

# Multi-label Arabic text categorization: A benchmark and baseline comparison of multi-label learning algorithms

Bassam Al-Salemi<sup>a,\*</sup>, Masri Ayob<sup>a</sup>, Graham Kendall<sup>b</sup>, Shahrul Azman Mohd Noah<sup>a</sup>

<sup>a</sup> Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Malaysia

<sup>b</sup> School of Computer Science, University of Nottingham, UK

## ARTICLE INFO

### Keywords:

Multi-label learning  
Arabic text categorization  
RTAnews  
Multi-label benchmark

## ABSTRACT

Multi-label text categorization refers to the problem of assigning each document to a subset of categories by means of multi-label learning algorithms. Unlike English and most other languages, the unavailability of Arabic benchmark datasets prevents evaluating multi-label learning algorithms for Arabic text categorization. As a result, only a few recent studies have dealt with multi-label Arabic text categorization on non-benchmark and inaccessible datasets. Therefore, this work aims to promote multi-label Arabic text categorization through (a) introducing “RTAnews”, a new benchmark dataset of multi-label Arabic news articles for text categorization and other supervised learning tasks. The benchmark is publicly available in several formats compatible with the existing multi-label learning tools, such as MEKA and Mulan. (b) Conducting an extensive comparison of most of the well-known multi-label learning algorithms for Arabic text categorization in order to have baseline results and show the effectiveness of these algorithms for Arabic text categorization on RTAnews. The evaluation involves four multi-label transformation-based algorithms: Binary Relevance, Classifier Chains, Calibrated Ranking by Pairwise Comparison and Label Powerset, with three base learners (Support Vector Machine, *k*-Nearest-Neighbors and Random Forest); and four adaptation-based algorithms (Multi-label *k*NN, Instance-Based Learning by Logistic Regression Multi-label, Binary Relevance *k*NN and RFBoost). The reported baseline results show that both RFBoost and Label Powerset with Support Vector Machine as base learner outperformed other compared algorithms. Results also demonstrated that adaptation-based algorithms are faster than transformation-based algorithms.

## 1. Introduction

The rapid increase of internet websites, users, online storage providers and social media increase the amount of data available on a daily basis. The International Data Corporation (Gantz and Reinsel, 2012) reports that the digital data on the internet will grow to 40,000 exabytes in 2020 from 130 exabytes in 2005. These data are usually unstructured and managing and organizing these data requires an accurate automatic text categorization system. For this reason, text categorization remains an important research topic and receives a lot of attention from the research community and industry.

In the classical text categorization, the multiclass machine learning algorithm, e.g. Naïve Bayes (Tang, Kay, and He, 2016), Support Vector Machine (Tong and Koller, 2001) and Random Forest (Wu, Ye, Zhang, Ng, and Ho, 2014), are used to build text categorization systems capable of assigning a given text to only one category. However, the texts in nature can belong to more than

\* Corresponding author.

E-mail address: [bassalemi@ukm.edu.my](mailto:bassalemi@ukm.edu.my) (B. Al-Salemi).

<https://doi.org/10.1016/j.ipm.2018.09.008>

Received 27 February 2018; Received in revised form 27 September 2018; Accepted 29 September 2018  
0306-4573/ © 2018 Elsevier Ltd. All rights reserved.

one category. For example, a news article under “*politic*” category may appear under other categories, e.g., “*economic*” and “*technology*”. While the state-of-the-art learning algorithms can predict only one label of a given example; they cannot be directly employed to solve the multi-label text categorization problem, in which each document belongs to more than one category (Elghazel, Aussem, Gharroudi, and Saadaoui, 2016). That makes solving the multi-label text categorization problem a real challenge.

Many works have addressed the multi-label learning tasks in general. The simplest solution is to composite the multi-label task into many single-label subtasks; then a multiclass learning algorithm is used to solve each subtask. In this regard, many learning methods had been introduced in the literature. e.g. Binary Relevance (Boutell, Luo, Shen, and Brown, 2004), Classifier Chains (Read, Pfahringer, Holmes, and Frank, 2011), Label Powerset (Tsoumakos and Vlahavas, 2007), Ranking by Pairwise Comparison (Hüllermeier, Fürnkranz, Cheng, and Brinker, 2008) and Calibrated Ranking by Pairwise Comparison (Fürnkranz, Hüllermeier, Mencia, and Brinker, 2008). Another solution for the multi-label task is to adapt the single-label learning algorithm to address the multi-label task directly. Many algorithms have been proposed in the literature that extended the state-of-the-art single-label learning algorithm. For example, MLkNN (Zhang and Zhou, 2007), BRkNN (Spyromitros, Tsoumakos, and Vlahavas, 2008), and IBLRML (Cheng and Hüllermeier, 2009) are multi-label algorithms extended from kNN. AdaBoost.MH (Schapire and Singer, 2000), and its variants MP-Boost (Esuli, Fagni, and Sebastiani, 2006), and RFBoost (Al-Salemi, Noah, and Ab Aziz, 2016) are multi-label Boosting algorithms, extended from AdaBoost (Freund and Schapire, 1997).

The multi-label learning algorithms have been widely used and investigated to address the text categorization problem in many languages, especially English. However, for Arabic only some insufficient recent studies have been done (Ahmed, Shehab, Al-Ayyoub, and Hmeidi, 2015; Hmeidi, Al-Ayyoub, Mahyoub, and Shehab, 2016; Shehab, Badarneh, Al-Ayyoub, and Jararweh, 2016; Taha and Tiun, 2016). Arabic is the native language of 380 million speakers (Mubarak and Darwish, 2014). It has a vast vocabulary and complex morphology (Abdul-Mageed, 2017; Romeo et al., 2017). As the sixth official language of the United Nations (Eldos, 2003), the online data in Arabic increases daily, and that require the need to develop efficient automatic text categorization systems. Arabic text categorization had been widely studied, however, the majority of these studies are concerned with the classical single-label text categorization. Only a few recent studies consider multi-label Arabic text categorization, which have been conducted on small, inaccessible and non-benchmark datasets. Therefore, this work aims to (a) introduce a new Arabic multi-label dataset for text categorization and any similar supervised learning task, and (b) conduct a comparative empirical evaluation of the well-known multi-label learning for Arabic multi-label text categorization and present the baseline results.

The proposed dataset was collected from Russia Today Arabic news portal. The reasons for choosing this news portal are (a) first, each news article has a unique identifier. This makes it easier for us to access and fetch the articles by their identifiers. This feature is not available in most of the popular news portals, such as Reuter, BBC, and CNN. And (b) because in Russia Today news portal, each news article enclosed with a set of keywords. These keywords help to automatically annotate the collected articles, as annotating the news articles manually is extremely difficult when dealing with a large number of articles. The choose of Russia Today Arabic news portal for collecting the proposed dataset has no any political tendencies or on the basis that it is a trustworthy news site. The main purpose of this study is to contribute to the research community by proposing a multi-label dataset regardless of the source used.

The rest of the paper is organized as follows: The second section offers a comprehensive literature study of multi-label learning algorithms and the related works in Arabic multi-label text categorization. In the third section, the collected dataset is introduced and described. The fourth section briefly introduces the multi-label methods used for the evaluation. Section five presents the experimental study and discusses the experimental results obtained. Finally, the sixth section concludes the paper's contributions and outcomes with some future directions.

## 2. Related work

This section discusses multi-label learning algorithms and relevant studies on multi-label Arabic text categorization.

### 2.1. Multi-label learning algorithms

The approaches to solving the task of multi-label problems can be grouped into two categories: problem transformation methods and algorithm adaptation methods (Tsoumakos and Katakis, 2007; Tsoumakos, Katakis, and Vlahavas, 2010). Problem transformation methods convert the multi-label problem into a set of single-label classification problems. Then, the supervised machine learning classifier that is initially established for single-label classification is used. The outputs of the single-label classifier are then combined to answer to the original task of multi-label classification. Algorithm adaptation methods extend the single-label classification techniques such that they can handle multiple labels directly.

#### 2.1.1. Problem transformation approaches

One of the most well-known transformation-based methods is Binary Relevance (BR; Boutell et al., 2004). BR works based on one-versus-all strategy; the multi-label task is decomposed into multiple binary tasks equivalent to the number of labels. Then, a single-label learning algorithm is used to solve each subtask. Even though it is a straightforward and simple method, BR has been criticized as failing in labels correlation, as each label is learned independently (Read et al., 2011). This limitation was managed in (Read et al., 2011) by proposing a Classifier Chain (CC) method. CC works by linking the binary classifiers in a randomly-ordered chain. Then, to tackle labels correlation, for a given example, each binary classifier incorporates the labels predicted by the previous classifiers as additional information.

Another approach to transforming the multi-label problem is by considering each multiple labels as combined single labels. An

example of this method is Label Powerset (LP; Tsoumakas and Vlahavas, 2007). In LP, each member of the power set of labels in the training set is considered as a single label. The number of the combined label in LP has an upper bound of  $2^m$ , the power set of  $m$  single labels. However, some combined labels will have a few training examples, which will negatively affect the classification performance. Read (2008) tackled this problem by pruning the infrequent atomic labels. Even though this procedure will improve the classification performance, the dataset will lose some of its multi-label structure.

In a Pairwise Comparison (PC) approach (Hüllermeier et al., 2008), the dataset with  $m$  single labels is transformed into  $m(m-1)/2$  binary datasets, one for each pair of labels. Fürnkranz et al. (2008) improves on this method by proposing calibrated ranking by pairwise comparison (CRPC). In CRPC, another  $m$  BR classifiers are added to the  $m(m-1)/2$  pairwise binary classifiers. Then, the artificial calibration label is used to represent the split-point between relevant and irrelevant labels. Even though, this setting will increase the classification performance, the training time would significantly increase compared to the BR method (Loza Mencía, Park, and Fürnkranz, 2010).

