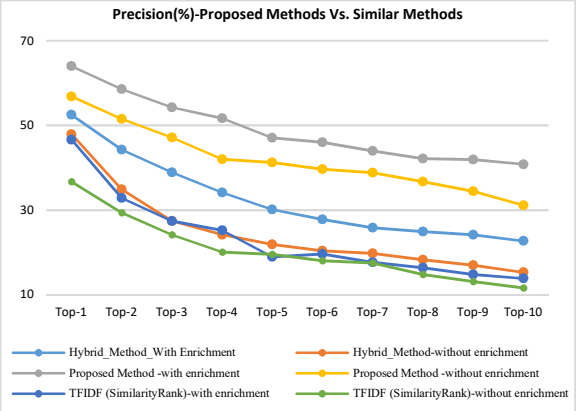

Figure 9. The Precision values for evaluating the overall effect of the enrichment module on the proposed and similar methods

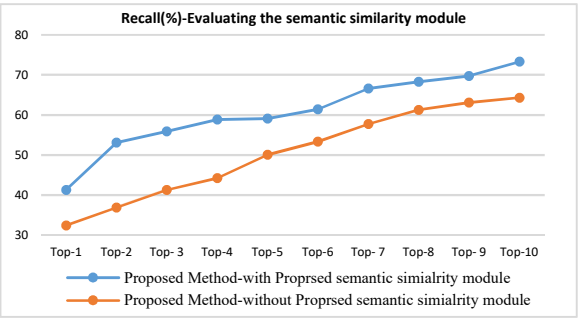

Figure 10. The Recall values when evaluating the overall effect of the semantic similarity module on the proposed method

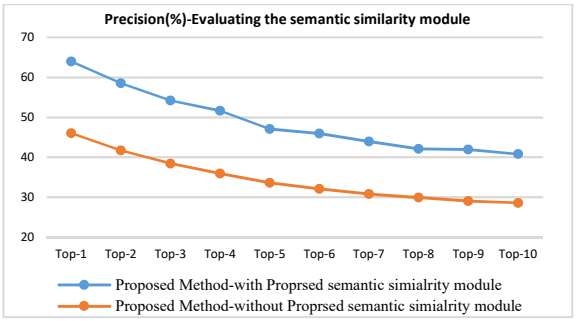

Figure 11. The Precision values when evaluating the overall effect of the semantic similarity module on the proposed method

In addition, the recall values suggest that when the proposed similarity module recommends a small number of #tags (N is small); they are more accurate. The reason is that the similarity module considers all the available information about messages.

5. 4. 4. Evaluating the Contribution Scoring Function and Its Effect on Overall Performance Next step is to evaluate the effectiveness of contribution scoring function in selecting the best concepts for semantic networks. To this end, we have replaced this scoring function with the CF-IDF weighting schema. The results are illustrated in Figures 12 and 13. The result suggest that the contribution scoring function does a better job in identifying the concepts that are essential in representing the context and information-content.

In the next step, the components of the contribution scoring function and their effect on the overall performance are evaluated. The evaluation results indicate that lexical cohesion metric performs poorly compared to semantic and Hierarchical cohesion metrics. In other words, knowledge-based metrics are better suited for features selection in representation module. In addition, the performance of lexical cohesion metric is somewhat similar to the CF-IDF weighting method. The co-occurrence cohesion metric achieves the best results among the components of the contribution scoring function. This is because it requires more semantic analysis of context compared to other metrics. The results are illustrated in Figure 14.

