Arabic web pages clustering and annotation using semantic class features

Hanan M. Alghamdi a,b,*, Ali Selamat b,c, Nor Shahriza Abdul Karim d

a Faculty of Computer Science, Umm Al-Qura University, Al-Gumhdh, Saudi Arabia
b Faculty of Computing, Universiti Teknologi Malaysia, UTM, Johor Bahru, Johor 81310, Malaysia
c UTM-IRDA Digital Media Center of Excellence, Universiti Teknologi Malaysia, UTM, Johor Bahru, Johor 81310, Malaysia
d Computer & Information Science Department, Prince Sultan University, 66833 Rafha Street, Riyadh 11586, Saudi Arabia

Available online 28 September 2014

KEYWORDS
k-Means;
Semantic similarity;
Text clustering;
Arabic webpage

Abstract To effectively manage the great amount of data on Arabic web pages and to enable the classification of relevant information are very important research problems. Studies on sentiment text mining have been very limited in the Arabic language because they need to involve deep semantic processing. Therefore, in this paper, we aim to retrieve machine-understandable data with the help of a Web content mining technique to detect covert knowledge within these data. We propose an approach to achieve clustering with semantic similarities. This approach comprises integrating k-means document clustering with semantic feature extraction and document vectorization to group Arabic web pages according to semantic similarities and then show the semantic annotation. The document vectorization helps to transform text documents into a semantic class probability distribution or semantic class density. To reach semantic similarities, the approach extracts the semantic class features and integrates them into the similarity weighting schema. The quality of the clustering result has evaluated the use of the purity and the mean intra-cluster distance (MICD) evaluation measures. We have evaluated the proposed approach on a set of common Arabic news web pages. We have acquired favorable clustering results that are effective in minimizing the MICD, expanding the purity and lowering the runtime.

© 2014 King Saud University. Production and hosting by Elsevier B.V. All rights reserved.

1. Introduction

The growth of Arabic web pages and the great amount of text contained in them, which hold unorganized informative data, urge the necessity to adopt solutions that can wisely manage these textual data (Elarnaoty et al., 2012). Because of the unstructured character of these texts, valuable knowledge cannot be efficiently understood by machines.

Many studies have been conducted to classify related information and to support the manipulation of texts available on the Internet. Document clustering is the most common
The semantic annotation can be used as a guide to understand and classify the document and reveal informative knowledge. In Nguyen et al. (2009) and Park and Lee (2012), the authors proposed a framework for clustering and labeling with the hidden topics of web documents. By revealing the hidden topics and preparing them for annotating clusters, more meaningful clusters can be produced, and the quality of clustering can be improved.

To the best of our knowledge, classifying documents according to semantic similarities for retrieving information from Arabic web pages is limited. In this research, we intend to extract the semantic features from Arabic web pages and cluster these pages according to the similarities of these features. We consider that a word that carries very strong semantic information can disclose hidden knowledge.

In the proposed method, we did not use any machine translation tools that may cause the loss of meaning or some semantic distortions that result from the wrong choice of words and language models (Larkey et al., 2004). Instead, we used available lexical resources for Arabic text to process Arabic language, such as Arabic VerbNet. The tool offers systematic investigation of the semantic/syntactic aspects of the morphological system. According to Hawwari et al. (2013), Arabic VerbNet is one of the lexical resources for Arabic verbs that provides large coverage for Arabic verb taxonomy with semantic aspects of the morphological system. The work of Mousser (2010), which is based on an English VerbNet project (Kipper et al., 2008), is a representation of Levin’s syntactic alterations into Arabic. In this research, we used Arabic VerbNet to find the semantic similarities between web pages. This resource gives essential information about the syntax and semantics of Arabic verbs by applying the concept of verb-classes. The current version of the work by Mousser (2010) has 202 classes populating 4707 verbs and 834 frames. These frames consider alternations where the verbs can appear. Every class is a hierarchical structure, providing syntactic and semantic information about verbs and pre-allocating them to subclasses.

3. Proposed model

The proposed model, as shown in Fig. 1, performs clustering with the semantic similarities of Arabic Web pages and produces document vectorization according to semantic features (density or the probability distribution) with the help of Arabic VerbNet lexical, and then it finds the semantic annotation of the resulting clusters.

It contains two main phases: (1) extracting semantic features and document vectorization to group Arabic Web pages
according to the semantic similarities and (2) showing the semantic annotation. There are three steps in this method, which are: (1) extraction of semantic class features, (2) document vectorization, and (3) clustering and annotation.

This model works as follows: the model takes unannotated documents (to be classified), it will recognize all opinion words by using semantic feature extraction, and then it will aggregate all of the words to give a semantic annotation to the document.