### 2.1.2. Algorithm adaptation approaches

Several studies had been conducted to extend the single-label learning algorithms to solve multi-label tasks. The first adaptation-based multi-label algorithms proposed in the literature are AdaBoost.MH and AdaBoost.MR by Schapire and Singer (1999) these were extended from the well-known boosting algorithm AdaBoost (Freund and Schapire, 1997). AdaBoost.MH is designed to minimize training Hamming loss, while AdaBoost.MR is designed to produce the hypotheses based on ranking the labels and placing the correct labels at the top of the ranking. The experimental results of a study conducted by Schapire and Singer (2000) showed that AdaBoost.MH outperformed AdaBoost.MR.

Several multi-label algorithms have been adapted from the instance-based learning algorithms k-Nearest-Neighbors (kNN). Zhang and Zhou (2007) proposed the well-known multi-label algorithm “Multi-Label kNN” (MLkNN). MLkNN works by assigning a set of labels to a given example based on the prior and posterior probabilities for the frequency of each label within the kNN. Cheng and Hüllermeier (2009) adapted the Instance-Based Learning algorithm by Logistic Regression (IBLR; Aha, Kibler, and Albert, 1991) for multi-label learning by proposing IBLR-ML which, when added to IBLR, the logistic regression which captures the inter-dependencies between the labels using the labels of neighbor examples as extra attributes in a logistic regression scheme.

Although they are simple multi-label learning methods, transformation-based methods still depend on single-label classifiers. The large number of single-label classifiers makes it hard to decide what transformation method is the state-of-the-art for multi-label classification. In this regard, the adapted multi-label learning algorithms could be a good alternative, as the single-label algorithm is adapted to directly solve the multi-label problem.

## 2.2. Multi-label Arabic text categorization

As mentioned earlier, single-label Arabic text categorization has been well studied, and many studies have been presented in the literature. However, for multi-label Arabic text categorization, only a few studies had been conducted, which are reviewed in this subsection.

In Ahmed et al. (2015), a study was conducted on Arabic multi-label text categorization using transformation-based approaches. A dataset of 10,000 documents distributed over five categories (“arts”, “sport”, “politics”, “economy” and “science”) was used for the evaluation. The dataset was collected from the BBC Arabic news portal, and the tags (keywords) of each news article were used as categories. An experimental evaluation of six transformation-based methods showed that Label Combination with SVM as a base learner yielded the best categorization accuracy.

The same dataset used in (Ahmed et al., 2015) was used in (Taha and Tiun, 2016) to evaluate the BR transformation-based method, with three base classifiers (NB, SVM, and kNN) for Arabic multi-label categorization. In their study, Taha and Tiun (2016) also analyzed the effect of feature selection on the classification performance. Three feature selection metrics were compared (Chi-square, mutual information, and odds ratio). The experimental results showed that using the combination of the evaluated single-label classifiers as a base learner with Chi-square yielded the best performance.

In Hmeidi et al. (2016), an Arabic multi-label text categorization system based on lexicons was proposed. Three transformation-based methods were compared. An Arabic multi-label dataset comprising 8800 texts collected from the BBC news portal, divided into 7390 for training and 1410 for the test. The experimental results showed that the stemmed lexicons extracted from the corpus yielded better accuracy than the original lexicons (with no stemming).

In Shehab et al. (2016) a multi-label Arabic dataset was collected from the CNN Arabic news portal. The dataset consisted of 10,997 documents distributed over six categories (“economics”, “Middle East”, “world”, “sports”, “science and technology”, and “miscellaneous”). In their experiment, the authors used the BR transformation with three single-label learning algorithms (kNN, Random Forest, and Decision trees). The experimental results showed that Decision trees outperformed both kNN and Random Forest.

All the studies above are concerned with Arabic multi-label text categorization. However, except Shehab et al. (2016), all the datasets are not publicly available. The dataset used in Shehab et al. (2016) was stored as an MS-Access database file<sup>1</sup> which makes it inapplicable to be easily evaluated in any of the existing multi-label learning tools. Moreover, all these studies used transformation-based methods, the adaptation-based methods, e.g., MLkNN, IBLRML, and AdaBoost.MH have not been investigated for Arabic multi-

<sup>1</sup> [https://www.researchgate.net/profile/Nizar\\_A\\_Ahmed/publication/301495171\\_Multi-label\\_Arabic\\_Data/data/5716735908aeeefeb022c37ea/Dataset4.mdb](https://www.researchgate.net/profile/Nizar_A_Ahmed/publication/301495171_Multi-label_Arabic_Data/data/5716735908aeeefeb022c37ea/Dataset4.mdb), last accessed 11/12/2017

label text categorization. Furthermore, all datasets used are small and balanced (the documents are approximately portioned evenly over the different categories), while most texts are imbalanced. e.g., texts in the real-world that talk about “space” occur less frequently than texts that mention “politics”.

These issues have motivated us to propose a large, imbalanced and accessible benchmark dataset of multi-label Arabic texts. Baseline results are also reported by evaluating four transformation-based multi-label methods with three single-label base learners, and four adaptation-based methods. The most widely used evaluation measures are used to evaluate the classification performance, making the baseline results more reliable and more easily comparable.

### 3. Dataset

The lack of publicly available datasets of Arabic multi-label texts leads to less research focus on Arabic multi-label learning texts categorization. This work aims to present an Arabic multi-label dataset as a set of multi-label texts to be used for text categorization and related supervised learning tasks.

#### 3.1. Dataset collection and preparation

The dataset presented in this work, namely “RTnews”, was collected from Russia Today Arabic news portal.<sup>2</sup> In RT Arabic portal, each news article has a unique identifier (ID). For example, the news article [https://arabic.rt.com/middle\\_east/912044-فيادي-الحشد-العراقي-الشعبي-البغدادي-الولايات-المتحدة-العراق](https://arabic.rt.com/middle_east/912044-فيادي-الحشد-العراقي-الشعبي-البغدادي-الولايات-المتحدة-العراق) has a unique ID “912,044”. To access this article, we need to write the URL in this form, “[https://arabic.rt.com/middle\\_east/912044](https://arabic.rt.com/middle_east/912044)”. This feature in RT news makes it easier to access the articles by simply changing the articles’ IDs.

For collecting the news articles, a web crawler we developed automatically generates different article links based on their IDs, and visits the websites, then stores the retrieved articles in HTML format using the ID as the filename. Interestingly, the PC was running for more than 48 hours through a high-speed internet connection to perform this task, which represents a significant saving of time in collecting the dataset if we do it manually.

The total number of collected texts before filtration (discussed later in this section) was 72,529 news articles, which represent nearly all articles published in 2016. The articles were saved in HTML format, taking 3.94 GB of storage space. Collecting this number of texts from the website, retrieving the body texts of the articles from HTML files and performing the annotation manually is costly and time-consuming tasks. Therefore, we have developed our crawler to perform all these tasks automatically. The pseudocode of the proposed crawler is shown in Fig. 1. An article identifier (*artID*) is firstly initialized with an ID given as an input (*inID*), and the article's URL is generated. The crawler tries to access the URL. If the URL is available in the news portal, the crawler visits that URL and fetches the accessed web page and return the content. Next, the crawler saves the article in the local storage using its ID as a file name and “HTML” as an extension. Then, for the next iteration, the value of *artID* is increased by one and the same procedures occur until reaching the needed number of articles.

The main challenge of preparing the datasets for text categorization and other machine learning tasks is to perform dataset annotation (Ahmed et al., 2015). Annotating such large dataset is time-consuming if it is performed manually. Moreover, the human annotators can be inconsistent. Luckily, news articles in RT Arabic website systematically annotated at the time it is posted online with a set of related topics of each article. Fig. 2 shows a news article from RT news portal, where at the end of the article, they include its related topics. Henceforth, these topics are used as categories (labels) of the article.

The HTML file of each news article has too many unnecessary tags and texts. Therefore, we need to retrieve the necessary contents which are the title, body text, and the related topics. Since all collected HTML files have the same structure, it is easier to retrieve the required contents by writing a suitable algorithm. The algorithm simply checks the HTML content line by line. The line containing the substring “< meta property = \og:title\ content =” contains the article's title. The block of text starting with “< /script > < div class = “fb-like” and ends with the first “< script type = “text/javascript” contains the body text of the article. The body text is ended with the related topics, separated by the tag “< /a > < a text >”. Note that, this is the HTML structure at the time this paper was written, it could be changed in the future by the website developers.

Once the main contents of the articles are retrieved from HTML files, a *regular expression* eliminates the symbols and non-Arabic alphabet. Each article (containing the title and the body text) are stored as textual files, and the related topics (categories) are treated as folders, in which each text file can be saved in more than one folder based on the multiple categories of the article.

The resultant dataset consists of 72,529 texts distributed over 271 different categories. However, most of these categories are redundant. Therefore, the categories which are the names of countries, places, organizations, persons, and companies, and the categories that have a few examples are all removed. As a result, out of the 271 categories, only 40 categories are retained.

Moreover, while the annotations of the articles were performed systematically, a filtration task is required to remove the unrelated articles from each category. To achieve that, a set of the most frequent keywords are firstly extracted from the texts under each category, then all documents that have similarity to the set of keywords less than a certain threshold were removed. To be more specific, the number of keywords was set to 50, the similarity metric used was the *Jaccard* similarity metric (Jaccard, 1908) and the threshold was (0.60). We selected these values (the number of keywords and the similarity threshold) based on our findings, which were experimentally yielded the best performance. Our pre-experimental findings showed that choosing a threshold less than 0.60

<sup>2</sup> <https://arabic.rt.com/>

```

Input: initial ID (inID) , number of articles to be accessed (num)
Output: A set of saved articles

begin
  for counter ← 0 to num do
    artID ← inID + counter;
    URL ← "https://arabic.rt.com/middle_east/" + artID + " – "
    if IsAccessible(URL) then
      Visit (URL);
      Article ← fetch(URL);
      Save (Article, artID, "HTML");
    end if
  end for
end

```

Fig. 1. Pseudocode of web crawler.



