Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 02, June 2020.

TAG RECOMMENDATION FOR SHORT ARABIC TEXT BY USING LATENT SEMANTIC ANALYSIS OF WIKIPEDIA

Iyad AlAgha[1] and Yousef Abu-Samra[2]

(Received: 8-Dec.-2019, Revised: 22-Jan.-2020 and 16-Feb.-2020, Accepted: 24-Feb.-2020)

ABSTRACT

Text tagging has gained a growing attention as a way of associating metadata that supports information retrieval and classification. To resolve the difficulties of manual tagging, tag recommendation has emerged as a solution to assist users in tagging by presenting a list of relevant tags. However, the majority of existing approaches for tag recommendation have focused on domain-specific tagging and tackled long-form text. Open-domain tagging can be challenging due to the lack of comprehensive knowledge and the intensive computations involved. Furthermore, tagging of short text can be problematic due to the difficulty of extracting statistical features. In terms of the language, most efforts have focused on tagging text written in English. The tagging of Arabic text has been challenged by the difficulty of processing the Arabic language and the lack of knowledge sources in Arabic.

This work proposes an approach for tag recommendation for short Arabic text. It exploits the Arabic Wikipedia as a background knowledge and uses it to generate tags in response to input short text. Latent semantic analysis is exploited to analyze Wikipedia content and find articles relevant to the input text. Then, tags are selected from the titles and categories of these articles and are ranked according to relevance.

The approach was evaluated based on experts' ratings of relevance of 993 tags. Results showed that the approach achieved 84.39% mean average precision and 96.53% mean reciprocal rank. A thorough discussion of results is given to highlight the limitations and the strengths of the approach.

KEYWORDS

Tag recommendation, Arabic, Short text, Latent semantic analysis, Wikipedia, Apache Spark.

1. INTRODUCTION

With the massive daily increase of data on the internet, especially text, automatic tagging services that attach informative and descriptive tags to texts have become a necessity for information aggregation and sharing [1]. Tagging is the practice of creating and managing labels called tags that categorize or describe the content by using simple keywords [2]. Many social media platforms, such as Twitter, Facebook and Flicker, provide their users with functionalities for manual tagging to support content categorization and search. However, manual tagging has many documented limitations, including being laborious, ambiguous and error-prone [3]-[4]. In addition, users are often permitted to use their own conventions and interests when creating tags, a thing that makes tags noisy and sparse. Alternatively, automatic text tagging has been investigated in several studies to generate tags without or with minimal intervention from the user [5]-[6]. Automatic text tagging techniques can be classified into two categories based on the source of generated tags [5]: 1) content-based tagging, which extracts tags from the target content by employing information extraction or text categorization techniques; 2) knowledge-based techniques, which use external knowledge sources, such as ontologies [7], folksonomies [8], Wikipedia [9] or Linked Open Data [10] to recommend tags related to the target content. These knowledge sources can support the tagging process by disambiguating words, inferring relationships and leading to better understandability of the target content [11].

Most of the work related to tag recommendation has been applied to long-form text [5]. When it comes to social media, text often has unique characteristics that pose additional challenges. It is often extremely short, poorly composed and tend to be more informal [12]-[13]. These challenges can obstruct the extraction of textual features of short text by applying conventional statistical techniques that work with long text [14]. In addition, most existing efforts have focused on domain-specific

1. I. AlAgha is with Department of Computer Science, Islamic University of Gaza, Gaza, Palestine. Email: ialagha@iugaza.edu.ps

2. Y. Abu-Samra is with Department of Computer Science, Islamic University of Gaza, Gaza, Palestine. Email: y_samra@hotmail.com

tagging. Open-domain tag recommendation can be challenging due to the lack of comprehensive knowledge sources and the intensive computations involved [10]. From the perspective of the language, the majority of works have focused on tagging text written in English or Latin languages. These works benefited from the advancement in the processing of these languages and the presence of rich English -and Latin-based knowledge resources. However, there has been little effort to support tag recommendation for Arabic texts on social media [7]. This has been challenged by the difficulties associated with the processing of the Arabic language and the lack of comprehensive knowledge sources in Arabic [15].

Driven by the above discussion, this work proposes a tag recommendation approach that generates and recommends tags for short Arabic texts. It aims to support open-domain tagging by using the Arabic version of Wikipedia as background knowledge. The choice of Arabic Wikipedia is motivated by its large coverage of various subject areas, a thing that makes it adequate for open-domain text tagging. Given an Arabic short text as input, the proposed approach will suggest a ranked list of tags with high affinity for input text. These tags are selected from Wikipedia articles that closely match with the input text. To achieve that, a topic model for Wikipedia is first created by using Latent Semantic Analysis (LSA) [16]. Without yet delving into the underlying theory, LSA is a matrix-factorization method commonly used in natural language processing and information retrieval. It seeks to better understand a corpus of documents and the relationships between the words in those documents. LSA is used to distil the Wikipedia as a corpus into a set of relevant concepts, each of which corresponds to a topic that the Wikipedia discusses. It then captures the relationships between documents and concepts and between terms and concepts. This can create a simpler representation of Wikipedia that makes it easy to find the set of articles relevant to terms in the input text. LSA is used in this work for the following reasons: First, it can create a low-dimensional representation of the corpus and thus can effectively handle huge data volumes as with Wikipedia. Second, it produces results that are more robust indicators of meaning as compared to the traditional word co-occurrence models. This is due to its ability to extract features that capture underlying latent semantic structure in the term usage across documents[17].

To handle the heavy computations involved in LSA, a cluster of computers was constructed and operated by using Apache Spark [18] as a parallel processing framework. The proposed approach was evaluated by tagging a set of 100 tweets and then assessing the relevance of generated tags. In total, 993 tags generated by our approach were rated as being "relevant" or "irrelevant" by human experts. Results showed that the approach achieved 84.39% mean average precision and 96.53% mean reciprocal rank. Results were also discussed in detail to highlight the limitations and the strengths of the approach.

2. RELATED WORK

Tag recommendation methods can be classified into four categories based on the underlying technology [5], [19]. The first category is tag co-occurrence methods, which exploit tags previously assigned to a collection of objects to suggest candidate tags to new objects [20]-[23]. They often exploit metrics related to tag frequency to suggest related tags based on tags already associated with other texts. The limitation of these works is that they assume the existence of a tagged corpus.

The second group of methods is content-based. These works do not use external corpora, but exploit the textual features of the target text, such as TF-IDF and association rules, to extract candidate terms and phrases and use them as tags [24]-[27]. The main issue with content-based techniques is that they become ineffective when applied on short texts such as tweets. They also lack novelty, because they generate tags that are already part of the target content [5]. Supervised approaches for tag recommendation also fall in this category. As recommendation can be modelled as a ranking problem, supervised approaches often use training samples consisting of candidate tags to which relevance labels are assigned as ground truth. The aim is to generate a model that maps the tag quality attributes into a relevance score or rank. Several works tried to model the tag recommendation problem as a multi-label text classification task by using different classifiers, such as Naïve Bayes [14], [28] and deep neural networks [29]-[31]. However, supervised approaches are often applicable to restricted domains and are challenged by the difficulty of obtaining labelled data.

Another category of tag recommenders include matrix factorization-based methods under which this work falls. These methods use matrix factorization to model pairwise interactions between users, items and tags, such as the ranking preferences of tags for each pair user-item [32]-[33]. Latent Semantic Analysis (LSA) and Latent Dirichet Allocation (LDA) are often used to process and decompose the co-occurrence matrix [34, 35]. LSA learns latent topics by performing a matrix decomposition (SVD) on the term-document matrix. LDA is a probabilistic topic model, where the goal is to decompose a term by the document probability distribution into two distributions: the term by topic distribution and the topic by document distribution. This work uses LSA rather than LDA, because the low-dimensional representation generated by LSA enables to easily measure similarities and no further processing is needed once it is obtained. The separation between the term, document and concept spaces in the outputs of LSA makes it easy to calculate term-to-term, document-to-document and term-to-document relevance by using cosine measure. In addition, there is a lack of well-established methods to choose the number of topics in LDA and it is unrealistic to test different numbers of topics until the best result is achieved [17].