5. 4. 5. Evaluating the Semantic Network Representation and Its Effect on Overall Performance The optimal accuracy and performance of proposed method depends on accurate

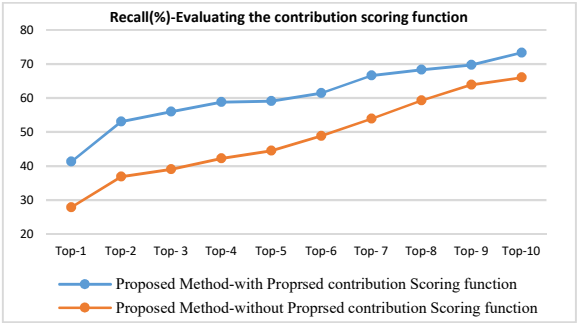

Figure 12. The Recall values for evaluating the overall effect of the contribution scoring function on the proposed method

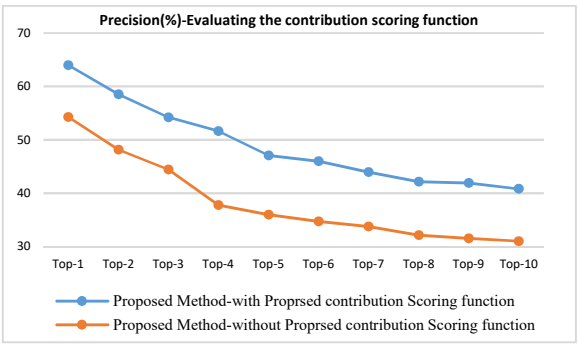

Figure 13. The Precision values for evaluating the overall effect of the contribution scoring function on the proposed method

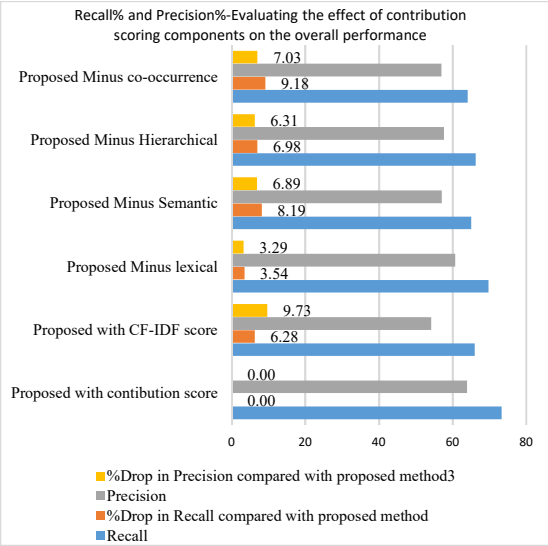

Figure 14. The Recall and Precision values when the effect of contribution scoring components on the overall performance

and comprehensive representation of semantics and information-content in messages. Therefore, it is essential to determine the efficacy of the semantic network representation module in modelling the information-content. The evaluation settings are depicted in Table 1. The results are illustrated in Figures 15 and 16. Illustrated results suggest that the proposed representation module is capable of accurate modelling of information-content and semantics in messages.

TABLE 1. Evaluation settings

Methods	Properties
The Proposed Method	All the properties of the proposed method are preserved
The Proposed Method-without semantic network representation	Messages are represented by Vector Space Models; only the WordNet and Wikipedia-based Enrichments are used, only portions of semantic similarity module that are not based on the relation between concepts are employed

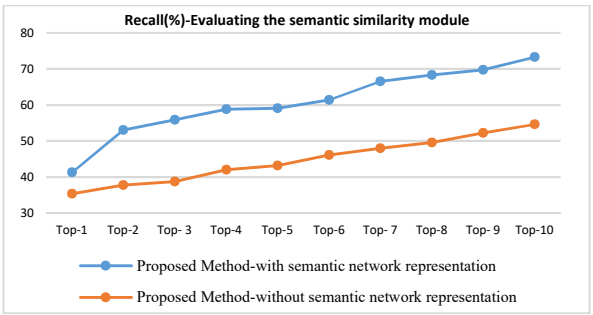

Figure 15. The Recall values for evaluating the overall effect of the representation module on the proposed method

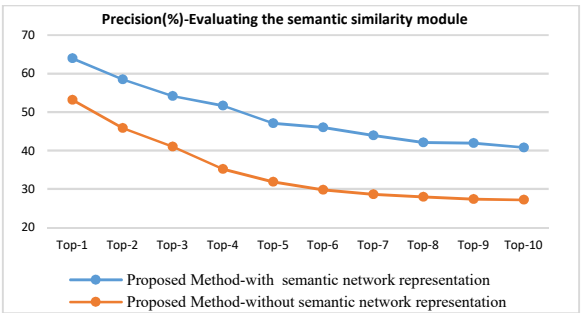

Figure 16. The Precision values for evaluating the overall effect of the representation module on the proposed method

It also contributes positively to the accuracy and precision of proposed method, even when only a small number of #tags are recommended. The representation module is very effective in capturing the semantic features and structures in messages. It can help the system identify messages that are similar to user's input message.