Fig. 2. Snapshot of a news article from RT website.

yields a poor classification performance and choosing a threshold above 0.60 yields a small number of multi-label texts. The number of remaining texts was 23,837 distributed over 40 categories, which is used to establish RTAnews multi-label dataset.

### 3.2. Dataset statistics

Table 1 lists the numbers of texts per categories including the categories IDs; names in Arabic and in English; and the size of the training set and the test set for each category. The text distributions over categories (in Table 1) is illustrated in Fig. 3. It shows that the dataset is imbalanced where some categories have a large number of examples; meanwhile, some others are less well represented. This feature makes the dataset more reliable for evaluation, as the texts in the real-world are imbalanced.

The RTAnews dataset is divided into 15,001 texts for the training and 8836 texts for the test, approximately two-thirds of the dataset are randomly selected for training and the rest for testing. According to (Dobbin and Simon, 2011), this is the optimal splitting proportion for high dimensional datasets. Table 2 summarizes the dataset statistics. The total number of features after stemming and stop-word removal are 68,810 features. The Label Cardinality (LC) of RTAnews dataset, which measures how “multi-label” the dataset is (Tsoumakas et al., 2010), is 1.161 according to Eq. (1).

$$LC_S = \frac{1}{n} \sum_{i=1}^n |Doc_i| \quad (1)$$

, where  $S$  refers to the dataset,  $n$  is the number of documents and  $|Doc_i|$  is the number of categories assigned to a document  $Doc_i$ .

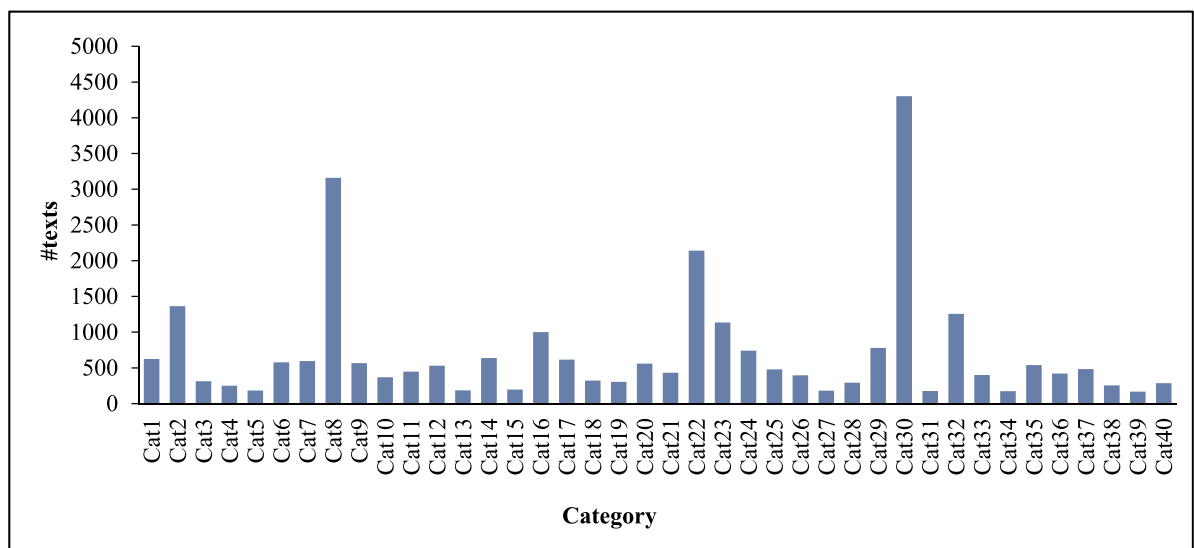
### 3.3. Dataset preprocessing

The typical text preprocessing tasks were applied to RTAnews dataset, which are normalization, tokenization, stop-word removal and stemming. Stemming is a fundamental task in text preprocessing in which the words are returned to their stems (Flores and Moreira, 2016). For stemming the texts in RTAnews, we have developed a light stemmer to remove the most frequented prefixes and suffixes. The set of prefixes and suffixes that were removed from the words are listed in Table 3.

After extracting the feature terms (stemmed words), the term frequency-inverse document frequency (tf-idf) weighting scheme (Karisani, Rahgozar, and Oroumchian, 2016; Wu, Gu, and Gu, 2017) was used for representing the texts as vectors of terms' weights. While feature selection is an important task to reduce the features dimensionality (Rehman, Javed, and Babri, 2017), the Chi-square statistic feature selection metric (Forman, 2003) was employed for this task. To use Chi-square for feature reduction, the score of a

**Table 1**  
RTAnews' categories and their portions of texts.

Category's index	Category's name in Arabic	Category's name in English	# texts	training	test
Cat1	أسلحة ومعدات عسكرية	Military equipment's	625	333	292
Cat2	أسواق النفط	Oil markets	1,364	882	482
Cat3	أولمبياد ريو دي جانيرو	Rio De Janeiro Olympic	314	172	142
Cat4	اتفاق إيران النووي	Iran nuclear deal	251	155	96
Cat5	استفتاء بريطانيا	Brexit	184	110	74
Cat6	اكتشافات	Inventions	577	332	245
Cat7	الأزمة الأوكرانية	Ukrainian crisis	595	352	243
Cat8	الأزمة السورية	Syrian crisis	3160	1,848	1,312
Cat9	الأزمة اليمنية	Yemeni crisis	566	349	217
Cat10	الاعتراف بدولة فلسطين	Recognition of the state of Palestine	369	240	129
Cat11	الانتخابات الأمريكية	American elections	447	290	157
Cat12	البحوث الطبية	Medical research	530	321	209
Cat13	البورصات	Stock market	186	121	65
Cat14	التقنية والمعلومات	Information and technology	639	401	238
Cat15	السياحة في العالم	World tourism	196	124	72
Cat16	المعارضة السورية	Syrian opposition	1,002	520	482
Cat17	الهجرة إلى أوروبا	Immigration to Europe	616	322	294
Cat18	أمراض	Diseases	323	193	130
Cat19	انقلاب تركيا	Turkey's coup	305	197	108
Cat20	تفجيرات	Bombings	559	327	232
Cat21	جرائم	Crimes	433	266	167
Cat22	جماعات مسلحة	Armed groups	2,142	1,304	838
Cat23	رياضات أخرى	Other sports	1,136	668	432
Cat24	صواريخ	Missiles	741	428	313
Cat25	طائرات حربية	Military aircraft	479	272	207
Cat26	عقوبات اقتصادية	Economic sanctions	397	224	173
Cat27	علم الآثار والتاريخ	Archeology and history	182	113	69
Cat28	عملية تحرير الموصل	Mosul operation	294	180	114
Cat29	فضاء	Space	780	328	302
Cat30	كرة القدم	Football	4,301	2,800	1,501
Cat31	كوارث جوية	Air disasters	177	110	67
Cat32	لاجئون	Refugees	1256	699	557
Cat33	مؤشرات اقتصادية	Economic indicators	400	240	160
Cat34	مخدرات	Drugs	175	104	71
Cat35	مشاهير	Celebrities	539	337	202
Cat36	مظاهرات	Demonstrations	422	265	157
Cat37	معلومات عامة	General information	484	280	204
Cat38	مناورات عسكرية	Military exercises	256	153	103
Cat39	موسيقى	Music	167	93	74
Cat40	هجمات باريس	Paris attacks	287	157	130



**Fig. 3.** The distribution of RTAnews articles over the different categories.

**Table 2**

Summarization of RTAnews dataset.

Number of categories	40	Average of documents per category	696
Training set size	15,001	Average of features per document	84
Test set size	8836	# documents with single label	20,374
Label cardinality	1.161	# documents with two labels	3114
# features after text preprocessing	68,810	# documents with more than two labels	349

**Table 3**

The most frequent Arabic suffixes and prefixes.

Prefixes	وال	ال	لا	بال	فال	كال	وبال	و
Suffixes	ها	هم	وا	ون	هما			

feature term  $t$  being assigned to a label  $l$  is computed according to Eq. (2).

$$\text{Chi}(t, l) \cong n \times \frac{(tp \times tn - fn \times fp)^2}{(tp + fp) \times (tp + fn) \times (tn + fn) \times (fp + tn)} \quad (2)$$

, where  $tp$  is the number of times  $t$  appears in  $l$ ,  $fp$  is the number of times  $t$  appears without  $l$ ,  $fn$  is the number of times  $t$  does not appear in  $l$  and  $tn$  is the number of times  $t$  does not appear without  $l$ . While Chi-square was essentially proposed for single-label problems, the features' scores of each label were obtained using the one-versus-all strategy. Thus, for multi-label texts, each text belongs to the current category treated as a different text in the rest of the categories.

After scoring all features using Chi-square, a different number of the high scoring features are selected for evaluation purposes which are 500, 1000, 2000, 3000, and 4000 features. Noting that, some versions of the dataset equivalent to the number of subsets of selected features will be available online in many formats, as will be described in the next section, to make it easier for researchers to evaluate the dataset and compare their results with our baseline results, and with the results of future work

### 3.4. Dataset formats

The stored format of the dataset uses a directory structure. The main folder of the dataset, which takes the dataset name “RTAnews”, contains two subfolders (“training” and “test”). Inside each subfolder, there are 40 subfolders, one for each category. Inside each category's folder, the texts are stored as textual files with “.txt” file extension. The texts with multiple categories are stored in their categories' folders with the same name.