Another common method for tag recommendation is based on graph analysis. Graph-based methods extract tags by analyzing the neighbourhood of the target text or user [36]-[37]. These methods are commonly used for tag recommendation in social networks [38]-[39], where the nodes of the graph correspond to users and edges connecting users. Collaborative filtering techniques [40]-[41] fall in this category, because they exploit the tagging history of users who are similar to the target user. These methods require the presence of graph datasets that capture the tagging behaviour and links between users.

The fourth category of methods for tag recommendation includes clustering-based methods which recommend tags based on clusters or topics of objects [42]-[43]. Given a collection of documents, these method start by applying a clustering or a classification algorithm to divide documents into groups. Then, tagging a new document is performed by first classifying that document into one or more clusters and then selecting the most relevant tags from those clusters as recommended tags. Despite the potential of clustering in reducing dimensionality of the problem, generic tags that describe the whole cluster are often generated, but are less descriptive of the specific content being tagged. These methods also do not perform well with short texts.

Besides the aforementioned categories, some works have tried to combine methods from multiple categories to improve the performance. For example, P. Lops, M. De Gemmis, G. Semeraro, C. Musto and F. Narducci [44] proposed an approach that combined collaborative filtering based on community tagging behaviour and content-based heuristic techniques. P. Symeonidis [45] combined tag clustering with matrix factorization. M. Lipczak, Y. Hu, Y. Kollet and E. Milios [46] proposed a method that extracts terms from the title and description of the target object (a content-based technique) and then expands the set of candidate tags by exploiting tag co-occurrences. Several efforts have tried to overcome the challenges of short-text processing by exploiting complementary knowledge sources, such as ontologies [7], Wikipedia [9] and Linked Open Data [10] to generate tags.

In the domain of Arabic language, several studies have explored the use of matrix factorization techniques, such as LSA and LDA, to process Arabic texts for different purposes. For example, F. S. Al-Anzi and D. AbuZeina [47] used LSA for classifying Arabic documents. They compared LSA with other classification methods and found that LSA outperforms the TF-IDF-based methods. Some works used LSA for Arabic text summarization [48]-[50] and found that LSA improved the clustering performance and resolves issues related to noisy information. M. Naili, A. H. Chaibi and H. B. Ghézala [51] used LDA to identify topics in Arabic texts and examined the impact of using different LDA parameters and Arabic stemmers. R. Mezher and N. Omar [52] approached the problem of automatic Arabic essay scoring by exploiting both syntactic features of text and LSA and found that augmenting the similarity matrix of LSA with syntactic features could improve the results. Although our work is similar to the aforementioned efforts with regard to the use of LSA on Arabic text, it differs in two aspects: 1) it has a different objective, which is tag recommendation for short Arabic text, whose features cannot be easily captured as compared to the long-form text. 2) Previous works applied LSA on the target content, but we applied it on the Arabic Wikipedia as a complementary knowledge source. 3) We tackled issues related to the processing of the enormous content of

Wikipedia by using Apache Spark as a parallel processing framework and performing dimensionality reduction.

Recently, there has been a growing interest among Arab researchers to exploit the Arabic version of Wikipedia for different purposes in computer science. Some works exploited the semi-structured content of Wikipedia to construct ontologies [53]-[54]. Others used Wikipedia features and structure to build Arabic-named entity corpora [55]-[56] or for entity linking [57]. Wikipedia-based categories have been also exploited to improve the categorization of Arabic text [58]. Some works used the Arabic Wikipedia to expand queries submitted to search engines or question answering systems [59]. The work in this paper adds to previous knowledge by extending the use of Arabic Wikipedia to support open-domain text tagging.

3. OVERVIEW OF THE APPROACH

The overall approach for tag recommendation is depicted in Figure 1 and is summarized as follows: It starts by reading, cleansing and processing the Wikipedia content to create a document-term matrix. Then, LSA is applied by performing Singular Value Decomposition (SVD) on the document-term matrix. This creates a low-rank approximation of the original matrix that models concepts in Wikipedia as well as the pairwise relations between terms, documents and concepts. The outputs of SVD will form the core of the tag recommendation system that will serve user queries as shown in the bottom part of Figure 1.

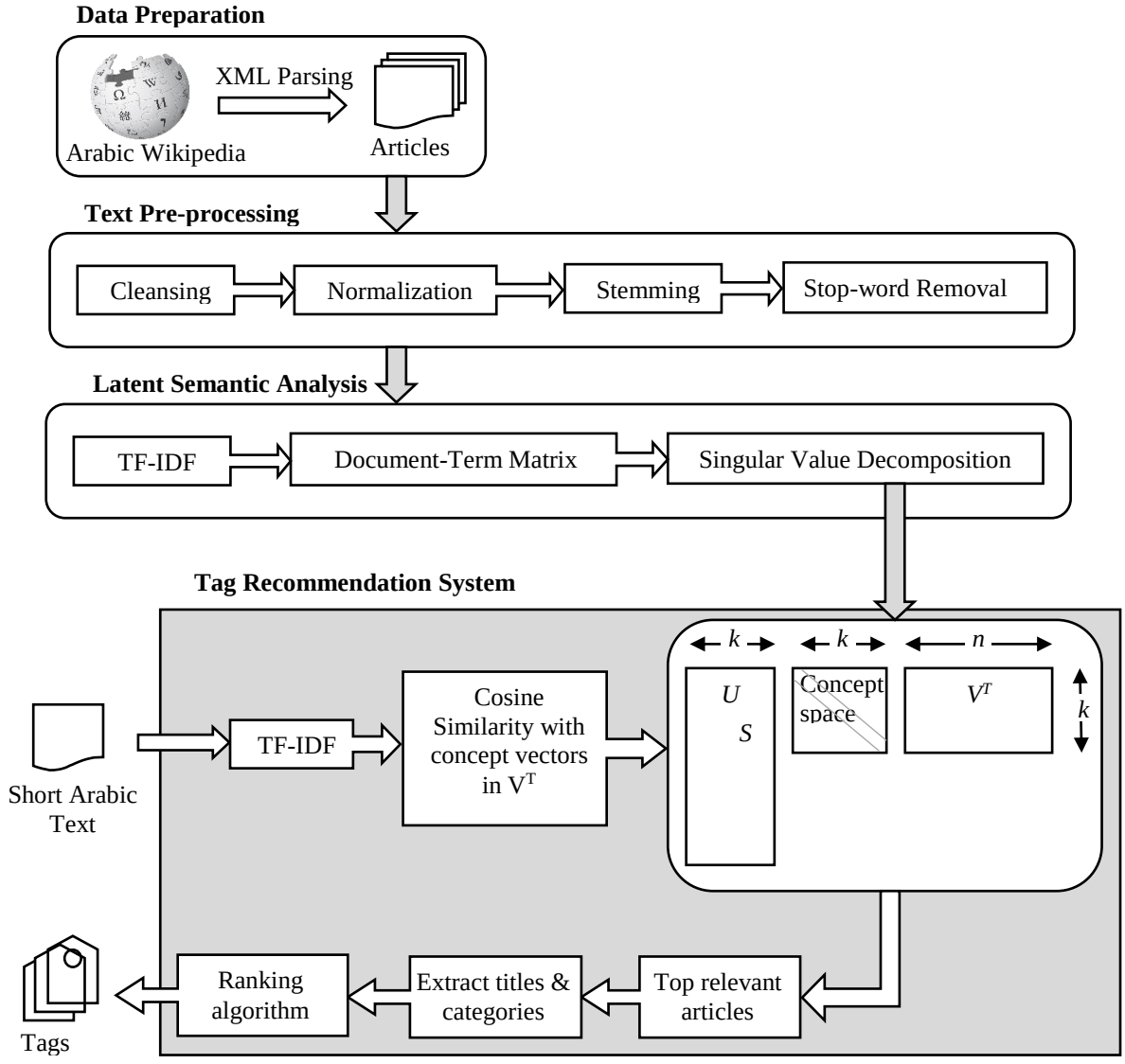

Figure 1. An approach for tag recommendation for short Arabic text based on LSA of Wikipedia.