5. 4. 6. Evaluating the Components of the Proposed Similarity Modules and their effect on Overall Performance The representation module is only effective when there is an effective semantic similarity module for calculating the similarity between the messages. Therefore, the precision of the similarity module is just as important as the comprehensiveness of the representation module. In this section, a number of experiments are prepared to evaluate the components of proposed similarity module. The settings of designed experiments are illustrated in Table 2.

The illustrated results suggest that among the components of similarity module, the IC_Structural-based components have the greatest effect on precision and efficiency of the proposed method. The concept_map_based and semantic_based measures are in the second and third place, respectively. The results

TABLE 2. The settings of designed experiments

#	Experiment	Description of the Experiment
1	Proposed	All the components of proposed similarity module
2	Proposed – IC_Structural_based	The proposed similarity module without IC_Structural_based scoring
3	Proposed - Semantic_Wiki	The proposed similarity module without Semantic_Wiki scoring
4	Proposed - Semantic_WordNet	The proposed similarity module without Semantic_WordNet scoring
5	Proposed - Concept_Maps_based	The proposed similarity module without Concept_Maps_based scoring
6	Proposed - Relation-based	The proposed similarity module without Relation-based scoring

demonstrate the superiority of measures that exploit the semantic-based, structural-based and information-content -based features to draw comparison between messages. Therefore, the structured knowledge of ontology, WordNet and Wikipedia makes them the perfect tool for computing semantic similarity (see Figures 7 and 8).

5. 4. 7. Evaluating The System Based on The Required Operational Time In the next step, the mean operational time required to build the representation model, to enrich the content, to find the contextually similar messages and to recommend #tags is computed and compared. The results are illustrated in Table 3.

The results suggest that the proposed method requires negligible operational time compared with other methods considering that the proposed method requires significant

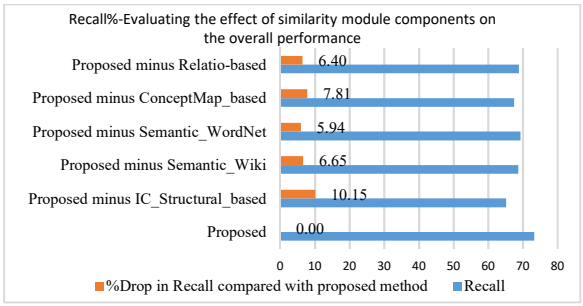

Figure 17. The Recall% when Evaluating the Similarity module

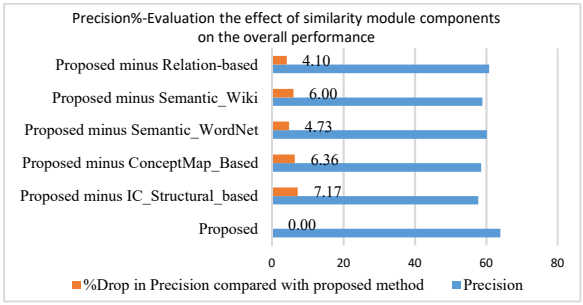

Figure 18. The Precision% when Evaluating the Similarity module

TABLE 3. mean operational time (computed for 10 random messages and then averaged)

Stages of Operation	TFIDF	Hybrid	Proposed Method
Pre-processing and preparation (offline)	5ms	5ms	5ms
representation model(offline)	17ms	17ms	89ms
Content enrichment(offline)	19ms	19ms	36ms
Similarity computation and ranking (offline)	15ms	74ms	216ms

computational operations for identifying and extracting information structures from KBs. Also, the results suggest that the proposed method can be used to provide online recommendations to active users. Although the enrichment module imposes negligible operational time, it has positive effect on the overall performance of system.