To make it easier for evaluating it, RTAnews is also available in many formats compatible with the existing multi-label learning tools, which are MULAN (Tsoumakas, Spyromitros-Xioufis, Vilcek, and Vlahavas, 2011), MEKA (Read, Reutemann, Pfahringer, and Holmes, 2016) and MultiBoost (Benbouzid, Busa-Fekete, Casagrande, Collin, and Kégl, 2012). In addition, a single-label version of the RTAnews is also available for both WEKA (Hall et al., 2009) and RapidMiner (Mierswa, Wurst, Klinkenberg, Scholz, and Euler, 2006) tools. All formats of the RTAnews dataset are publicly available online on its web page.<sup>3</sup>

## 4. Multi-label learning methods

This section describes the evaluated multi-label methods that are categorized as transformation approaches and algorithm adaptation approaches.

### 4.1. Problem transformation approaches

Four common transformation-based multi-label approaches were evaluated for Arabic multi-label text categorization, on the RTAnews dataset, which are:

#### 4.1.1. Binary Relevance

Binary Relevance (BR; Boutell et al., 2004) is the most straightforward and widely used transformation method. In BR, the multi-label task is transposed to  $m$  binary tasks, one for each label. Then each binary classifier is trained individually for each label, such that the instances not relevant to the label at hand are considered as negative instances. Given an unseen instance, the unions of the binary classifiers that predict it as a positive instance are the multiple labels of that instance.

#### 4.1.2. Classifier Chains

The Classifier Chains (CC) method (Read et al., 2011) tackles the label's correlation problem that is not considered in the BR method. CC trains  $m$  binary classifiers, which  $m$  is the number of labels, then; these  $m$  classifiers are linked in a randomly-ordered

<sup>3</sup> <https://data.mendeley.com/datasets/322pzsdxyw/1>

chain. For classifying a given unseen sample, each binary classifier incorporates the labels predicted by the previous classifiers as additional information. This is accomplished by extending the feature vector that is associated with each classifier with the values of the previous labels in the randomly-order chain in the training phase. However, the disadvantage of CC is that, the labels are ordered randomly, which can lead to a poor classification performance.

#### 4.1.3. Calibrated Ranking by pairwise comparison

Ranking by Pairwise Comparison (RPC) methods works based on *one-versus-one* transformation. That is, the multi-label dataset with  $m$  labels are transformed into  $m(m-1)/2$  binary datasets, one per pair of labels (Hüllermeier et al., 2008). Each binary dataset uses the instances that belong to one of the two labels considered as positive instances and negative to the other label. Then, the binary classifiers are trained for each binary dataset. For a given unseen instance, all models are invoked, and a rank obtained by the counting votes for each label as a prediction of the given instance.

Fürnkranz et al. (2008) proposed Calibrated Ranking by Pairwise Comparison (CRPC). CRPC works by adding to the pairwise decomposition (the  $m(m-1)/2$  binary datasets) another  $m$  binary datasets, one per label. The key idea of this method is to introduce an artificial calibration label  $\lambda_0$ , which represents the split-point between relevant and irrelevant labels. The additional binary datasets trained by the BR approach to hold the parallel task setting, making the approach easily applicable to this setting. However, the problem of this method is that the training time would significantly increase compared to the BR method.

#### 4.1.4. Label Powerset

In Label Powerset (LP; Tsoumakas and Vlahavas, 2007), the multi-label problem is transformed into one single-label multiclass problem by considering each member of the power set of labels in the training set as a single label, called an atomic label. That is, the multiple labels of each document in the training set are combined as one atomic label. Then, the multiclass classifier is used, and the given unseen sample will be assigned to one of those atomic labels as multiple class labels. The number of atomic labels produced by LP transformation is upper bounded by  $2^m$ , the power set of  $m$  labels, which may lead to computational complexity when the labels density of the training set is high, and the dataset size is large. Moreover, some atomic labels are associated with rare examples and that makes the learning task difficult and may negatively affect the classification accuracy.

The base learners of the transformation-based approaches are the single-label multiclass classification algorithms. In this regard, the aforementioned multi-label approaches were evaluated with the following single-label classifiers as base learners:

**Support Vector Machine (SVM; Vapnik, 2013)** in its purest form, the linear SVM, is a hyperplane that separates a set of positive examples from a set of negative samples with maximum margin. In the linear SVM (as shown in Fig. 4), the margin is defined by the distance of the hyperplane to the nearest of the positive and negative examples.

Sequential Minimal Optimization (SMO; Platt, 1998) is a fast, improved and widely used variant of the well-known classification algorithm SVM. Instead of training the SVM using a sizeable quadratic programming (QP) optimization problem (Joachims, 1998), which is known as “Chunking”, SMO breaks this large QP problem into a set of smallest possible QP problems. Then, the small QP problems are solved analytically. This setting makes SMO faster than the traditional SVM and suitable for large datasets. In this study, the SMO is used for the evaluation as a base classifier for the transformation-based algorithm.

**k-Nearest-Neighbors (kNN; Aha et al., 1991)** is a typical learning algorithm that has been widely applied to address many multiclass learning problems. kNN is an instance-based learning algorithm that does not explicitly learn a model. Instead, it memorizes the training instances. Then, for predicting the label of a given instance, the algorithm looks for the  $k$  most similar training instances and votes for the appropriate label. The similarity is defined according to a distance metric between two data points. Given two data points  $x$  and  $y$  – represented as vectors  $x = (x_1, x_2, \dots, x_n)$  and  $y = (y_1, y_2, \dots, y_n)$ , respectively – the Euclidean distance is defined in Eq. (3).

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (3)$$

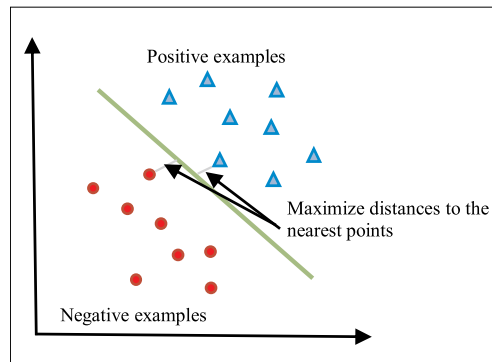


Fig. 4. A linear Support Vector Machine.

**Random Forest** (RF; Breiman, 2001) is as an ensemble learning algorithm based on Bagging (Breiman, 1996) with decision tree as a base classifier. In standard trees, each node is split using the best split among all variables. RF changes this setting by splitting each node using the best among a subset of predictors randomly chosen at that node. Also, each tree is constructed using a different bootstrap sample of the data. Using this strategy makes RF performs very well compared to many other classification algorithms (Breiman, 2001).

#### 4.2. Adapted algorithms

The following are the evaluated multi-label learning algorithms that are adapted from well-known single-label learning algorithms:

##### 4.2.1. Multi-label $k$ -Nearest Neighbors

MLkNN (Zhang and Zhou, 2007) is one of the most well-known multi-label algorithms. In MLkNN, the traditional kNN algorithm is adapted for multi-label learning. The maximum posterior (MAP) principle is used to determine the label set of a given instance based on prior and posterior probabilities for the frequency of each label within the kNN.

##### 4.2.2. Instance-Based Learning by logistic regression multi-label

IBLR-ML (Cheng and Hüllermeier, 2009) was adapted from the traditional multiclass algorithm kNN. In IBLR-ML, the Instance-Based Learning (IBL; Aha et al., 1991) is combined with logistic regression. It allows capture of the interdependencies between the class labels using the labels of neighbor examples as extra attributes in a logistic regression scheme so that the estimate of the multiple labels of a given instance is given.

##### 4.2.3. Binary Relevance kNN

Binary Relevance kNN (BRkNN; Spyromitros et al., 2008) is another multi-label classifiers adapted from kNN algorithm. Instead of using BR transformation with the kNN algorithm as one-against-all transformation, BRkNN extends the kNN algorithm so that independent predictions are made for each label, following a single search of the  $k$ -nearest neighbors. To avoid the case where BR outputs an empty set for any test instance, BRkNN considers the percentage of the  $k$ -Nearest Neighbors of the estimated label as a measure of the label confidence, it assigns the label to a given instance the one has the higher confidence.

##### 4.2.4. RFBoost

AdaBoost.MH (Schapire and Singer, 1999) is a multi-label boosting algorithm that is extended from the well-known AdaBoost (Freund and Schapire, 1997). As a boosting algorithm, AdaBoost.MH works by iteratively constructing a set of weak classifiers of decision stumps. The final classifier is then built as a composition of the selected weak classifier. The disadvantage of AdaBoost.MH is that the computational time is linear to the size of training features (Al-Salemi, Ab Aziz, and Noah, 2015a, b).

Al-Salemi et al. (2016) proposed an accelerated version of AdaBoost.MH, named “RFBoost”. In RFBoost, the feature ranking is used to rank the training features. Then, in each boosting round, only a small number of the top-ranked features are selected to build the weak classifiers. This strategy makes RFBoost faster and more accurate than AdaBoost.MH (Al-Salemi, Ayob, and Noah, 2018).

## 5. Experiments and results

The comparative evaluation of the multi-label approaches listed in Section 4 is conducted for Arabic multi-label text categorization on the proposed dataset RTAnews. In this section, the evaluation metrics used to evaluate the classification performance and experimental settings are first described. Then, the experimental results are presented, analyzed and discussed.

### 5.1. Evaluation metrics

To obtain the contingency matrix, Table 4, the estimated categories of each text in the test set are predicted and compared to the actual categories. Then, the cumulative values of the  $TP$ ,  $FP$ ,  $FN$ , and  $TN$  – where  $T$ ,  $F$ ,  $P$  and  $N$  represent “True”, “False”, “Positive” and “Negative”, respectively – are summed up.