It takes input text in Arabic, converts it into TF-IDF vector and compares it with concept vectors in SVD results. The aim is to find the concept that is most similar to the input text, which in turn will yield finding Wikipedia articles relevant to the input text. Finally, tags are selected from the titles and categories of these articles and are ranked according to their relevance scores. The approach is explained in detail in the following sections.

4. PRE-PROCESSING OF ARABIC WIKIPEDIA

Wikipedia makes its content available as XML dump files. For this work, we used the dump file of the Arabic version of Wikipedia published in October 2019. It contains 1,238,570 pages, including 435,672 actual articles and 267,580 categories. Pre-processing the Arabic Wikipedia, which is about 6.4 GB of raw text and performing the LSA computations demand a huge memory and processing power. Thus, we used a cluster of computers and Apache Spark as a cluster-computing framework. Apache Spark can distribute computational power to automatically parallelize and execute tasks on a large cluster of computers. It also provides a highly-optimized machine learning library called MLlib [60] which can perform matrix factorization on numerous datasets.

The dump file was first parsed to filter out non-informative pages, such as disambiguation pages, redirect pages, empty pages and templates. This has left only 435675 articles (about 35% of total articles) to be used for LSA. These articles were then processed to extract the textual content, the title and the associated categories. Table 1 presents some details of the pre-processed content.

Table 1. Information on the pre-processed content of Arabic Wikipedia.

Size of XML dump file	6.39 GB
No. of categories	267580
No. of pages	1238570
No. of redirect pages	437726
No. of disambiguation pages	10473
No. of template pages	345759
No. of discussion pages	181
No. of pages with empty body	8756
No. of articles used for LSA	435675

Articles were further processed by performing text pre-processing steps, including cleansing, normalization, stemming and stop-word removal. The cleansing step aims to remove words and phrases that increase the size of the corpus but do not affect the performance, such as the Latin alphabets, special characters, numbers and punctuations. This can both save space and improve fidelity. Normalization is then applied to convert the text into a more convenient and standard form. Normalization of Arabic text may be more complicated as compared to English text, because Arabic words are often connected to pronouns, prefixes and suffixes. In addition, Arabic letters such the 'إ' or 'ي' may be written in different ways. The Stanford Arabic Word Segmenter [61] was used to apply orthographic normalization to raw Arabic text. Afterwards, light stemming [62] was performed to reduce inflected or derived words to their word stem, base or root form. This is crucial, because different formations and derivations of the word may degrade the performance of LSA. We used Farasa [63] for light stemming of Arabic text.

After the pre-processing phase, each Wikipedia article was represented as a title, a list of tokens (cleansed, stemmed and non-stop-words) and a list of associated categories. The next step is related to the articles' details to vectors, which are necessary to perform SVD.

5. SINGULAR VALUE DECOMPOSITION (SVD)

Each article should be represented as a TF-IDF vector. This is done by computing the frequencies of each term within the document and within the entire Wikipedia. Since TF-IDF vectors are likely to have lots of zero values, they are converted into sparse vectors. A sparse-vector representation is more space-efficient, since it only stores the indices of the terms and non-zero values. The collection of

sparse vectors form the document-term matrix, where each row corresponds to a document, each column corresponds to a term and each element indicates the importance of a term to a document.

With the document-term matrix in hand, the analysis can proceed to factorization and dimensionality reduction. MLlib, the machine learning library in Apache Spark, contains an implementation of the SVD that can process enormous matrices. SVD takes the document-term matrix and returns three matrices that approximately equal it when multiplied together, as shown in the following Equation.

$$M_{m\times n} = U_{m\times k}\, S_{k\times k}\, V^{T}{}_{k\times n}$$

where:

- M is the document-term matrix that is input to the SVD implementation.
- *m*, *n*, *k* are the number of documents, number of terms and number of concepts, respectively.
- U is an $m \times k$ matrix, where each row corresponds to a document and each column corresponds to a concept. Each element in U refers to the importance of a document to a concept. Thus, it defines a mapping between the document space and the concept space.
- V^T is a $k \times n$ matrix whose columns are basis of the term space. Each column corresponds to a term and each row corresponds to a concept. Each element in V^T refers to the importance of a term to a concept. Thus, it defines a mapping between the term space and the concept space.
- S is a $k \times k$ diagonal matrix, where each diagonal element in S corresponds to a single concept (and thus a row in V^T and a column in U). A concept captures a thread of variation in the data and often corresponds to a topic that Wikipedia discusses. Each element in S corresponds to the importance of a concept in the corpus.

Note that the three matrices are related so that each diagonal element in S corresponds to a column in U and a row in V^T. The decomposition is parameterized with a number *k*, less than or equal to *n*, which indicates how many concepts to keep around. *k* should be chosen to be less than *n* to create a low-dimensional approximation of the original document-term matrix. A key insight of LSA is that only a small number of concepts is required to ensure that the approximation will be the closest possible to the original matrix. Based on other studies that used LSA with Wikipedia [64, 65], *k* was chosen to be 1000 in our experiment, which is enough to represent the number of different topics discussed in the Arabic Wikipedia.

To illustrate how SVD outputs are interpreted in our approach to find Wikipedia articles that closely match an input text, consider the example shown in Figure 2. It shows the matrices generated after implementing SVD on five sample articles that contain seven unique terms in total. *k*, which denotes the number of concepts, is set to two. It is emphasized that this is a simplified example presented for the purpose of illustration only.

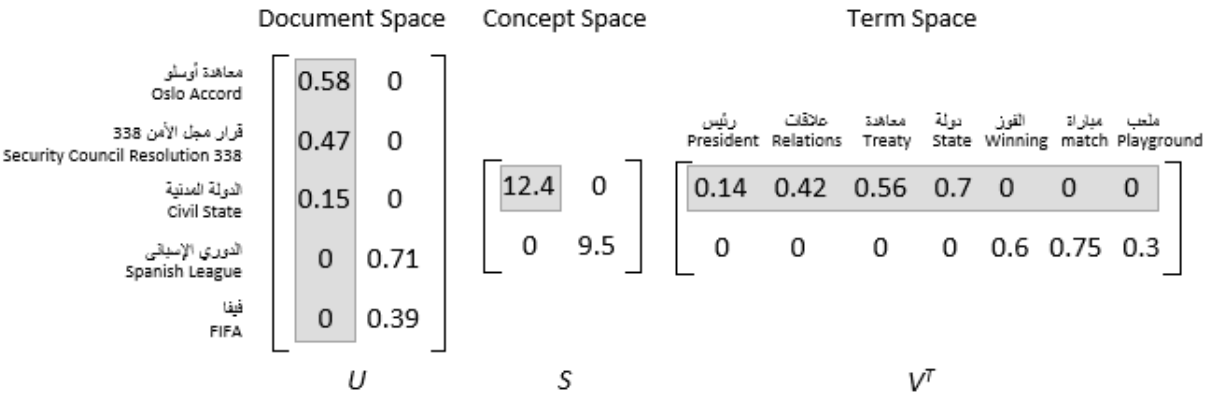

Figure 2. An example of SVD for five documents and seven terms.

Note that each diagonal element in S denotes the weight of a concept; i.e., how important the concept is to the corpus. In the given example, the shaded concept whose weight is 12.4 is the most important, because it holds the largest value. This concept is mapped to the first column in U and to the first row

in V^T. Similarly, the second concept, whose weight is 9.5, is mapped to the second column in U and to the second row in V^T.

Each column in U indicates the degrees of relevance of each article to the corresponding concept in S. For example, the first column of U that is shaded in Figure 2 shows that the article titled: "معاهدة أوسلو (Oslo Accord) ", with the value of 0.58, is the most relevant to the first concept, followed by the article titled: "..قرار مجلس الأمن (Security Council Resolution..) ". Likewise, the second column of U indicates that the article titled: " الدوري الإسباني (Spanish League)", with the value of 0.71, is the most relevant to the second concept. Zero elements in U indicate articles irrelevant to the corresponding concepts in S.