3.1. Extraction of semantic class feature

The task of feature extraction is to find a semantic correspondence between one or more semantic classes and a document. For example, consider the document contents and semantic class repository as shown in Fig. 2. This task must extract the features of semantic class found in the document in terms of the attribute/value. Therefore, it combines and refines the document’s terms and maps them to the target semantic class.

For each text, the verbs are extracted and grouped together based on the semantic class. The process follows the rule-based POS tagging constructed in Al-Shalabi and Kanaan (2004). Each word entry is tagged as a noun, a verb or stopwords. In this study, verbs will only be used to assign the semantic class.

The verb can be related to its semantic class using the lexicon-based task (Ahmed, 2009) in Arabic VerbNet (Mousser, 2010). In this task, each semantic class has a set of verb lexicon associated with it. This task is simply a search through a lexicon list. If the verb is found, an appropriate class is assigned to the extracted word or term in the document. By the end of this task, it will produce the features in vector space using the word and semantic class as the vectors. Consequently, each of the extracted documents may have more than one semantic class.

3.1.1. Using Arabic VerbNet

To generate semantic features from input verbs, an analyzer was implemented as part of our model to generate these features automatically. Arabic VerbNet frames contain the description of the syntactic and semantic information of each verb. The semantic meaning of the verb, such as the cause, emotional state, motion, and made of, are connected with each frame (Mousser, 2011).

The analyzer will extract the verbs found on Arabic VerbNet and the semantic frames related to these verbs and build a local frame database to be used with our model. The semantic information found on extracted frames will be used as a semantic class in our model. Therefore, the semantic classes will be related to verbs according to the local frames extracted from Arabic VerbNet. The semantic class of the verb can be defined as a semantic feature of this verb.

For each verb resulting from POS tagging, the analyzer will look up the verb in the local frame database and relate this verb to the appropriate semantic class. Because of polysemy, each verb can be related to more than one semantic class.

3.2. Document vectorization

The document vectorization task is shown in Fig. 3 that represents a document using the semantic class feature extracted from the previous step. The term “semantic class” means the classes of verbs as obtained from the VerbNet. The task carries out document vectorization, which converts each document text into vectors that characterize the contained semantic class features through the exploitation of the semantic class density or the probability distribution.
This task uses the semantic class features to provide a series of probabilities to which the document can be assigned according to a pre-specified set of semantic classes that are based on semantic class feature extraction. The utilization of vectorization in our method helps to transform text documents into a semantic class probability distribution or semantic class density in the vector space. As a result, these vectorizations can be used to calculate the semantic similarity between web pages.

3.2.1. Semantic class density

The distribution of the semantic class within a webpage may provide extra implicit knowledge. Semantic class density assumes that a semantic class’s density in sampled documents is a good approximation of its density in the complete database. The relevant documents have approximately the same density as semantic classes.

The semantic class density is the average frequency of one class over all of the documents. In this step, the weight of the semantic class \(clsi\) in the document \(doc_d\) is calculated as a formula for merging the class frequency of each semantic class and the total frequencies of all of the semantic classes to obtain the total weight of class \(clsi\) and is as follows:

\[
W(clsi|doc_d) = \frac{\mathcal{O}(clsi)}{\sum_{i \in doc_d} \mathcal{O}(clsi)}
\]  

(1)

where \(\mathcal{O}(clsi)\) is the total occurrence of class \(clsi\) in document \(doc_d\) and it is calculated as a sum of frequencies over all words as follows:

\[
\mathcal{O}(clsi) = \sum_i F(w_i) \Rightarrow w_i \in clsi, \text{ and } w_i \in doc_d
\]

(2)

If the density value of a semantic class \(clsi\) in a document \(doc\) is computed as, \(W(clsi|doc)\), then each document \(doc_d\) has \(x\) weights of semantic classes, which will represent the (document vectorization) \(DV(doc_d) = W(clsi_1|doc_d), W(clsi_2|doc_d), W(clsi_3|doc_d), \ldots W(clsi_x|doc_d)\).

3.2.2. Semantic class probability distribution

Every document can be represented by its probability distribution on the semantic classes as a feature vector space. The Bayes formula can be employed to calculate the probabilities to which the document can be assigned according to a pre-defined set of classes.