The most common performance measures widely used to evaluate text categorization system are *Precision* ( $P$ ) and *Recall* ( $R$ ). For a category  $l$ ,  $P$  and  $R$  measures are defined according to Eqs. (4) and (5), respectively.

**Table 4**

The contingency matrix of text-category assignment.

		Actual category $l$	
		Positive	Negative
Predicted category $l$	Positive	True positive $TP_l$	False positive $FP_l$
	Negative	False negative $FN_l$	True negative $TN_l$

$$R_l = \frac{TP_l}{TP_l + FN_l} \quad (4)$$

$$P_l = \frac{TP_l}{TP_l + FP_l} \quad (5)$$

In fact, there is a trade-off between precision and recall in which if the prediction of each category is “true” for each test example, then the classifier will obtain high precision score and low recall score. Therefore, F1-measure (Rijsbergen, 1979) is used to measure the performance as a harmonic mean of  $R$  and  $P$ . Thus, for a category  $l$ , the categorization performance is measured using  $F1$  according to Eq. (6).

$$F1_l = 2 \frac{R_l \times P_l}{R_l + P_l} = 2 \frac{TP_l}{2TP_l + FN_l + FP_l} \quad (6)$$

To measure the global categorization performance, two types of measures are usually used. The macro-averaged, which gauged the category-level performance, and Micro-averaged, which measure the examples-level performance. The Macro-averaged merely is obtained as the average of all categories’ scores. Thus, the Macro-averaged of Recall, Precision and  $F1$  are computed as Macro – averaged Recall =  $\frac{1}{M} \sum_l R_l$ , Macro – averaged Precision =  $\frac{1}{M} \sum_l P_l$  and Macro-averaged  $F1 = \frac{1}{M} \sum_l F1_l$ , respectively.

To obtain the Micro-averaged scores, the accumulative summations of TP, FP and FN values for each text in the test set are first computed. Then, the scores are calculated as follows: Micro – averaged Recall =  $\frac{\sum_l TP_l}{\sum_l TP_l + \sum_l FN_l}$ , Micro – averaged Precision =  $\frac{\sum_l TP_l}{\sum_l TP_l + \sum_l FP_l}$  and Micro-averaged  $F1 = 2 \frac{\sum_l TP_l}{2 \sum_l TP_l + \sum_l FN_l + \sum_l FP_l}$ .

## 5.2. Experimental settings

The transformation-based methods are evaluated by means of the single-label base learners. From here onwards, the abbreviation of the transformation-based method followed by the abbreviation of the base learner, separated by a hyphen symbol, is used to refer to the transformation methods and its base learner. For example, “BR-SVM” refers to BR method with the base learner SVM. Noting that, each transformation-based method and each base learner is treated as an independent classifier. For instance, “BR-SVM” and “BR-kNN” are different multi-label classifiers.

All evaluated multi-label methods were performed using the MULAN tool (Tsoumakas et al., 2011), except for RFBoost where we developed a Java-based system to implement it. The dataset was formatted using Dense multidimensional ARFF format. In Dense format, each text is represented as a vector of data points, in which each data point is a pair of the index of the feature and its weight. For all evaluated methods, the default setting in MULAN were used, except for all instance-based methods (kNN and its variants), where the number of nearest neighbors  $k$  is set to 10. For RFBoost, the maximum number of iterations given is set to 4000, and the size of the ranked features is set to 100. The Chi-square which is used for feature selection, is also used to rank the features in RFBoost.

RTAnews was evaluated using different subsets of training features. The size of feature subsets chosen were 200, 500, 1000, 2000, 3000 and 4000 features. The number of experiments runs for each set of features is 16, one for each method. The total number of experiments for all sets of features and all methods were 96. To make the evaluation more reliable and applicable for statistical analysis, we concern the experimental results at each set of selected features is an independent observation. Hence, the Friedman test (Demšar, 2006) was used for analyzing the significant differences in the classification performance of the evaluated methods as defined in Eq. (7).

$$\chi_F^2 = \frac{12N_d}{k(k-1)} \left[ \sum_j R_j^2 - \frac{k(k+1)^2}{4} \right] \quad (7)$$

where  $N_d$  is the number of the observations,  $k$  is the number of the evaluated methods and  $R_j$  is the average rank of each algorithm.

By performing the Friedman test and obtaining the distribution with  $(k - 1)$  degree of freedom, the  $p$ -values are computed at a 5% significant level. Once the *null-hypothesis* that the methods have the same performance is rejected, the *Nemenyi post-hoc* test at 0.05 significance level is performed for comparing all methods with each other.

## 5.3. Results and discussion

While each performance evaluation measure gauges the performance in a different way than the other measures, we represent and discuss the results of each measure separately.

### 5.3.1. Precision measure

Fig. 5 illustrates the Macro-averaged and Macro-averaged Precision scores of all classifiers on the RTAnews dataset at varying sizes of training features. Each box plot indicates the mean, minimum and maximum of Precision scores obtained for each classifier at different sizes of the training features. It is clear that for the transformation-based methods BR, CLR, and CC, the base classifier kNN yielded the best results among all base classifiers (kNN, RF, and SVM). SVM outperformed both kNN and RF with LP method. Meanwhile, RF outperformed SVM with BR, CLR and CC methods.

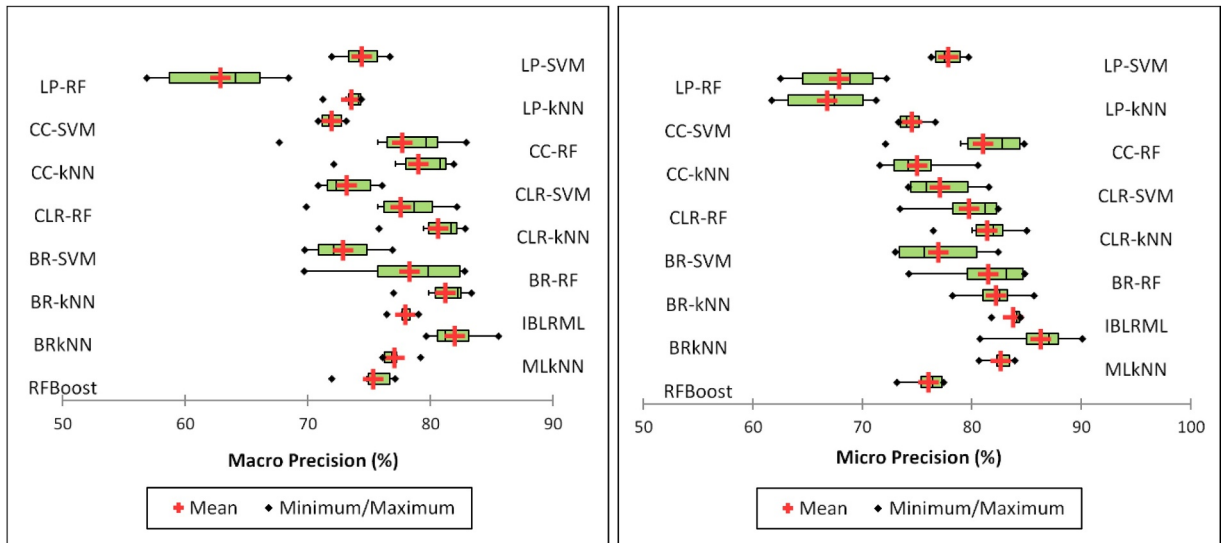


Fig. 5. Boxplots of Macro and Micro averaged Precision scores with varying subsets of training features.

For the adaptation-based algorithms (MLkNN, BRkNN, IBLRML, and RFBoost), BRkNN achieved the best Macro-averaged Precision scores, on average, followed by IBLRML while RFBoost achieved the worst performance. The best overall Macro-averaged Precision score (see Table 5) is 0.8557 obtained by BRkNN when the number of selected features is 1000 features. BR-kNN obtained the best second score of Macro-averaged Precision is 0.8334 with 2000 of selected features.

Regarding the Micro-averaged Precision results, as illustrated in Fig. 5, SVM outperformed both kNN and RF with LP method. RF outperformed both kNN and SVM with CC method. For both BR and CLR, kNN outperformed both RF and SVM. Among all transformation-based method, BR-kNN achieved the best Micro-averaged Precision score, which is 0.8566 with 4000 training features.

For the adaptation-based algorithms (MLkNN, BRkNN, IBLRML, and RFBoost), MLkNN obtained the best Micro-averaged Precision scores, whereas RFBoost yielded the lowest scores. The best Micro-averaged Precision score obtained by MLkNN was 0.901 with 4000 selected features, which is the best-obtained score overall.

Precision, according to and Eq. (5) is the fraction of the relevant texts being assigned to a category among all assigned texts by the system. The high score of the precision means that the number of texts that are irrelevant and designated by the classifier as relevant examples (*false positive*) is very small. However, Precision does not consider how many of the relevant examples to a category are correctly assigned, which makes it difficult to fully gauge classification performance.

Table 5

The best-obtained scores (%) of all compared methods.