On the other hand, each row in V^T refers to the degrees of relevance of each term to the corresponding concept in S. For example, the first row in V^T that is shaded in Figure 2 reveals that the term " دولة (State)" is the most relevant to the first concept, because it has the highest value, while the term " مباراة (Match)" from the second row in V^T is the most relevant to the second concept.

Knowing this relationship between U, S and V^T matrices, SVD can tell which Wikipedia articles most closely match a set of query terms. Given a set of terms as input, the first step will be to create a TF-IDF vector of input query and find its representation as a new row of the low-rank document-term matrix approximation. Then, similar articles can be discovered by computing the cosine similarity between the new input vector and the other entries in this matrix.

6. MAPPING INPUT TEXT TO RELEVANT WIKIPEDIA ARTICLES

The implementation of SVD on the content of Arabic Wikipedia, as explained above, is performed only once and the SVD outputs are maintained in memory to form the core of the tag recommendation system shown in Figure 1. The system is now ready to take a short Arabic text as input and generate a ranked list of relevant tags as output. The input text will undergo the same text preprocessing steps applied on the Wikipedia content, including cleansing, normalization, stemming and stop-word removal. It is then converted into a sparse vector with TF-IDF weights of terms.

Let d be the TF-IDF vector of input query. We would like to map d into its representation in the SVD space, $\bar{d}$, by applying the following transformation [6]:

$$\bar{d} = S^{-1} U^T d$$

The next step is to find documents in U that are most similar to $\bar{d}$. This can be achieved by computing the cosine similarity between $\bar{d}$ and every row in U. The cosine similarity is employed, because it is simple, very efficient to evaluate especially for sparse vectors and gives normalized values in the range from 0 to 1. Documents that achieve highest cosine similarity scores refer to Wikipedia articles that are most relevant to input text. The next step will be to use these articles to generate recommended tags.

7. TAG GENERATION AND RANKING

Up to this point, input text should have been matched with relevant Wikipedia articles by using SVD outputs. These articles are ranked from the most to the least relevant based on the similarity to input text. Then, recommended tags are extracted from the titles and categories of these articles. While titles tend to be more specific and unique, categories are often more generic and referral to broader subject areas. Thus, tags selected from both titles and categories make a list of diversified and complementary descriptors covering both specific and broad subjects pertaining to the input text. However, number of titles and categories obtained from articles could be large. Thus, they should be filtered and ranked to get only the most relevant ones. The algorithm we use to filter and rank titles and categories can be explained as follows:

Let D = (d_1,d_2,..d_i,.., d_n) be an ordered set of documents obtained from SVD, where i indicates the rank of d; i.e., its relevance to the input text. Let A={a_1, a_2,..,a_m} be the set of terms in input text and B={b_1, b_2,..,b_k} the set of terms in the title of d. Then, each title of d_i is scored using the following Equation:

$$Score\ of\ t_i = \frac{1 + |A \cap B|}{1 + \log i}$$

where t_i is the title of d_i. $|A \cap B|$ is the number of terms occurring in both A and B. Based on this equation, each title is weighted based on criteria that consider both the overlap between the title and the input text and the rank of the document. That is, a title is scored higher if it shares more terms with input text and belongs to a highly ranked article based on SVD. Note that the above measure assures that each title has a non-zero score even if it has no overlap with the input text. This is necessary, because a title that does not overlap with the input text can still be relevant. Scores of titles are then normalized by being rescaled to have values between 0 and 1.

After scoring titles, we now move on to score categories. Since documents obtained from SVD can collectively have a large number of categories, it is necessary to choose only most relevant ones. In addition, we cannot rely on the overlapping of texts to filter categories as we did with titles. Categories are less likely to be included in the target text, because they often describe broader subjects or general classifications. Instead, categories are filtered based on their frequency in articles so that categories that occur more often are prioritized. The score of a category c is computed using the following Equation:

$$Score\ of\ c = \frac{frequency\ of\ c\ in\ D}{|D|}$$

where D is the set of documents obtained from SVD. Based on this equation, the score of a category ranges from 0 to 1, where the category gets the score of 1 if it appears in all documents in D.

Finally, titles and categories are grouped and ordered based on their normalized scores. In our experiment, the number of recommended tags for each input text was limited to the top ten tags. Table 2 shows the tagging results for a sample input tweet including the top scored titles and categories. Only shaded tags, which got the highest scores, are recommended to the end user.

Table 2. A sample input tweet with the tagging results as generated by our approach.

الفرق بين المبرمج ومصمم الجرافيك (The difference between the programmer and the graphic designer)	
Top categories	Top titles
علم الحاسوب (Computer Science)	تصميم الجرافيك (Graphic Design)
تصميم الجرافيك (Graphic Design)	مبرمج (Programmer)
مهن الحاسوب (Computer Professions)	فريق العمل لإنتاج برمجيات الوسائط المتعددة (Multimedia Production Team)
مبرمجون (Programmers)	علم الحاسوب (Computer Science)
تصميم الاتصال (Communication Design)	مصمم جرافيك (Graphic Designer)
هندسة الحاسوب (Computer Engineering)	رسوميات (Graphics)
مهن وسائل الإعلام (Media Careers)	تصميم المعلومات (Information Design)

8. EVALUATION

The objective of the evaluation is to assess the extent to which our tag recommendation approach can suggest tags relevant to the input Arabic text. Existing approaches for tag recommendation have been evaluated either by exploiting tags previously assigned by the users as a ground truth [66]-[67] or manually by relying on external users to evaluate the recommendations [57], [68]-[69]. In this work, we used the second approach, because we are not aware of any dataset of pre-tagged Arabic short texts that we can compare our results to. In addition, we emphasize that comparing LSA with other text-similarity measures is out of the scope of this work. The differences between semantic similarity measures have been experimentally explored in several studies [70]-[71]. Instead, we focus on the problem of short Arabic text tagging and use LSA as an unsupervised approach to achieve this purpose due to its output that facilitates similarity calculations.

Thus, we created a dataset consisting of 100 tweets selected randomly from three different domains: sports, technology and news. The tweets were classified as follows: sports: 36 tweets, technology: 41 tweets and news: 23 tweets. These tweets were used as input to the recommendation approach. The output for each tweet was a set of tags ordered by the system based on the relevance to the input tweet. Only top ten tags per tweet were considered in the evaluation. Thus, 1000 recommendations were

collected in total for the 100 input tweets. These recommendations were then rated by human experts. As tweets were categorized into three distinct domains, each group of tweets was rated by two experts in each domain. Experts rated each tweet as either "relevant" or "irrelevant". Only tags that both experts agreed upon were considered in the evaluation. Finally, 993 tags rated by experts were considered for the evaluation process. Table 3 shows sample tweets from our dataset. The complete dataset including the tweets, the generated tags and the ratings of experts can be downloaded from: https://github.com/YousefSamra/ShortTextTagging and instructions can be found on: http://tiny.cc/op50iz.

Table 3. Sample tweets from the dataset.

Subject	Tweet
Sports	موقعة قويّة بين تشيلسي ومان سيتي وليفربول يترصّد A strong match between Chelsea and Man City and Liverpool stalks
Technology	سيانوجين مود رائدة تطوير رومات الأندرويد CyanogenMod is a pioneer in the development of Android ROM
News	فلسطين المحتلة: الصحفي محمد القيق يواصل إضرابه عن الطعام بسجون الصهاينة Occupied Palestine: Journalist Muhammad Al-Qeeq continues his hunger strike in Israeli prisons

8.1 Experimental Settings

The experiment was carried out in a computer lab consisting of 20 laptops, all with the specifications shown in Table 4. The laptops were all connected to a single LAN and controlled and scheduled by Apache Spark framework installed on a master computer. The cluster was used to operate the code for pre-processing the Wikipedia content and performing LSA. After getting the outputs of SVD, the system became ready to take a short text as input and to produce tags rapidly as outputs.