We calculate the probability of the documents for each semantic class \((cls_1, cls_2, cls_3, \ldots cls_x)\) using Eq. (3) (Isa et al., 2008; Mohammad et al., 2007). If the probability value for a document \(doc_d\) to be assigned to a semantic class \(clsi\) is computed as \(P(clsi|doc_d)\), then each document \(doc_d\) has \(x\) probability distributions \(P(clsi_1|doc_d), P(clsi_2|doc_d), P(clsi_3|doc_d), \ldots P(clsi_x|doc_d)\) as shown in Table 1. The detailed description of the Bayesian vectorization is given in Isa et al. (2008) as follows:

\[
P(clsi|w_i) = \frac{P(w_i|cls_i) \cdot P(cls_i)}{P(w_i)}
\]

(3)

where \(w_i\) is the word extracted from document \(doc_d\), \(z\) refers to the total number of words in \(doc_d\), \(cls_i\) refers to the semantic class number \(i\) and \(x\) is the total number of available semantic classes.

3.3. Clustering and annotation

The document vectorization is designed as an input to the \(k\)-means clustering for classification purposes. In this model, the document vectorization task is employed as a text representation model, where the document vectorization represents a text as understandable machine vectors and aids in the creation analysis and classification system. The outcomes of the clustering step are clusters with semantic informative data that can be used to add semantic annotation to the resulting clusters.

### Table 1  Semantic class probability distribution calculation.

| \(doc_j\) | \(P(clsi_1|w_3) + P(clsi_1|w_2) + P(clsi_1|w_1)\) | \(P(clsi_1|doc_1) + \cdots + P(clsi_1|doc_z)\) |
|---|---|---|
| \(doc_2\) | \(P(clsi_1|w_3) + P(clsi_1|w_2) + P(clsi_1|w_5)\) | \(P(clsi_1|doc_2) + \cdots + P(clsi_1|doc_z)\) |
| \(doc_d\) | \(P(clsi_1|w_1) + P(clsi_1|w_2) + P(clsi_1|w_4)\) | \(P(clsi_1|doc_d) + \cdots + P(clsi_1|doc_z)\) |
3.3.1. Clustering

In the clustering step, the squared Euclidean distance (Cha, 2007; Deza and Deza, 2006) is used to present the degree of closeness or separation of the target document to the chosen cluster. Clustering with the squared Euclidean distance metric is faster than clustering with the regular Euclidean distance (Fabbri et al., 2008). The squared Euclidean distance between two distributions $P(d_i)$ and $P(d_j)$ is calculated in Eq. (4) as follows:

$$\text{Sim}_{d_{e}} = \sum_{i=1}^{d} (P(d_i) - P(d_j))^2$$  

(4)

Afterward, the $k$-means will cluster the vectorized $doc_d$ to the suitable cluster that contains a similar attribute based on the probability distribution or density of the semantic classes. The documents found in every cluster are amassed depending on the likeness of the semantic classes recognized in each of them.

3.3.2. Semantic annotation

Semantic annotation can be defined as the process of inserting semantic tags in a document that allows the documents to be processed either by humans or by using automated software agents. The semantic annotation for Arabic web page is assigned based on the highest semantic feature relevance scores as a way to produce a picture about the knowledge contained and its semantics in the domain (Malik and Rizvi, 2011). The extracted feature from the first task helps specify the most relevant semantic annotation.

To annotate the related clusters, the appropriateness scores for every semantic feature discovered in the cluster are figured. Then, the annotation of the cluster is the mixture of the five topics with the highest scores. Hence, clusters with high scores demonstrate that the document is more related to the semantic topics with the highest scores. Therefore, clusters with high scores are selected based on the highest semantic feature relevance scores. Then, the annotation of the cluster is the mixture of the five topics with a weight stronger than the mean of cluster $k$.

For every semantic feature discovered in the cluster are figured. Then, the annotation of the cluster is the mixture of the five topics with the highest scores. Hence, clusters with high scores demonstrate that the document is more related to the semantic topics with the highest scores. Therefore, clusters with high scores are selected based on the highest semantic feature relevance scores. Then, the annotation of the cluster is the mixture of the five topics with a weight stronger than the mean of cluster $k$.

To define $SL_j^k$, the mean score among the corpus (Smith and Tesic, 2006) is used. These methods are explained as follows:

- Mean score among cluster $k$: This method calculates the average of the semantic topic weight to determine the relation of this topic to the cluster $k$. As a result, those semantic features with a weight stronger than the mean of cluster $k$ are the most important topics of this cluster sorted by weight from the most important to the least important. Let the mean value of semantic topic $j$ belong to cluster $k$ as in Eq. (5), where $\mu_{St}$ is the mean value of semantic topic $j$ among the corpus.

$$SL_j^k = \mu_{St_j}$$  

(5)

- Mean ratio among corpus (MRAC): This method is based on measuring the significance score of the semantic topic to all other documents found in the corpus (Smith and Tesic, 2006). Let the main ratio of the semantic topic $j$ belong to cluster $k$ as in Eq. (6), where $\mu_{St}$ is the mean value of semantic topic $j$ among the corpus.

$$SL_j^k = \frac{\mu_{St_j}}{\mu_{St}}$$  

(6)

4. Experimental set-up and evaluation

The goal of this assessment is to figure out the effect of the recommended model, which aims to find the semantic similarities between web pages and extract the semantic annotation on the cluster quality and performance. The default number of clusters is set as the same number of pre-assigned categories in the dataset and its multiples.