Classifier	Macro-Precision		Macro-Recall		Macro-F1		Micro-Precision		Micro-Recall		Micro-F1	
	score	f. size	score	f. size	score	f. size	score	f. size	score	f. size	score	f. size
<b><u>Problem transformation approaches</u></b>												
BR-kNN	83.34	2000	47.60	200	56.22	200	85.66	4000	55.26	200	64.77	200
BR-RF	82.80	4000	57.43	200	62.23	200	84.81	3,000	61.99	200	67.56	200
BR-SVM	76.89	500	61.24	4000	65.88	4000	82.39	200	64.20	4000	69.89	500
CLR-kNN	82.83	2000	47.43	200	55.57	200	85.03	4000	55.33	200	64.21	200
CLR-RF	82.14	3000	59.31	200	63.50	200	82.41	3000	63.98	200	68.38	200
CLR-SVM	76.05	500	64.59	4000	67.84	4000	81.56	200	67.30	4000	70.72	4000
CC-kNN	81.89	4000	50.92	200	57.16	200	80.55	4000	57.10	200	63.52	200
CC-RF	82.91	4000	57.91	200	61.78	200	84.80	4000	62.44	200	69.01	500
CC-SVM	73.12	500	62.02	2000	66.58	4000	76.67	500	65.08	4000	69.57	500
LP-kNN	74.37	500	52.16	200	57.99	200	71.27	200	58.08	200	64.00	200
LP-RF	68.42	500	52.80	200	57.80	200	72.19	500	59.68	200	65.28	500
LP-SVM	76.70	4000	64.86	4000	69.79	4000	79.72	4000	67.39	4000	73.04	4000
<b><u>Algorithm adaptation approaches</u></b>												
RFBoost	77.11	4000	63.74	500	68.98	4000	77.44	3000	68.57	2000	72.57	3000
MLkNN	79.21	3000	48.37	1000	56.39	1000	83.92	4000	53.52	1000	64.97	1000
BRkNN	85.57	1000	43.40	200	52.68	200	90.10	4000	51.41	200	62.83	200
IBLRML	79.01	500	47.50	200	56.44	200	84.43	2000	51.88	500	64.08	500

Scores in boldface denote the best results overall among each group of approaches (problem transformation and algorithms adaptation)

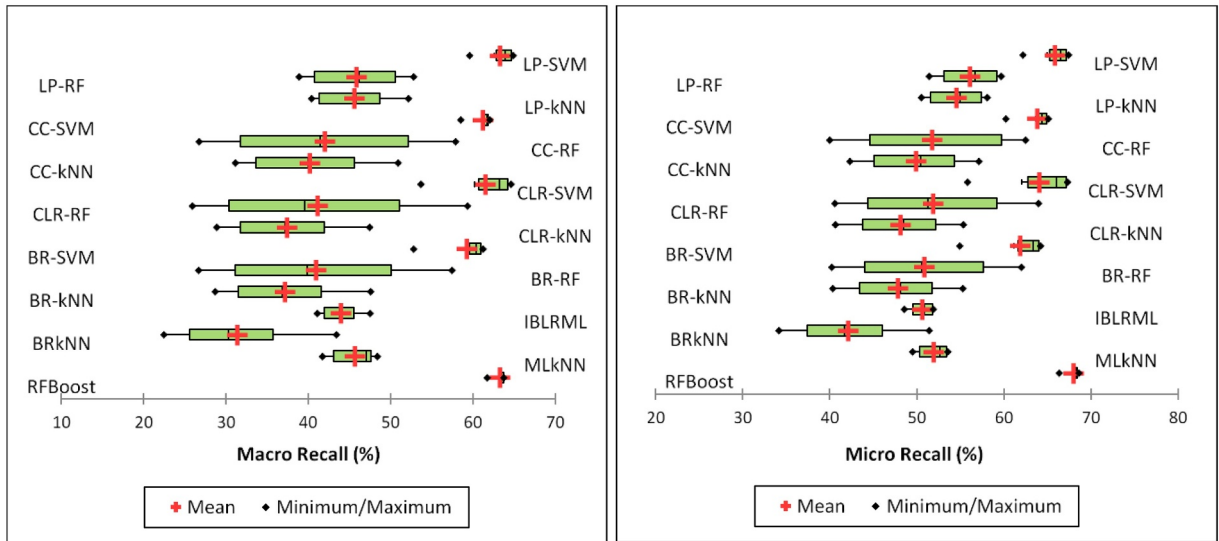


Fig. 6. Boxplots of Macro and Micro averaged Recall scores with varying subsets of training features.

### 5.3.2. Recall measure

Unlike Precision, Recall measures how many texts belong to a category are correctly assigned. The high Precision score means that an algorithm assigns more relevant examples to a category than irrelevant ones (quality), while the high Recall means that an algorithm assigned most of the relevant examples to a category (quantity).

Fig. 6 illustrates the classification performance of the evaluated methods measured by Macro-averaged and Micro-averaged Recall with different numbers of selected features. Among all the base classifier of the transformation-based methods, SVM dramatically outperformed both *k*NN and RF and yielded the best classification performance measured by both Macro and Micro Recall for all methods. *k*NN, the one that obtained better Precision results than SVM, achieved the worst performance in term of Recall. Meanwhile, RF came after SVM and outperformed *k*NN. Among all transformation methods, LP followed by CLR outperformed the other methods BR and CC in general, mainly when SVM is used as a base classifier. For the adaptation-based algorithms, RFBoost dramatically exceeded the performance of the other methods MLkNN, BRkNN, and IBLRML, followed by MLkNN. Meanwhile, BRkNN, which outperformed the other adaptation-based algorithms in terms of Precision measure, obtained the poorest Recall scores.

Comparing the performance of transformation-based and adaptation-based methods, RFBoost, generally, achieved the best performance gauged by Recall (Macro-averaged and Micro-averaged), followed by LP-SVM and then CLR-SVM. However, the best Macro and Micro averaged Recall scores overall were obtained by LP-SVM, which are 0.6486 and 0.6739, respectively, using 4000 features. The second-best overall scores were achieved by RFBoost, which are 0.6374 of Macro-averaged Recall with 500 features, and 0.6857 of Micro-averaged Recall with 2000 features.

### 5.3.3. F1 measure

The trade-off between Precision and Recall when measuring the classification performance makes use of only one of them not enough to evaluate the classification performance. That is, the small number of *true positive* assignments, the number of irrelevant examples that are assigned to a category will increase the precision. The small number of *true positive* assignments comes with a larger number of *false negative* assignments, and this will lead to a low Recall score. For example, RFBoost obtained the lowest Precision scores among all adaptation-based algorithms, and at the same time, it got the highest Recall score.

The F1 measure as to address the limitation of both measures, Recall and Precision, by evaluating the classification performance as the harmonic mean of both Recall and Precision, according to Eq. (6). The F1 measure is more reliable to gauge the performance when compared with the individual measures Recall and Precision.

Fig. 7 illustrates the performance of all evaluated methods measured by both Macro-averaged and Micro-averaged F1. Concerning the transformation-based methods, it is clear that SVM yielded the best performance with all methods, followed by RF. Meanwhile, *k*NN obtained the lowest F1 scores. Among all transformation methods, LP outperformed all other transformation-based methods, BR, CLR, and CC. Regarding the adaptation-based algorithms, RFBoost achieved the best performance and obtained the best Macro and Micro averaged F1 scores, followed by MLkNN and then IBLRML; meanwhile, BRkNN achieved the poorest performance.

RFBoost outperformed all compared methods, in general, followed by LP-SVM. Meanwhile, LP-SVM obtained the highest F1 scores. The best overall score obtained by LP-SVM was 0.6486 and 0.7304 of both Macro-averaged F1 and Micro-averaged F1, respectively, with 4000 features. These scores are slightly higher than the best scores achieved by RFBoost which are 0.6374 of Macro-averaged F1 and 0.7257 of Micro-averaged F1 with 500 and 2000 features, respectively.

For statically analysis, we performed the Friedman test followed by the *Nemenyi post-hoc* test using Macro-averaged F1 and Micro-averaged F1 scores. While it is the harmonic mean of both Recall and Precision, F1 measure is the most suitable measure for

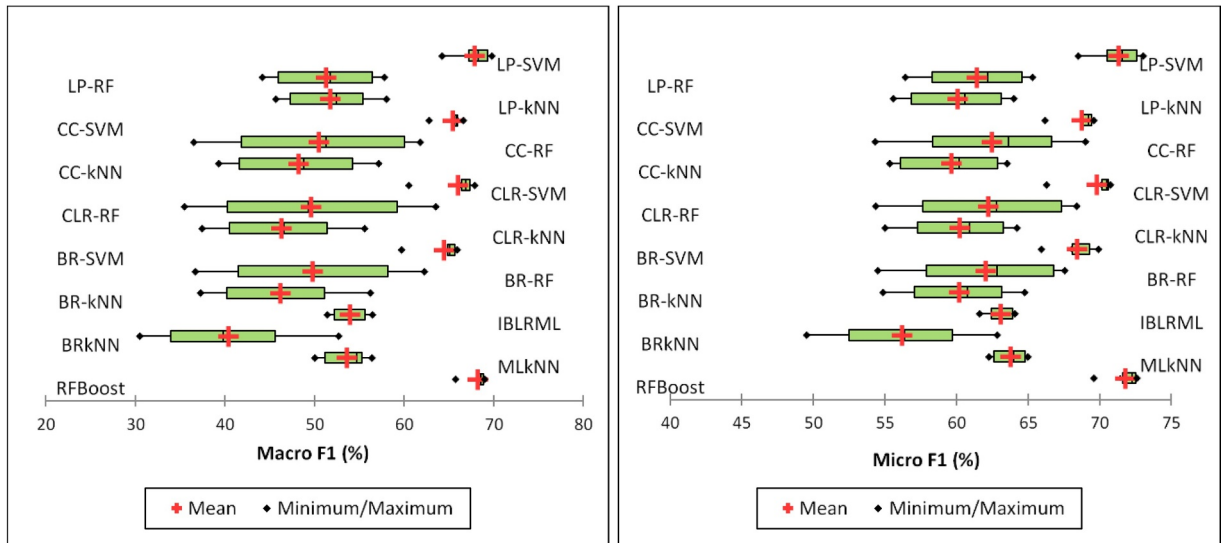


Fig. 7. Boxplots of Macro and Micro averaged F1 scores with varying subsets of training features.

statistically analyzing the performance of the compared methods.