Table 4. Specifications of laptops used in the experiment.

Machine	HP laptop
CPU	Core i7 2.6 GHz
RAM	6 GB
OS	Windows 10, 64bit

8.2 Evaluation Metrics

In tag recommendation, the most important result for the end user is to receive a list of recommendations, ordered from the most to the least relevant. So, we used three metrics that are commonly used to evaluate recommendation systems [72]-[73]. These metrics are:

Precision at position k (P@k), where k denotes the number of recommended tags for each tweet. We aim to explore how the precision is affected when changing the number of tags to be examined. P@k is computed using the following equation:

$$P@k = \frac{number\ of\ relevant\ tags\ in\ top\ k\ positions}{k}$$

Mean Average Precision (MAP): The average precision for the query q is computed using the following equation:

$$AP(q) = \frac{\sum_{k=1}^{m} P@k(q)}{number\ of\ relevant\ tags\ for\ q}$$

where, m is the total number of results for the query q and P@k is the precision at position k. The mean average precision for a set of queries is the mean of the average precision scores for each query:

$$MAP = \frac{1}{N}\sum_{q=1}^{N} AP(q)$$

Mean Reciprocal Rank (MRR): While the first two metrics emphasize the quality of the top k tags, the MRR focuses on a practical goal, which is how deep the user has to go down a ranked list to find one useful tag [74]. MRR is the average of the reciprocal ranks of results for a sample of queries N and is calculated using the following Equation:

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i}$$

where $rank_i$ refers to the rank position of the first relevant tag for the i-th query.

Average processing time: To assess the efficiency of the system, the average time required to tag each tweet was recorded .This is the time elapsed from inputting each tweet until the tags are generated.

The above evaluation metrics were then calculated according to the rates obtained from the experts. Table 5 shows how the metrics were calculated for a sample tweet. It shows the ordered list of recommended tags (10 tags), along with the experts' ratings of each tag and the calculated values of metrics.

Table 5. Expert ratings and evaluation metrics for a sample tweet.

RR	AP@k	P@k	Experts' rating (relevant =1, irrelevant = 0)	واتساب 2017: توقف خدمة واتس اب عن العمل على بعض الهواتف. اكتشف ان كان هاتفك من القائمة (WhatsApp 2017: WhatsApp has stopped working on some phones. Find out if your phone is on the list)	
1	0.82602	1	1	واتسآب (WhatsApp)	1
		0.5	0	سناب شات (Snapchat)	2
		0.666667	1	تراسل فوري (Instant Messaging)	3
		0.75	1	برمجيات آي أو إس (iOS Software)	4
		0.8	1	برمجيات أندرويد (Android Software)	5
		0.833333	1	برمجيات متعددة المنصات (Multi-platform Software)	6
		0.857143	1	برمجيات اتصال (Communication Software)	7
		0.875	1	مراسلة فورية (Instant Messaging)	8
		0.777778	0	برمجيات بلاك بيري (Blackberry Software)	9
		0.7	0	برمجيات سيمبيان (Symbian Software)	10

8.3 Results and Discussion

Table 6 shows the evaluation results. A total number of 933 tags were gathered and assessed by experts. 658 out of 933 were assessed as relevant, giving an AP@10 of 71.94%. Our approach also achieved a MAP of 84.39% and a MRR of 96.53%, indicating that the tagging approach achieved high precision.

Table 6. Evaluation results.

Number of generated tags @ k=10	933
Number of correct tags	658
AP@10	71.94%
MAP	84.39%
MRR	96.53%
Avg. processing time	2.54 sec.

In addition, the average processing time was 2.54 seconds. This result indicates that the approach is suitable for real-time usage, especially when considering the huge size of Wikipedia content and the

intensive computations involved. When using sufficient computing and storage resources, LSA becomes an efficient text-mining technique, because it creates and uses a low-dimensional representation of the original document-term matrix [16].

The tagging performance was also explored across different subject domains. As shown in Table 7, the values of MAP and MRR for the three subject domains were close, indicating that the approach performed well in the three domains. This result indicates that the Arabic Wikipedia can be a reasonable choice as a background knowledge for open-domain tagging due to its generality and coverage of a wide range of topics.

Table 7. Results across different subjects.

Subject	No. of tweets	MAP	MRR
Sports	36	80.81%	95.46%
Technology	41	85.85%	96.83%
News	23	87.12%	97.83%

Figure 3 depicts how the average precision (AP@k) changes as k changes from 1 to 10. The precision is highest when k=1 and then declines consistently as k increases. This indicates that the approach often orders tags according to their relevance so that most relevant tags come on the top of the list.

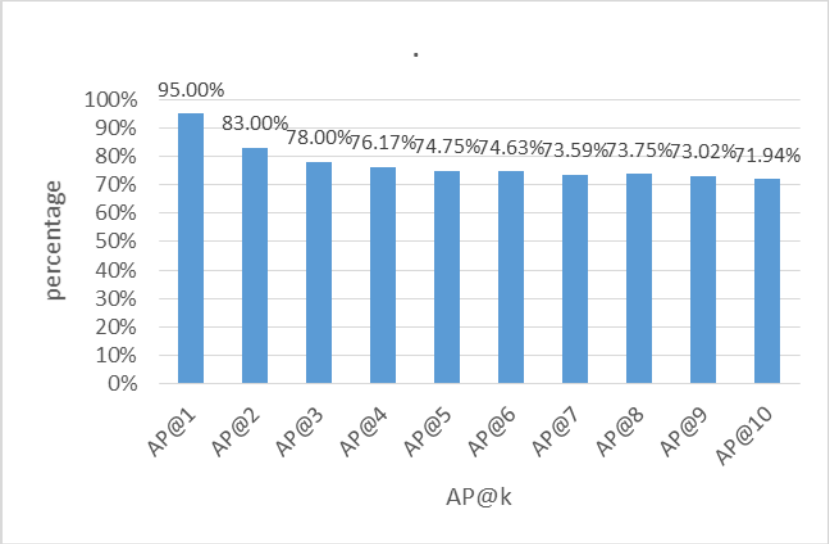

Figure.3 AP(1-100)@k(1-10).

To further explain our results, generated tags were inspected thoroughly to identify the main sources of strengths and weaknesses. The following discusses the main observations:

Term ambiguity: One challenge of any automatic tagging service is the ability to resolve word ambiguity and pick tags that conform to the context of the input text. The approach showed good performance with respect to handling polysemy; i.e., recognizing terms that have different meanings in different contexts, as was evident from several examples. Consider the following tweet: " زيدان: أخشى أن يسقط ريال مدريد مجددا - Zidane: I am afraid Real Madrid will fall again": The name " زيدان (Zidane)" could refer to many public figures, such as a French former player, a philosopher and an actor. However, the approach suggested the tag "زين الدين زيدان (Zinedine Zidane)" which refers to the intended person. In another example, the tweet " مواعيد العمل في معبر الكرامة بعد غد.. - Working hours at Al-Karamah crossing after tomorrow.." was associated with the tag "جسر الملك حسين (King Hussein Bridge)" which is the alternative name of Al-Karamah crossing between Palestine and Jordan. Tags in the previous examples were consistent with the context and the intended meanings of input terms. This can be attributed to the LSA's ability to base similarity scores on a deeper understanding of the corpus. LSA can capture Wikipedia concepts and associate articles with relevant concepts. When the input

text is compared with articles, it will be able to recover the relationship between terms, such as " ريال مدريد (Real Madrid)" and " زيدان (Zidane)" based on the co-occurrence of both terms in articles associated with the same concepts. However, there were a few cases where the approach failed to resolve ambiguity and thus generated false tags. For example, the tweet: " ملعب كرة قدم في قطر على شكل بيت شعر..- A football stadium in Qatar in the form of Bedouin tent.." was tagged with: "قافية (rhyme)", "إيقاع شعري (Poetic rhythm)", "شعر (Poetry)" and "ملاعب كرة قدم في قطر (Football stadiums in Qatar)". While the first three tags are not related to the context, the fourth is relevant, but is ranked lower. This result can be explained by the lack of articles discussing both sports and football stadiums in Qatar. The ability of LSA to handle polysemy is often proportional to the number and depth of articles covering the ambiguous terms. In addition, our implementation of LSA does not consider the diacritization of Arabic[75], which is the process of restoring the diacritical marks, for handling morphological and syntactic ambiguity. Thus, the approach was not able to distinguish the difference between the words "شِعْر (Poetry)" and "شَعْر (Hair)".