When we execute the tests, we need to prepare the collected web pages for the classification algorithms. The pre-processing stage is intended to gather and extract related web pages and to diminish the noise terms (undesired term inside the content) to simplify the technique of measuring the weighting of features. This phase consists of collecting the URL seeds of web sites as a dataset using a web extractor agent and a text pre-processing stage that incorporates tokenization and normalization, tagging, and stemming. The pre-processing phase is depicted clearly in Alghamdi and Selamat (2012). The remainder of this section clarifies the evaluation criteria and the test results.

4.1. Datasets

In this study, we have collected corpus from the archives of online Arabic newspapers because there are no common Arabic datasets available by which to test the proposed model. These newspapers are namely Al-Akhbar2, Alhayat3, Aldostor4, Gomhuria online5, Akhbar Alarab.Net6, Alriyadh7 and Saudi Times8. These online newspapers are commonly used for many applications related to Arabic text language (Alsalseem, 2011, 2013; Karima et al., 2012; Saleh and Al-Khalifa, 2009). The collected datasets contain 753 documents with a dissimilar length of words. There are six categories to which the documents belong as explained in Table 2. We employed a web extractor agent (Easy Web Extract version 2.79) to extract the textual data from these web pages.

Table 2: Arabic dataset.

<table>
<thead>
<tr>
<th>Category name</th>
<th>Number of documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Political news</td>
<td>194</td>
</tr>
<tr>
<td>Economic news</td>
<td>133</td>
</tr>
<tr>
<td>Sports news</td>
<td>126</td>
</tr>
<tr>
<td>Social news</td>
<td>60</td>
</tr>
<tr>
<td>Cultural news</td>
<td>101</td>
</tr>
<tr>
<td>Technology &amp; Science news</td>
<td>139</td>
</tr>
<tr>
<td>Total</td>
<td>753</td>
</tr>
</tbody>
</table>

4 Aldostor online news available at http://dostor.org/.
5 Gomhuria online available at http://www.gomhuriaonline.com/.
7 Alriyadh online newspaper available at http://www.alriyadh.com/section.home.html.
9 Easy Web Extract webpage (http://webextract.net/).
4.2. Evaluation criteria

The quality of the clustering result using the above datasets is evaluated using three evaluation measures, namely, purity measure, mean intra-cluster distance (MICD) and Davies–Bouldin index (DBI). These measures are widely used to evaluate the performance of unsupervised classification algorithms (Chawla and Gionis, 2013; Forsati et al., 2013; Huang, 2008; Rana et al., 2013). These evaluation measures are computed as follows:

- Purity measure: this measure is used to estimate the coherence of a resulting cluster. Our approach evaluates the degree to which a cluster encloses documents from a particular category. The purity of a single cluster $C_i$ of size $e_i$ is formally defined in Eq. (7):

$$\text{Purity}(C_i) = \frac{1}{e_i} \max_{h} e_i^h$$  \hspace{1cm} (7)

where $\max_{h} e_i^h$ represents the main category in cluster $C_i$ and $e_i^h$ that corresponds to the number of documents in cluster $C_i$ that annotate to category $h$. In an optimal cluster that consists of group documents from a single category, its purity rate is one (Huang, 2008).

- Mean intra-cluster distance (MICD): This measure is the distance between data vectors and its cluster center where the low MICD signifies a compact cluster and the one with a high MICD signifies a loose cluster (Rana et al., 2013). The MICD is calculated in Eq. (8):

$$\text{MICD} = \frac{1}{N_k} \sum_{i \in C_k} \|c_i - \mu_i\|^2$$  \hspace{1cm} (8)