6. CONCLUSION

In this paper, a new knowledge-intensive and content-based #tag recommendation approach for social networking platforms was introduced. The proposed system was evaluated on the *tweets2011* dataset. The evaluation results suggest that the proposed system demonstrates robust results in terms of recall and precision values. In addition, it is capable of assigning appropriate #tags to messages without #tags and can recommend suitable #tags even when little content is available in messages. This is due to the integration of multiple knowledge sources into the components of the proposed system. This system does not rely solely on the content of the messages, but also identifies latent semantic features and structures using the structured knowledge and exploit all available information about messages to recommend accurate #tags. Moreover, the evaluation of content enrichment module suggests that the proposed method coupled with enrichment module results in higher performance. When little information is available about the content, the enrichment module allows the system to identify related information or semantic structures. The significance of this module lies in the fact that the majority of messages contain little information. Therefore, a mechanism is needed to extend the system understanding of context. The results support the assumption that content enrichment play a pivotal role in optimal performance of the proposed method. The evaluation of representation module indicates the superiority of proposed method compared with vector space models. The robustness of the representation model in modelling the extracted features, enriched contents and semantics is the strength of this model. It presents the system with wide range of information and enables the system to make an informed decision. In addition, the proposed hybrid and multi-layer semantic similarity module exhibits higher accuracy and precision compared with other similarity measures. The multi-layered nature of this module guarantees that system can detect commonalities and differences in all types of extracted features and semantic structures between messages. In other words, it guarantees that all available information about messages are effectively contributing in calculating the similarity between messages. Moreover, similarity methods that exploit semantics-based, information content-based and structural-based features to draw comparison between messages exhibit better

performance and can be used as reliable tools for computing semantic similarity between textual resources. However, the semantic similarity module requires more computational time in comparison with similar methods. Although, based on the volume of computations, the recorded operational time is logical and is suitable for implementing an online system, it has led us to do more to reduce the computational time. In essence, integrating structured knowledge of ontology and KBs is instrumental in robust and optimal performance of the proposed #tag recommendation method. As a future work, we are determined to reduce the computational time in semantic similarity module. In addition, we are planning to use the word-embedding techniques and deep learning methods for feature extraction. As a future work, we are trying to use the proposed system in other text mining applications to determine whether the proposed system is a multi-purpose framework. The illustrated results do not necessarily mean that this method is the best method for #tag recommendation. It just can be considered as a successful implementation of a knowledge-based recommendation system, suitable for #tag recommendation and other text mining applications where little content is available for decision-making.