Synonymy: One of the advantages of LSA is its potential to recognize synonyms and alternative words by condensing related terms [76]. This advantage was evident in many results, where the approach could recommend synonyms of terms from the input text. For example, the tweet " #بريطانيا عاجل | إغلاق محطة #لندن بريدج للقطارات والمنطقة المحيطة بسبب تحذير أمني - Britain urgent: London Bridge station and the surrounding area have been closed due to a security warning ", was tagged with the terms: "المملكة المتحدة (UK)","إنجلترا (England)". Similarly, the tweet "..المعالج أكثر المنتجات تعقيداً اليوم - The processor is the most sophisticated product today.." was tagged with the following terms " وحدة المعالجة المركزية (CPU)" and " تصميم وحدة المعالجة المركزية (CPU Design)", which are all synonyms of the term " المعالج (The Processor)". A common limitation of content-based recommendation techniques is the lack of novelty, because they extract tags from the own content of the target text. This limitation significantly diminished in our approach, because tags were extracted from Wikipedia articles rather than from the target text.

Tag selection procedure: As explained earlier, the proposed approach uses a tag selection procedure that considers both titles and categories of articles. This combination of titles and categories often resulted in tags that varied in generality and covered both narrow and broad topics. For example, the tweet "غسان كنفاني: روائي وقاص وصحفي فلسطيني تم اغتياله على يد الموساد الصهيوني - Ghassan Kanafani: a Palestinian novelist, storyteller and journalist assassinated by the Mossad " had the following tags in order: "غسان كنفاني (Ghassan Kanafani)", "أدباء وكتاب فلسطين- Authors and writers of Palestine" and "الصراع العربي الإسرائيلي- Arab-Israeli Conflict". The first tag is a title of an article, while the rest are categories. In another example: " شركة أوراكل المتخصصة بحلول قواعد البيانات وتكنولوجيا المعلومات بلغت قيمتها 168 مليار دولار - Oracle, which specializes in database solutions and IT, valued at $ 168 billion " was tagged with the following terms in order: "أوراكل (Oracle)","نظام إدارة قواعد البيانات العلائقية (Relational Database System)","تقنية" (Technique) and "بيانات ضخمة (Big Data)". The first two tags in the former example refer to titles, while the last two are categories. In both examples, titles are generally more concise and descriptive than categories, while categories are more generic and can give a broad insight into the subject area pertaining to the tweets. This can fulfil the interests of users that may vary with respect to the desired specificity of results.

9. CONCLUSION AND FUTURE WORK

This work presents an approach for tag recommendation of short Arabic text. It uses LSA to uncover the latent concepts within Wikipedia and to provide scores of similarity between documents, concepts and terms. The LSA model was used to find Wikipedia articles that best match with the target text. Tags are selected from titles and categories of retrieved articles to provide recommendations covering both specific and broad topics. In addition, selected tags are ranked based on several factors that include the overlap between the title and the input text, the rank of corresponding articles and the frequency of category in articles.

The evaluation of the approach by assessing resultant recommendations against expert judgments has proved the effectiveness and efficiency of the approach. In addition, the inspection of results has provided an insight into the strengths and weaknesses of the approach. The contribution of this work is

two-folded: First, it tackles the problem of open-domain and real-time tag recommendation for short Arabic text, which is a problem that remains briefly addressed in the literature. Second, it exploits Wikipedia as a comprehensive source of tags and analyzes it by using LSA to match the input query with relevant articles. To our knowledge, little effort has been devoted to leveraging the Arabic version of Wikipedia with LSA for tag recommendation.

There are many directions to extend this work: First, we aim to improve the tagging results by testing techniques other than LSA, such as LDA and supervised approaches. Second, we may explore the unique challenges associated with the Arabic language, such as the diacritization of text and its impact on results. Third, we may explore the use of LSA with Wikipedia for other applications, such as question answering and text summarization. Third, we aim to deploy the proposed tagging service and integrate it with social media platforms in order to evaluate it at a larger scale.

REFERENCES

[1] V. Oliveira, G. Gomes, F. Belém, W. Brandao, J. Almeida, N. Ziviani and M. Gonçalves, "Automatic Query Expansion Based on Tag Recommendation," Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp. 1985-1989, 2012.

[2] O. Nov, M. Naaman and C. Ye, "What Drives Content Tagging: The Case of Photos on Flickr," ACM, pp. 1097-1100, 2008.

[3] M. R. Bouadjenek, H. Hacid and M. Bouzeghoub, "Social Networks and Information Retrieval, How Are They Converging? A Survey, a Taxonomy and an Analysis of Social Information Retrieval Approaches and Platforms," Information Systems, vol. 56, pp. 1-18, 2016.

[4] G. Sriharee: "An Ontology-based Approach to Auto-tagging Articles," Vietnam Journal of Computer Science, vol. 2, no. 2, pp. 85-94, 2015.

[5] F. M. Belém, J. M. Almeida and M. A. Gonçalves, "A Survey on Tag Recommendation Methods," Journal of the Association for Information Science and Technology, vol. 68, no. 4, pp. 830-844, 2017.

[6] O. Vechtomova, "Introduction to Information Retrieval," Proc. of the 40th European Conference on IR Research, 2009.

[7] I. Al-Agha and O. Abu-Dahrooj: "Multi-level Analysis of Political Sentiment Using Twitter Data: A Case Study of the Palestinian-Israeli Conflict," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 5, no. 3, 2019.

[8] "Latent Semantic Indexing," [Online], Availab.:https://en.wikipedia.org/wiki/Latent_semantic_analysis.

[9] H.-K. Hong, G.-W. Kim and D.-H. Lee: "Semantic Tag Recommendation Based on Associated Words Exploiting the Interwiki Links of Wikipedia," Journal of Information Science, vol. 44, no. 3, pp. 298-313, 2018.

[10] L. Jayaratne, "Content Based Cross-domain Recommendation Using Linked Open Data," GSTF Journal on Computing, vol. 5, no. 3, 2017.

[11] S. Vairavasundaram, V. Varadharajan, I. Vairavasundaram and L. Ravi, "Data Mining-based Tag Recommendation System: An Overview," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 5, no. 3, pp. 87-112, 2015.

[12] W. Guo, H. Li, H. Ji and M. T. Diab, "Linking Tweets to News: A Framework to Enrich Short Text Data in Social Media", ACL, vol. 1, pp. 239-249, 2013.

[13] S. Garcia Esparza, M. P. O'Mahony and B. Smyth, "Towards Tagging and Categorization for Micro-blogs," Paper presented at the 21st National Conference on Artificial Intelligence and Cognitive Science (AICS 2010), Galway, Ireland, 30 August-1 September, 2010.

[14] R. Dovgopol and M. Nohelty, "Twitter Hash Tag Recommendation," arXiv preprint arXiv:1502.00094, 2015.

[15] I. M. AlAgha and A. Abu-Taha, "AR2SPARQL: An Arabic Natural Language Interface for the Semantic Web," AR2SPARQL: An Arabic Natural Language Interface for the Semantic Web, vol. 124, no. 18, 2015.

[16] T. K. Landauer, D. S. McNamara, S. Dennis and W. Kintsch, Handbook of Latent Semantic Analysis, Psychology Press, 2013.

[17] T. Cvitanic, B. Lee, H. I. Song, K. Fu and D. Rosen, "LDA *v.* LSA: A Comparison of Two Computational Text Analysis Tools for the Functional Categorization of Patents", ICCBR Workshops, 2016.