- Davies–Bouldin index (DBI): This measure aims to find well-separated, compact clusters. It takes into account within cluster vectors the variance and distance between clusters centers (Demiriz et al., 1999). The smaller value of DBI shows a better clustering result. It has been found to be among the best indices (Arbelaitz et al., 2013; Rendón and Abundez, 2011). The DBI is calculated in Eq. (9):

$$\text{DBI} = \frac{1}{k} \sum_{i=1}^{k} \max_{j \neq i} \left\{ \frac{\text{diam}(c_i) + \text{diam}(c_j)}{\|\mu_i - \mu_j\|^2} \right\}$$  \hspace{1cm} (9)

where the diameter of a cluster is defined as:

$$\text{diam}(c_i) = \frac{1}{N_i} \sum_{x \in c_i} \|x - \mu_i\|^2$$  \hspace{1cm} (10)

where $\text{diam}(c_i)$ and $\text{diam}(c_j)$ are the average distances of all data vectors in clusters $i$ and $j$ to their respective cluster centroids in Eq. (10), $\mu_i$ is the center of cluster $c_i$ consisting of $N_i$ points and $\|\mu_i - \mu_j\|^2$ is the Euclidean distance between these centroids.

A strong structure and good clustering have a small MICD (similar data vectors are grouped together), smaller DBI (well separated compact clusters) and high purity.

5. Results

The results of applying the proposed model are gathered from two experiments. The first experiment is performed to show the fitness of the proposed solutions compared to the standard $k$-means. The second experiment is performed to illustrate the output of the semantic annotation process, where we can label the resulting clusters.

Fig. 4 demonstrates the results of the proposed approaches using a purity evaluation. Document vectorization (semantic class probability distribution or semantic density) with $k$-means is adequate for creating more coherent clusters that are well divided in relation to the categories. Document vectorization with $k$-means signifies that the clusters have high purity scores.

Fig. 5 shows the comparison results based on an MICD evaluation. The low value of MICD means that all points in the cluster are close to each other. The resulting clusters using vectorization (semantic class probability distribution or semantic density) with $k$-means appear to be more compact. The proposed model tends to outperform the standard $k$-means.

The comparison results based on the DBI evaluation are shown in Fig. 6. The smaller value of DBI signifies that there

![Figure 4](image-url)  \hspace{1cm} Figure 4  \hspace{0.5cm} Purity results.
is a good separation distance between clusters and that the distances between points in the cluster and its center are small. The resulting clusters using vectorization for either semantic class probability distribution or semantic density with $k$-means appear to have a smaller value of DBI, which means the proposed approach outperforms the standard $k$-means.

The consumed time for comparing approaches is shown in Fig. 7, where the time elapsed is measured in seconds. We can see that the runtime rises gradually when the number of clusters increases. The runtime of the standard $k$-means is substantially longer than the time consumed by the other two solutions. In contrast, the time elapsed using document
vectorization (semantic class probability distribution or semantic density) with $k$-means is substantially shorter. The vectorization with semantic density consumed 85 s to simply classify 753 documents into 54 clusters using this solution.

The semantic annotations for cluster number 6 when using (semantic class density or semantic class probability distribution) vectorization with $k$-means are shown in Fig. 8 and Fig. 9, respectively. Each cluster is represented by the five highest and relevant semantic features along with the percentage of each of them. The relevance score is based on two variables: mean score of the topic among cluster $k$ and mean ratio of the topic among the corpus.

6. Discussion and conclusion

The presented results show that the proposed solutions are reasonably accurate and fast. Through the proposed document vectorization solutions with $k$-means, we have succeeded in increasing the purity and decreasing the MICD and BDI compared to the standard $k$-means algorithm. Furthermore, we managed to lower the runtime using the proposed solutions. Next, using the proposed document vectorization with $k$-means, the dimension of the documents is reduced from $753 \times 4681$ to $753 \times 131$. Hence, the document vectorization in our method helps to transform text documents into a semantic class probability distribution or semantic class density in the vector space.

Moreover, the document vectorization allows us to represent the web pages according to semantic class features, which are later used to calculate the semantic similarity between web pages. The semantic annotations in these web pages reveal informative exposition about the communications used within these pages. As shown in Fig. 8 and Fig. 9, respectively, we can see that each cluster can be labeled with five semantic features with different percentages according to the verbs found in each cluster. Consequently, the suggested solutions are able to show the semantic features shared between similar web pages that are grouped together in one cluster.
We believe that using the proposed approach is a promising technique to classify Arabic web pages according to the semantic similarities between them with a low runtime and an accurate performance. This approach is meant to enhance the document representation models for text clustering based on semantic similarities. For future work, we plan to utilize the proposed approach to extract the semantic orientation of Arabic web pages related to terrorism and extremism.

Acknowledgments

The authors would like to extend their thanks to the Ministry of Education, Malaysia, Universiti Teknologi Malaysia (UTM) under Vot 03H02, Umm Al-Qura University (UQU) and the Ministry of Higher Education, Saudi Arabia for supporting this research.

References