7. REFERENCES

1. Gong, Y., Zhang, Q., and Huang, X., "Hashtag recommendation for multimodal microblog posts," ***Neurocomputing***, Vol. 272, (2018), 170–177.
2. Bermingham, A. and Smeaton, A. F., "Classifying sentiment in microblogs: is brevity an advantage?," In Proceedings of the 19th ACM International Conference on Information and Knowledge Management - CIKM '10, ACM Press, (2010), 1833–1836.
3. Bollen, J., Mao, H., and Zeng, X., "Twitter mood predicts the stock market," ***Journal of Computational Science***, Vol. 2, No. 1, (2011), 1–8.
4. Izadi, S. and Ghasemzadeh, M., "Using Generalized Language Model for Question Matching," ***International Journal of Engineering - Transactions C: Aspects***, Vol. 26, No. 3, (2012), 241–244.
5. Mohammadi, A. and Hamidi, H., "Analysis and Evaluation of Privacy Protection Behavior and Information Disclosure Concerns in Online Social Networks," ***International Journal of Engineering - Transactions B: Applications***, Vol. 31, No. 8, (2018), 1234–1239.
6. Sakaki, T., Okazaki, M., and Matsuo, Y., "Earthquake shakes Twitter users: real-time event detection by social sensors," In Proceedings of the 19th International Conference on World Wide Web - WWW '10, ACM Press, (2010), 851–860.
7. Becker, H., Naaman, M., and Gravano, L., "Learning similarity metrics for event identification in social media," In Proceedings of the Third ACM International Conference on Web Search and Data Mining - WSDM '10, ACM Press, (2010), 291–300.
8. Guy, I., Avraham, U., Carmel, D., Ur, S., Jacovi, M. and Ronen, I., "Mining expertise and interests from social media," In Proceedings of the 22nd International Conference on World Wide Web - WWW'13, ACM Press, (2013), 515–526.
9. Otsuka, E., Wallace, S. A., and Chiu, D., "Design and evaluation of a Twitter hashtag recommendation system," In Proceedings of the 18th International Database Engineering & Applications Symposium on - IDEAS '14, ACM Press, (2014), 330–333.
10. Tomar, A., Godin, F., Vandersmissen, B., De Neve, W. and Van de Walle, R., "Towards Twitter hashtag recommendation using distributed word representations and a deep feed forward neural network," In International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, (2014), 362–368.
11. Wang, Y., Qu, J., Liu, J., Chen, J. and Huang, Y., What to Tag Your Microblog: Hashtag Recommendation Based on Topic Analysis and Collaborative Filtering, Springer, Cham, (2014), 610–618.
12. Hmimida, M. and Kanawati, R., A graph-based meta-approach for tag recommendation, Springer, Cham, (2017), 309–320.
13. Ding, Z., Qiu, X., Zhang, Q. and Huang, X., "Learning topical translation model for microblog hashtag suggestion," In Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, AAAI Press, (2013), 2078–2084.
14. Kywe, S.M., Hoang, T.A., Lim, E.P. and Zhu, F., On Recommending Hashtags in Twitter Networks, Springer, Berlin, Heidelberg, (2012), 337–350.
15. Li, J., Xu, H., He, X., Deng, J. and Sun, X., "Tweet modeling with LSTM recurrent neural networks for hashtag recommendation," In International Joint Conference on Neural Networks (IJCNN), IEEE, (2016), 1570–1577.
16. Ben-Lhachemi, N. and Nfaoui, E.H., "Using Tweets Embeddings For Hashtag Recommendation in Twitter," ***Procedia Computer Science***, Vol. 127, (2018), 7–15.
17. Devi, G.R., Veena, P.V., Kumar, M.A. and Soman, K.P., "Entity Extraction for Malayalam Social Media Text Using Structured Skip-gram Based Embedding Features from Unlabeled Data," ***Procedia Computer Science***, Vol. 93, (2016), 547–553.
18. Araque, O., Corcuera-Platas, I., Sanchez-Rada, J.F. and Iglesias, C.A., "Enhancing deep learning sentiment analysis with ensemble techniques in social applications," ***Expert Systems with Applications***, Vol. 77, (2017), 236–246.
19. Iyyer, M., Manjunatha, V., Boyd-Graber, J. and Daumé III, H., "Deep Unordered Composition Rivals Syntactic Methods for Text Classification," In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), (2015), 1681–1691.
20. Wang, Y., Huang, H., Feng, C., Zhou, Q., Gu, J. and Gao, X., "CSE: Conceptual Sentence Embeddings based on Attention Model," In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), (2016), 505–515.
21. Wieting, J., Bansal, M., Gimpel, K. and Livescu, K., "Towards Universal Paraphrastic Sentence Embeddings," In International Conference on Learning Representations (ICLR 2016), (2015), 1–19.
22. Gong, Y., Zhang, Q., Han, X. and Huang, X., "Phrase-based hashtag recommendation for microblog posts," ***Science China Information Sciences***, Vol. 60, No. 1, (2017), 012109:1–012109:13.
23. Zangerle, E., Gassler, W., and Specht, G., "Recommending #-tags in twitter," In Proceedings of the Workshop on Semantic Adaptive Social Web (SASWeb 2011). CEUR Workshop Proceedings (Vol. 730), (2011), 67–78.
24. Mikolov, T., Chen, K., Corrado, G. and Dean, J., "Efficient Estimation of Word Representations in Vector Space," ***Arxiv Preprint Arxiv:1301.3781***, (2013), 1–12.
25. Weston, J., Chopra, S. and Adams, K, "# tagspace: Semantic embeddings from hashtags," In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), (2014), 1822–1827.
26. Gong, Y. and Zhang, Q., "Hashtag Recommendation Using Attention-Based Convolutional Neural Network.," In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, AAAI Press, (2016), 2782–2788.
27. Chen, K., Chen, T., Zheng, G., Jin, O., Yao, E. and Yu, Y., "Collaborative personalized tweet recommendation," In Proceedings of the 35th International ACM SIGIR Conference on