[18] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman and M. J. Franklin, "Apache Spark: A Unified Engine for Big Data Processing," Communications of the ACM, vol. 59, no. 11, pp. 56-65, 2016.

[19] R. Singh and A. Rani, "A Survey on the Generation of Recommender Systems," International Journal of Information Engineering and Electronic Business, vol. 9, no. 3, pp. 26-35, 2017.

[20] C. Wartena, R. Brussee and M. Wibbels, "Using Tag Co-occurrence for Recommendation," Proc. of the 9th International Conference on Intelligent Systems Design and Applications (ISDA), Pisa, Italy, pp. 273-278, 2009.

[21] R. Damaševicius, R. Valys and M. Woźniak, "Intelligent Tagging of Online Texts Using Fuzzy Logic," Proc. of IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1-8, 2016.

[22] G. V. Menezes, J. M. Almeida, F. Belém, M. A. Gonçalves, A. Lacerda, E. S. De Moura, G. L. Pappa, A. Veloso and N. Ziviani, "Demand-driven Tag Recommendation," Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part II, pp. 402-417, 2010.

[23] K. Yanai, "VisualTextualRank: An Extension of Visualrank to Large-scale Video Shot Extraction Exploiting Tag Co-occurrence," IEICE Transactions on Information and Systems, vol. 98, no. 1, pp. 166-172, 2015.

[24] M. P. Lipczak and E. Milios, "Efficient Tag Recommendation for Real-life Data," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 3, no. 1, pp. 1-21, 2011.

[25] Y. Wu, Y. Yao, F. Xu, H. Tong and J. Lu, "Tag2word: Using Tags to Generate Words for Content Based Tag Recommendation," Proc. of the 25th International ACM Conference (CIKM '16), pp. 2287-2292, 2016.

[26] J. Wang, L. Hong and B. D. Davison, "Tag Recommendation Using Keywords and Association Rules," RSDC'09, pp. 1-14, 2009.

[27] F. M. Belém, E. F. Martins, J. M. Almeida and M. A. Gonçalves, "Personalized and Object-centered Tag Recommendation Methods for Web 2.0 Applications," Information Processing & Management, vol. 50, no. 4, pp. 524-553, 2014.

[28] I. Katakis, G. Tsoumakas and I. Vlahavas, "Multi-label Text Classification for Automated Tag Suggestion", ECML/PKDD, pp. 1-9, 2008.

[29] Y. Gong and Q. Zhang: "Hashtag Recommendation Using Attention-based Convolutional Neural Network," Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI-16), pp. 2782-2788, 2016.

[30] Y. Wang, J. Li, I. King, M. R. Lyu and S. Shi, "Microblog Hashtag Generation *via* Encoding Conversation Contexts," arXiv preprint arXiv:1905.07584, 2019.

[31] H. T. Nguyen, M. Wistuba, J. Grabocka, L. R. Drumond and L. Schmidt-Thieme, "Personalized Deep Learning for Tag Recommendation", Proc. of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, pp. 186-197, 2017.

[32] Y. Yang, L. Han, Z. Gou, B. Duan, J. Zhu and H. Yan, "Tagrec-CMTF: Coupled Matrix and Tensor Factorization for Tag Recommendation," IEEE Access, vol. 6, pp. 64142-64152, 2018.

[33] C. Lu, B. Shen, L. Zhang and J. Allebach, "Tag Recommendation *via* Robust Probabilistic Discriminative Matrix Factorization," Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1170-1174, 2016.

[34] J. Yao, Y. Wang, Y. Zhang, J. Sun and J. Zhou, "Joint Latent Dirichlet Allocation for Social Tags," IEEE Transactions on Multi-media, vol. 20, no. 1, pp. 224-237, 2017.

[35] M. A. Masood, R. A. Abbasi, O. Maqbool, M. Mushtaq, N. R. Aljohani, A. Daud, M. A. Aslam and J. S. Alowibdi, "MFS-LDA: A Multi-feature Space Tag Recommendation Model for Cold Start Problem," Program, vol. 51, no. 3, pp. 218-234, 2017.

[36] T.-A. N. Pham, X. Li, G. Cong and Z. Zhang, "A General Graph-based Model for Recommendation in Event-based Social Networks," International Conference on Data Engineering, pp. 567-578, 2015.

[37] M. Hmimida and R. Kanawati, "A Graph-coarsening Approach for Tag Recommendation," Proc. of the International World Wide Web Conferences Steering Committee, pp. 43-44, 2016.

[38] Y. Chen, H. Dong and W. Wang, "Topic-graph Based Recommendation on Social Tagging Systems: A Study on Research Gate," ACM, pp. 138-143, 2018.

[39] M. Rawashdeh, M. F. Alhamid, J. M. Alja'am, A. Alnusair and A. El Saddik, "Tag-based Personalized Recommendation in Social Media Services," Multimedia Tools and Applications, vol. 75, no. 21, pp. 13299-13315, 2016.

[40] M. A. Chatti, S. Dakova, H. Thüs and U. Schroeder, "Tag-based Collaborative Filtering Recommendation in Personal Learning Environments," IEEE Transactions on Learning Technologies, vol. 6, no. 4, pp. 337-349, 2013.

[41] S. Panigrahi, R. K. Lenka and A. Stitipragyan, "A Hybrid Distributed Collaborative Filtering Recommender Engine Using Apache Spark," Procedia Comp. Science, vol. 83, pp. 1000-1006, 2016.

[42] Y. Song, L. Zhang and C. L. Giles, "Automatic Tag Recommendation Algorithms for Social Recommender Systems," ACM Transactions on the Web (TWEB), vol. 5, no. 1, p. 4, 2011.

[43] R. Krestel and P. Fankhauser, "Personalized Topic-based Tag Recommendation," Neurocomputing, vol. 76, no. 1, pp. 61-70, 2012.

[44] P. Lops, M. De Gemmis, G. Semeraro, C. Musto and F. Narducci, "Content-based and Collaborative Techniques for Tag Recommendation: An Empirical Evaluation," Journal of Intelligent Information Systems, vol. 40, no. 1, pp. 41-61, 2013.

[45] P. Symeonidis, "ClustHOSVD: Item Recommendation by Combining Semantically Enhanced Tag Clustering with Tensor HOSVD", IEEE Transactions on Systems, Man and Cybernetics: Systems, vol. 46, no. 9, pp. 1240-1251, 2015.

[46] M. Lipczak, Y. Hu, Y. Kollet and E. Milios, "Tag Sources for Recommendation in Collaborative Tagging Systems," ECML PKDD Discovery Challenge, vol. 497, pp. 157-172, 2009.

[47] F. S. Al-Anzi and D. AbuZeina, "Toward an Enhanced Arabic Text Classification Using Cosine Similarity and Latent Semantic Indexing," Journal of King Saud University-Computer and Information Sciences, vol. 29, no. 2, pp. 189-195, 2017.

[48] H. Froud, A. Lachkar and S. A. Ouatik, "Arabic Text Summarization Based on Latent Semantic Analysis to Enhance Arabic Documents Clustering," arXiv preprint arXiv:1302.1612, 2013.

[49] K. Al-Sabahi, Z. Zhang, J. Long and K. Alwesabi, "An Enhanced Latent Semantic Analysis Approach for Arabic Document Summarization," Arabian Journal for Science and Engineering, vol. 43, no. 12, pp. 8079-8094, 2018.

[50] H. Alazzam and A. Alsmady, "A Distributed Arabic Text Classification Approach Using Latent Semantic Analysis for Big Data," Proc. of the 12th IEEE International Scientific and Technical Conference on Computer Sciences and Information Technologies (CSIT), pp. 58-61, 2017.

[51] M. Naili, A. H. Chaibi and H. B. Ghézala, "Empirical Study of LDA for Arabic Topic Identification," HAL Id: hal-01444574, 2016.

[52] R. Mezher and N. Omar, "A Hybrid Method of Syntactic Feature and Latent Semantic Analysis for Automatic Arabic Essay Scoring," Journal of Applied Sciences, vol. 16, no. 5, p. 209, 2016.