By treating the Macro and Micro averaged F1 scores of the compared methods at different subsets of the selected features (200, 500, 1000, 2000, 3000, and 4000) as independent observations; thus, we have six different observations for each method, which enable us to use a nonparametric pairwise comparison test. Accordingly, the F1 scores for all compared methods at each subset of selected features are firstly ranked, then we perform a Friedman test with a 5% significance level. The estimated  $p$ -value was approximately zero,  $p < 10^{-10}$ , which mean that the null hypothesis that all classifiers achieved the same performance is rejected and the differences between their performance is significant. Having rejected the null hypothesis, the Nemenyi post-hoc test is applied to compare all methods to each other, at 0.05 significance level.

Fig. 8 represents the critical diagrams of the Macro and Micro F1 scores from the Nemenyi post-hoc test of all compared methods. Each critical diagram represents the average ranks, which are obtained from the Nemenyi post-hoc test of the methods, in which the red lines mean that the differences between the methods are less than the critical distance and the methods do not differ significantly. It is clear that RFBoost, followed by LP-SVM are the most accurate classifier. They obtained the highest ranks and significantly outperform most of the other classifiers, for both Macro and Micro averaged F1. All other transformation methods that have SVM as a base learner comes after that, taking the following order: CLR-SVM, CC-SVM, and BR-SVM. Meanwhile, BRkNN and the transformation methods with kNN achieved the poorest performance overall.

Even though it outperforms the other transformation-based methods, LP ignores the multi-label structure of the dataset. That is because LP combines the multiple labels of each training sample as one combined label and used the base learner to build the classification model and classify the test examples. However, some combined labels in the test set are not existing in the training set, which in this case, some samples in the test set will not be assigned to any label. For example, if a document  $d$  in the test set has multiple categories  $c_m$  and  $c_n$ , then LP combines these categories to one category, let's say  $c_{mn}$ . Now, if the combined label  $c_{mn}$  does not occur in the training set, it means there is no document in the training set that has the same multiple categories  $c_m$  and

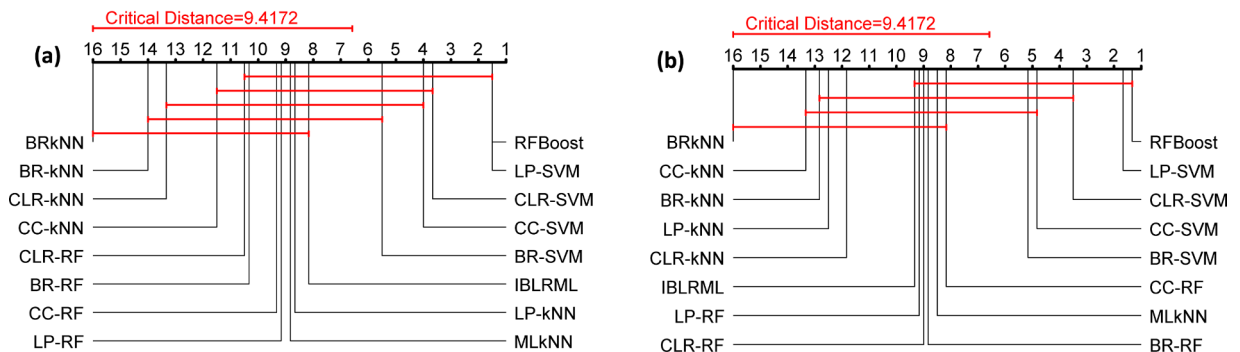


Fig. 8. Critical diagrams from Nemenyi test of (a) Macro-averaged F1 and (b) Micro-averaged F1.

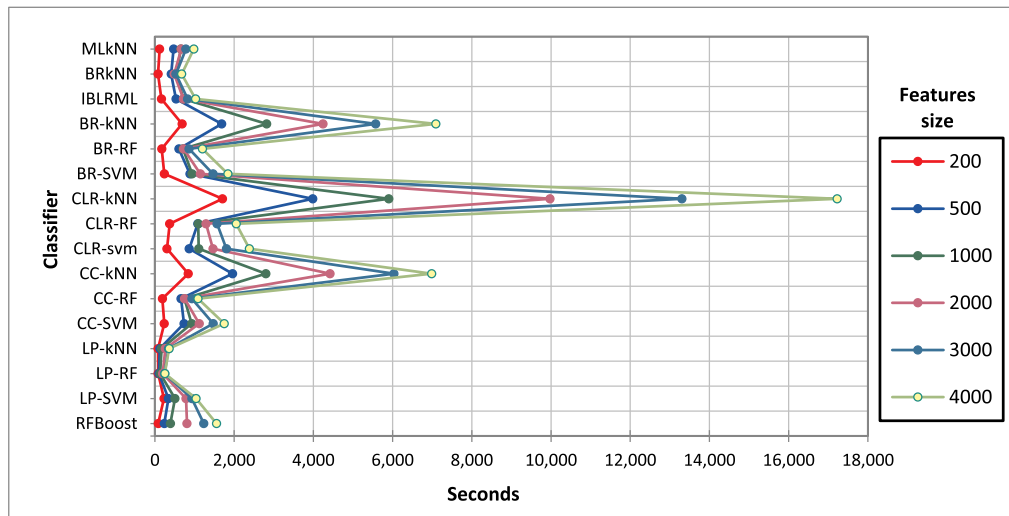


Fig. 9. The computational cost of evaluated methods with different sizes of selected features.

$c_n$ , the document  $d$  will not be classified to any category. This is the reason why LP outperformed the other transformation-based methods.

#### 5.4. Computational cost

Fig. 9 illustrates the computational time (in seconds) of all evaluated methods at different sizes of the selected training features. Noting that the system used for performing all experiments was developed in Java and all experiments were carried out on a PC with Intel Core-i5 processor at 3.00 GHz on 8.00 GB of RAM with Windows 10 64-bit operating system.

LP transformation-based method is the fastest in general. The reason behind this is that LP runs the multiclass base classifier only one time. It treats the multi-label task as a multiclass task, by combining the multiple labels that appear in the training set as one label when inducing the classification model. The disadvantage of this setting is that the dataset loses its natural multi-label structure.

The adaptation-based methods are much faster than the transformation-based method, except for LP. The adaptation-based algorithms, in terms of computation time, take the following order: BRkNN, MLkNN, IBLRML, and RFBoost. For transformation-based methods, the performance time depends on the base learner and the size of the training features. The transformation methods along with their base learner take, based on average time, can be ranked in the following order: LP-RF, LP-kNN, LP-SVM, BR-RF, CC-RF, CC-SVM, BR-SVM, CLR-RF, CLR-SVM, BR-kNN, CC-kNN, and finally CLR-kNN. Among the multi-class base classifiers, RF is the fastest, followed by SVM. Meanwhile, kNN is the slowest. Increasing the number of the training features will dramatically increase the computational time of the methods that use kNN as a base classifier.

## 6. Conclusion

This paper presents a benchmark of multi-label Arabic texts for text categorization. The benchmark, namely “RTAnews” is a collection of annotated Arabic news articles, collected from the Russia Today portal in Arabic. RTAnews consists of 23,837 texts distributed over 40 categories and divided into 15,001 texts for the training and 8,836 texts for the test. RTAnews is an imbalanced dataset, in which some categories have large numbers of texts and other categories have small numbers. The imbalanced structure of RTAnews makes it more reliable to be evaluated for multi-label text categorization, as the texts are generally imbalanced.

The paper also introduces an extensive comparison of the multi-label learning methods for Arabic text categorization using RTAnews. Four problem transformation-based methods (Binary Relevance, Classifier Chains, Calibrated Ranking by Pairwise Comparison, and Label Powerset) and five algorithm adaptation-based methods (Multi-label  $k$ -Nearest Neighbors, Instance-Based Learning by Logistic Regression Multi-label, Binary Relevance kNN, and RFBoost) were evaluated. Three well-known multiclass algorithms (Support Vector Machine,  $k$ -Nearest-Neighbors, and Random Forest) were evaluated as base learners for the transformation-based methods.

The experimental results showed that RFBoost significantly outperforms the other methods. The transformation-based methods perform well when the Support Vector Machine is used as a base classifier. Among the transformation methods, Label Powerset achieves the best and the fastest performance. In the future, we will extend the evaluation to involve other existing multi-label learning algorithms. We will also give more attention to the preprocessing stage, such as employing different stemming algorithms, feature weighting schemes and feature reduction metrics.

## Acknowledgments

This work was supported by Universiti Kebangsaan Malaysia grant Dana Impak Perdana (DIP-2014-039).

## Supplementary materials

Supplementary material associated with this article can be found, in the online version, at [doi:10.1016/j.ipm.2018.09.008](https://doi.org/10.1016/j.ipm.2018.09.008).