Research and Development in Information Retrieval - SIGIR '12, ACM Press, (2012), 661–670.
28. Al-Dhelaan, M. and Alhawasi, H., "Graph Summarization for Hashtag Recommendation," In 3rd International Conference on Future Internet of Things and Cloud, IEEE, (2015), 698–702.
29. Ma, Z., Sun, A., Yuan, Q. and Cong, G., "Tagging Your Tweets: A Probabilistic Modeling of Hashtag Annotation in Twitter," In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management - CIKM '14, ACM Pres**s**, (2014), 999–1008.
30. Liu, Z., Liang, C., and Sun, M., "Topical word trigger model for keyphrase extraction," In Proceedings of COLING, (2012), 1715–1730.
31. Li, J. and Xu, H., "Suggest what to tag: Recommending more precise hashtags based on users' dynamic interests and streaming tweet content," ***Knowledge-Based Systems***, Vol. 106, (2016), 196–205.
32. Meng, L., Huang, R., and Gu, J., "A review of semantic similarity measures in wordnet," ***International Journal of Hybrid Information Technology***, Vol. 6, No. 1, (2013), 1–12.
33. Kolb, P., "Disco: A multilingual database of distributionally similar words," In Proceedings of KONVENS, (2008), 1–8.
34. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S. and Bizer, C., "DBpedia-A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia", ***Semantic Web***, Vol. 6, No. 2, (2012), 167-195.
35. Sánchez, D., Batet, M., Isern, D. and Valls, A., "Ontology-based semantic similarity: A new feature-based approach," ***Expert Systems with Applications***, Vol. 39, No. 9, (2012), 7718–7728.
36. McInnes, B.T. and Pedersen, T., "Evaluating measures of semantic similarity and relatedness to disambiguate terms in biomedical text," ***Journal of Biomedical Informatics***, Vol. 46, No. 6, (2013), 1116–1124.
37. TREC. 2011. Tweets2011. Retrieved from http://trec.nist.gov/data/tweets/ (retrieved March 2018, archived by WebCite® at http://www.webcitation.org/6W1ZVkk8o).
38. Twitter4j open-source library (2016, Mar 09). Twitter4j open-source library. [Web-post]. Retrieved Jun 18, 2018, http://yusuke.homeip.net/twitter4j/en/index.html.
39. Baziz, M., Boughanem, M., and Traboulsi, S., "A concept-based approach for indexing documents in IR," In proceedings of INFORSID, (2005), 489–504.
40. Malo, P., Siitari, P., Ahlgren, O., Wallenius, J. and Korhonen, P., "Semantic Content Filtering with Wikipedia and Ontologies," In IEEE International Conference on Data Mining Workshops, IEEE, (2010), 518–526.
41. McCandless, M., Hatcher, E., and Gospodnetic, O., Lucene in action: covers Apache Lucene 3.0, Manning Publications Co., (2010).
42. Finlayson, M., "Java libraries for accessing the princeton wordnet: Comparison and evaluation," In Proceedings of the Seventh Global Wordnet Conference, (2014), 78–85.
43. Panchenko, A., Ruppert, E., Faralli, S., Ponzetto, S.P. and Biemann, C., "Unsupervised does not mean uninterpretable : the case for word sense induction and disambiguation," In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1, Long Papers), (2017), 86–98.
44. Liu, B., Web data mining: exploring hyperlinks, contents, and usage data, Springer Science & Business Media, (2007).
45. Seco, N., Veale, T., and Hayes, J., "An intrinsic information content metric for semantic similarity in WordNet," In Proceedings of the 16th European Conference on Artificial Intelligence, IOS Press, (2004), 1089–1090.
46. Agirre, E., Banea, C., Cer, D., Diab, M., Gonzalez-Agirre, A., Mihalcea, R., Rigau, G. and Wiebe, J., "Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation," In Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016), (2016), 497–511.