[53] N. I. Al-Rajebah and H. S. Al-Khalifa, "Extracting Ontologies from Arabic Wikipedia: A Linguistic Approach," Arabian Journal for Science and Engineering, vol. 39, no. 4, pp. 2749-2771, 2014.

[54] M. M. Boudabous, L. H. Belguith and F. Sadat, "Exploiting the Arabic Wikipedia for Semi-automatic Construction of a Lexical Ontology," International Journal of Metadata, Semantics and Ontologies, vol. 8, no. 3, pp. 245-253, 2013.

[55] F. Alotaibi and M. Lee, "Mapping Arabic Wikipedia into the Named Entities Taxonomy", Proceedings of COLING 2012, pp. 43-52, 2012.

[56] M. Al-Smadi, B. Talafha, O. Qawasmeh, M. N. Alandoli, W. A. Hussien and C. Guetl, "A Hybrid Approach for Arabic Named Entity Disambiguation," Proc. of the 15th International Conference on Knowledge Technologies and Data-drive, ACM, 2015.

[57] F. Fayad and I. AlAgha, Automatic Linking of Short Arabic Texts to Wikipedia, M.Sc. Thesis, Faculty of Information Technology, The Islamic University-Gaza, Palestine, 2013.

[58] A. Yahya and A. Salhi, "Arabic Text Categorization Based on Arabic Wikipedia," ACM Transactions on Asian Language Information Processing (TALIP), vol. 13, no. 1, p. 4, 2014.

[59] A. Mahgoub, M. Rashwan, H. Raafat, M. Zahran and M. Fayek, "Semantic Query Expansion for Arabic Information Retrieval," Arabic Natural Language Processing Workshop (EMNLP), Doha, Qatar, pp. 87-92, 2014.

[60] X. Meng, J. Bradley, B. Yuvaz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. Tsai, M. Amde and S. Owen, "MLlib: Machine Learning in Apache Spark," JMLR, vol. 17, no. 34, pp. 1-7, 2016.

[61] W. Monroe, S. Green and C. D. Manning, "Word Segmentation of Informal Arabic with Domain Adaptation," Proceedings of the 52nd Annual Meeting of the Association for Computational Sciences, vol. 2, pp. 206-211, 2014.

[62] L. S. Larkey, L. Ballesteros and M. E. Connell, "Light Stemming for Arabic Information Retrieval," Arabic Computational Morphology, (Springer, Dordrecht,), pp. 221-243, 2007.

[63] K. Darwish and H. Mubarak, "Farasa: Fast and Accurate Arabic Word Segmenter," [Online], Available: http://alt.qcri.org/farasa/segmenter.html, Accessed: 9 Feb. 2017.

[64] E. Gabrilovich and S. Markovitch, "Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis," Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07), pp. 1606-1611, 2007.

[65] D. Ștefănescu, R. Banjade and V. Rus, "Latent Semantic Analysis Models on Wikipedia and Tasa," Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC'14), pp. 1417-1422, 2014.

[66] F. M. Belém, C. S. Batista, R. L. Santos, J. M. Almeida and M. A. Gonçalves, "Beyond Relevance: Explicitly Promoting Novelty and Diversity in Tag Recommendation," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 7, no. 3, p. 26, 2016.

[67] J. Chakraborty and V. Verma, "Diversification in Tag Recommendation System Using Binomial Framework," Information and Communication Technology for Sustainable Development, Springer, pp. 423-430, 2018.

[68] B. Bi and J. Cho, "Automatically Generating Descriptions for Resources by Tag Modeling," Proceedings of the 22nd ACM International Conference on Information & Knowledge Management (CIKM '13), pp. 2387-2392, 2013.

[69] R. Prokofyev, A. Boyarsky, O. Ruchayskiy, K. Aberer, G. Demartini and P. Cudré-Mauroux, "Tag Recommendation for Large-scale Ontology-based Information Systems," Proc. of the International Semantic Web Conference, Springer, pp. 325-336, 2012.

[70] N. Niraula, R. Banjade, D. Ștefănescu and V. Rus, "Experiments with Semantic Similarity Measures Based on LDA and LSA," Proc. of the International Conference on Statistical Language and Speech Processing, Springer, pp. 188-199, 2013.

[71] C.-G. Chiru, T. Rebedea and S. Ciotec, "Comparison between LSA-LDA-lexical Chains," Proceedings of the 10th International Conference on Web Information Systems and Technologies (WEBIST), pp. 255-262, 2014.

[72] M. Allahyari and K. Kochut, "Semantic Tagging Using Topic Models Exploiting Wikipedia Category Network", 2016 IEEE 10th International Conference on Semantic Computing (ICSC), pp. 63-70, 2016.

[73] T. Bogers and A. van den Bosch: "Recommending Scientific Articles Using Citeulike," Proceedings of the ACM Conference on Recommender Systems, pp. 287-290, 2008.

[74] M. Sun, Y.-N. Chen and A. I. Rudnicky: "HELPR, A Framework to Break the Barrier across Domains in Spoken Dialog Systems," Dialogues with Social Robots, Springer, pp. 257-269, 2017.

[75] M. Maamouri, A. Bies and S. Kulick, "Diacritization: A Challenge to Arabic Treebank Annotation and Parsing," Proceedings of the Conference of the Machine Translation SIG of the British Computer Society, 2006.

[76] T. K. Landauer, "LSA as a Theory of Meaning," Handbook of Latent Semantic Analysis, vol. 3, 2007.

ملخص البحث:

لقد اكتسب موضوع عمل بطاقات اقتباس من النصوص اهتماماً متزايداً كطريقة للربط بين البيانات من شأنها أن تدعم استرجاع المعلومات وتصنيفها. ولحلّ المشكلات المتعلقة بالقيام بذلك يدوياً، ظهرت تقنيات لتسهيل الأمر على المستخدمين عن طريق توفير قائمة من البطاقات التي تقتبس من النصوص.

وتجدر الإشارة الى أن غالبية الطرق القائمة التي تستخدم لهذا الغرض إنما تركز على الحقل أو المجال، كما أنها تعالج نصوصاً طويلة. وتشكل الطرق المعتمدة على الحقل او المجال تحدياتٍ جمّة بسبب نقص المعرفة الشاملة والحسابات المعقدة التي تتضمنها.

علاوة على ذلك، قد ينطوي التعامل مع النصوص القصيرة على بعض الإشكاليات نظراً لصعوبة استخلاص السّمات الإحصائية منها. ومن حيث اللغة، انصبت الجهود المبذولة بهذا الخصوص على النصوص الإنجليزية. أما القيام بعمل بطاقات اقتباس من النصوص المكتوبة بالعربية فليس بالأمر اليسير؛ لصعوبة معالجة تلك النصوص وشُحّ مصادر المعرفة باللغة العربية.

هذا العمل يقترح طريقة للقيام بهذه المهمة بالنسبة للنصوص القصيرة بالعربية. وتستخدم الطريقة المقترحة موسوعة "ويكيبيديا" العربية كخلفية معرفية من أجل عمل بطاقات اقتباس مقترحة من نصوص قصيرة. ويستفاد من تحليل الدلالات الكامنة في الألفاظ في تحليل نصوص قصيرة من الموسوعة المذكورة وإيجاد فقراتٍ لها علاقة بالنصوص المدخَلة. بعدئذٍ يتم انتقاء البطاقات المتعلقة بالنصّ من العناوين والفئات الخاصة بتلك الفقرات ومن ثم ترتيبها وفق درجة علاقتها بالنصّ.

تم تقييم الطريقة المقترحة بناءً على التقديرات الممنوحة لها من الخبراء بتطبيق ذلك على (993) بطاقة. وأظهرت النتائج أن الطريقة المقترحة أحرزت معدّل متوسط دقة قدره (84.39%) ومتوسط رتبة عكسي قدره (96.53%) واشتملت الدراسة على مناقشة مستفيضة للنتائج التي توصلت إليها؛ لإلقاء الضوء على نقاط القوة والضعف للطريقة المقترحة.