## References

- Abdul-Mageed, M. (2017). Modeling Arabic subjectivity and sentiment in lexical space. *Information Processing and Management*. <https://doi.org/10.1016/j.ipm.2017.07.004>.
- Aha, D. W., Kibler, D., & Albert, M. K. (1991). Instance-based learning algorithms. *Machine Learning*, 6(1), 37–66. <https://doi.org/10.1007/bf00153759>.
- Ahmed, N. A., Shehab, M. A., Al-Ayyoub, M., & Hmeidi, I. (2015). Scalable multi-label Arabic text classification. Paper presented at the *Proceedings of the 6th international conference on information and communication systems (ICICS)* <https://doi.org/10.1109/IACS.2015.7103229>.
- Al-Salemi, B., Ab Aziz, M. J., & Noah, S. A. (2015a). Boosting algorithms with topic modeling for multi-label text categorization: A comparative empirical study. *Journal of Information Science*, 41(5), 732–746. <https://doi.org/10.1177/0165551515590079>.
- Al-Salemi, B., Ayob, M., & Noah, S. A. M. (2018). Feature ranking for enhancing boosting-based multi-label text categorization. *Expert Systems with Applications*, 113, 531–543. <https://doi.org/10.1016/j.eswa.2018.07.024>.
- Al-Salemi, B., Aziz, M. J. A., & Noah, S. A. (2015b). LDA-AdaBoost.MH: Accelerated AdaBoost.MH based on latent Dirichlet allocation for text categorization. *Journal of Information Science*, 41(1), 27–40. <https://doi.org/10.1177/0165551514551496>.
- Al-Salemi, B., Noah, M. H. A. S., & Ab Aziz, M. J. (2016). RFBoost: An improved multi-label boosting algorithm and its application to text categorisation. *Knowledge-Based Systems*, 103(Supplement C), 104–117. <https://doi.org/10.1016/j.knosys.2016.03.029>.
- Benbouzid, D., Busa-Fekete, R., Casagrande, N., Collin, F.-D., & Kégl, B. (2012). MultiBoost: A multi-purpose boosting package. *Journal of Machine Learning Research*, 13(Mar), 549–553.
- Boutell, M. R., Luo, J., Shen, X., & Brown, C. M. (2004). Learning multi-label scene classification. *Pattern Recognition*, 37(9), 1757–1771. <https://doi.org/10.1016/j.patcog.2004.03.009>.
- Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2), 123–140. <https://doi.org/10.1007/bf00058655>.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/a:1010933404324>.
- Cheng, W., & Hüllermeier, E. (2009). Combining instance-based learning and logistic regression for multilabel classification. *Machine Learning*, 76(2–3), 211–225. [https://doi.org/10.1007/978-3-642-04180-8\\_6](https://doi.org/10.1007/978-3-642-04180-8_6).
- Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7, 1–30.
- Dobbin, K. K., & Simon, R. M. (2011). Optimally splitting cases for training and testing high dimensional classifiers. *BMC Medical Genomics*, 4(1), 31. <https://doi.org/10.1186/1755-8794-4-31>.
- Eldos, T. (2003). Arabic text data mining: A root-based hierarchical indexing model. *International Journal of Modelling and Simulation*, 23(3), 158–166. <https://doi.org/10.1080/02286203.2003.11442267>.
- Elghazel, H., Aussem, A., Gharroudi, O., & Saadaoui, W. (2016). Ensemble multi-label text categorization based on rotation forest and latent semantic indexing. *Expert Systems with Applications*, 57, 1–11. <https://doi.org/10.1016/j.eswa.2016.03.041>.
- Esuli, A., Fagni, T., & Sebastiani, F. (2006). MP-Boost: A multiple-pivot boosting algorithm and its application to text categorization. Paper presented at the *Proceedings of the string processing and information retrieval* [https://doi.org/10.1007/11880561\\_1](https://doi.org/10.1007/11880561_1).
- Flores, F. N., & Moreira, V. P. (2016). Assessing the impact of stemming accuracy on information retrieval – a multilingual perspective. *Information Processing and Management*, 52(5), 840–854. <https://doi.org/10.1016/j.ipm.2016.03.004>.
- Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of machine learning research*, 3, 1289–1305 Mar.
- Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1), 119–139. <https://doi.org/10.1006/jcss.1997.1504>.
- Fürnkranz, J., Hüllermeier, E., Mencía, E. L., & Brinker, K. (2008). Multilabel classification via calibrated label ranking. *Machine Learning*, 73(2), 133–153. <https://doi.org/10.1007/s10994-008-5064-8>.
- Gantz, J., & Reinsel, D. (2012). The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east. *IDC iView IDC Analyze the Future*, 2007(2012), 1–16.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter*, 11(1), 10–18.
- Hmeidi, I., Al-Ayyoub, M., Mahyoub, N. A., & Shehab, M. A. (2016). A lexicon based approach for classifying Arabic multi-labeled text. *International Journal of Web Information Systems*, 12(4), 504–532. <https://doi.org/10.1108/ijwis-01-2016-0002>.
- Hüllermeier, E., Fürnkranz, J., Cheng, W., & Brinker, K. (2008). Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172(16), 1897–1916. <https://doi.org/10.1016/j.artint.2008.08.002>.
- Jaccard, P. (1908). Nouvelles recherches sur la distribution florale. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 44, 223–270.
- Joachims, T. (1998). *Making large-scale SVM learning practical* Universität Dortmund Technical Report, SFB 475: Komplexitätsreduktion in Multivariaten Datenstrukturen.
- Karisan, P., Rahgozar, M., & Oroumchian, F. (2016). A query term re-weighting approach using document similarity. *Information Processing and Management*, 52(3), 478–489. <https://doi.org/10.1016/j.ipm.2015.09.002>.
- Loza Mencía, E., Park, S.-H., & Fürnkranz, J. (2010). Efficient voting prediction for pairwise multilabel classification. *Neurocomputing*, 73(7), 1164–1176. <https://doi.org/10.1016/j.neucom.2009.11.024>.
- Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., & Euler, T. (2006). Yale: Rapid prototyping for complex data mining tasks. Paper presented at the *Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining* <https://doi.org/10.1145/1150402.1150531>.
- Mubarak, H., & Darwish, K. (2014). Using Twitter to collect a multi-dialectal corpus of Arabic. Paper presented at the *Proceedings of the EMNLP workshop on Arabic natural language processing (ANLP)* <https://doi.org/10.3115/v1/w14-3601>.
- Platt, J. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines. In Microsoft Research Technical Report.
- Read, J. (2008). A pruned problem transformation method for multi-label classification. Paper presented at the *Proceedings of the New Zealand computer science research student conference (NZCSRS 2008)*.
- Read, J., Pfahringer, B., Holmes, G., & Frank, E. (2011). Classifier chains for multi-label classification. *Machine Learning*, 85(3), 333–359. <https://doi.org/10.1007/s10994-011-5256-5>.
- Read, J., Reutemann, P., Pfahringer, B., & Holmes, G. (2016). Meka: A multi-label/multi-target extension to weka. *The Journal of Machine Learning Research*, 17(1), 667–671.
- Rehman, A., Javed, K., & Babri, H. A. (2017). Feature selection based on a normalized difference measure for text classification. *Information Processing and Management*, 53(2), 473–489. <https://doi.org/10.1016/j.ipm.2016.12.004>.
- Rijsbergen, C. J. V. (1979). *Information retrieval*. Butterworth-Heinemann.

- Romeo, S., Da San Martino, G., Belinkov, Y., Barrón-Cedeño, A., Eldesouki, M., Darwish, K., et al. (2017). Language processing and learning models for community question answering in Arabic. *Information Processing and Management*. <https://doi.org/10.1016/j.ipm.2017.07.003>.
- Schapire, R. E., & Singer, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3), 297–336. <https://doi.org/10.1023/A:1007614523901>.
- Schapire, R. E., & Singer, Y. (2000). BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39(2), 135–168. <https://doi.org/10.1023/A:1007649029923>.
- Shehab, M. A., Badarneh, O., Al-Ayyoub, M., & Jararweh, Y. (2016). A supervised approach for multi-label classification of Arabic news articles. Paper presented at the 7th international conference on computer science and information technology (CSIT) <https://doi.org/10.1109/csit.2016.7549465>.
- Spyromitros, E., Tsoumakas, G., & Vlahavas, I. (2008). An empirical study of lazy multilabel classification algorithms. *Artificial intelligence: Theories, models and applications*. Springer 401–406. [https://doi.org/10.1007/978-3-540-87881-0\\_40](https://doi.org/10.1007/978-3-540-87881-0_40).
- Taha, A. Y., & Tiun, S. (2016). Binary relevance (br) method classifier of multi-label classification for Arabic text. *Journal of Theoretical and Applied Information Technology*, 84(3), 414.
- Tang, B., Kay, S., & He, H. (2016). Toward optimal feature selection in naive Bayes for text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 28(9), 2508–2521. <https://doi.org/10.1109/tkde.2016.2563436>.
- Tong, S., & Koller, D. (2001). Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2, 45–66 Nov.
- Tsoumakas, G., & Katakis, I. (2007). Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3), 1–13.
- Tsoumakas, G., Katakis, I., & Vlahavas, I. (2010). Mining multi-label data data mining and knowledge discovery handbook. Springer 667–685. [https://doi.org/10.1007/978-0-387-09823-4\\_34](https://doi.org/10.1007/978-0-387-09823-4_34).
- Tsoumakas, G., Spyromitros-Xioulis, E., Vilcek, J., & Vlahavas, I. (2011). Mulan: A java library for multi-label learning. *Journal of Machine Learning Research*, 12, 2411–2414 Jul.
- Tsoumakas, G., & Vlahavas, I. (2007). Random k-labelsets: An ensemble method for multilabel classification. machine learning *Proceedings of the ECML* (pp. 406–417). Springer. [https://doi.org/10.1007/978-3-540-74958-5\\_38](https://doi.org/10.1007/978-3-540-74958-5_38).
- Vapnik, V. (2013). *The nature of statistical learning theory*. Springer Science & Business Media.
- Wu, H., Gu, X., & Gu, Y. (2017). Balancing between over-weighting and under-weighting in supervised term weighting. *Information Processing and Management*, 53(2), 547–557. <https://doi.org/10.1016/j.ipm.2016.10.003>.
- Wu, Q., Ye, Y., Zhang, H., Ng, M. K., & Ho, S.-S. (2014). ForesTexter: An efficient random forest algorithm for imbalanced text categorization. *Knowledge Based Systems*, 67, 105–116.
- Zhang, M.-L., & Zhou, Z.-H. (2007). ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7), 2038–2048.