Automatic Hashtag Recommendation in Social Networking and Microblogging Platforms Using a Knowledge-Intensive Content-based Approach

M. Jaderyan, H. Khotanlou

Department of Computer Engineering, Bu-Ali Sina University, Hamedan, Iran

P A P E R I N F O

Paper history:
Received 23 April 2019
Received in revised form 23 May 2019
Accepted 05 July 2019

Keywords:
Content Enrichment
Hashtag Recommendation
Knowledge-Intensive
Ontology
Semantic Network Representation
Structured Knowledge Base

چکیده

در پلتفرم‌های شبکه‌های اجتماعی یا محیط‌های میروبلاگ، غالباً از هشتگ برای دسته‌بندی پیام‌ها و انتقال محتوای کلیدی آن‌ها استفاده می‌شود. همچنین برخی شبکه‌های اجتماعی نظیر توئیتر محدودیت بر روی تعداد کاراکترهای پیام‌ها قائل می‌شوند. از هشتگ می‌توان به عنوان ابزاری مفید برای کمک به بیان بهتر محتوای پیام‌های کاربران استفاده کرد. در این مقاله یک سیستم دانش‌محور و مبتنی بر محتوای پیشنهاد دهنده هشتگ معرفی می‌شود. سیستم پیشنهادی از طریق یکپارچه‌سازی دانش ساخت یافته در تمامی واحدهای روش پیشنهادی عمل می‌کند. در مرحله اول ویژگی‌های مرتبط، ساختارهای معنایی و محتوای اطلاعاتی ازپیام‌ها استخراج می‌شوند. از آنجا که محتوای اندکی در پیام‌ها وجود دارد، یک واحد غنی‌سازی محتوا ارائه شده است تا ساختارهای اطلاعاتی را شناسایی نماید که می‌توانند به سیستم در بهبود نمایش پیام‌ها کمک نمایند. پیام‌های استخراج شده توسط ساختار شبکه‌های معنایی نمایش داده می‌شوند. سپس یک واحد ترکیبی و چندلایه محاسبه معنایی شباهت، اشتراکات و تفاوت‌های موجود میان ویژگی‌ها، معنا و محتوای اطلاعات دو پیام را شناسایی می‌کند. در نهایت، هشتگ‌های کاندید براساس هشتگ‌های موجود در پیام‌های مشابه با پیام کاربر پیشنهاد داده می‌شوند. برای ارزیابی سیستم پیشنهادی از داده‌های *Tweets2011* استفاده شده است. نتایج نشان می‌دهد که سیستم پیشنهادی قادر است تا هشتگ‌های مطلوبی را در زمان عملیاتی ناچیز و حتی در شرایطی که محتوای اندکی در پیام‌ها موجود است پیشنهاد دهد. واحد غنی‌سازی توانایی مدل‌سازی دقیق معنا و محتوای اطلاعاتی را دارد. همچنین یکپارچه‌سازی واحد محاسبه معنایی شباهت با واحد غنی‌سازی، دقت و کیفیت هشتگ‌های تولید شده را بهبود می‌بخشد. واحدهای توسعه داده شده در روش پیشنهادی تاثیر مثبت و شگرفی بر عملکرد سیستم دارند. سیستم پیشنهادی را می‌توان به‌عنوان توسعه‌ای موفق از یک سیستم دانش‌محور پیشنهاد دهنده هشتگ در نظر گرفت.

***doi**: 10.5829/ije.2019.32.08b.06*