Arabic Text Categorization Based on Arabic Wikipedia

ADNAN YAHYA, Birzeit University
ALI SALHI, Birzeit University

This paper describes an algorithm for categorizing Arabic text, relying on highly categorized corpus-based data sets, obtained from the Arabic Wikipedia by using manual and automated processes to build and customize categories. The categorization algorithm was built by adopting a simple categorization idea, then moving forward to more complex one. We applied tests and filtration criteria to end with the best and most efficient results that our algorithm can achieve. The categorization depends on the statistical relation between the input text and the reference (training) data supported by well defined Wikipedia-based categories. Our algorithm supports two levels for categorizing Arabic text; categories are grouped into a hierarchy of main categories and subcategories. This introduces a challenge due to the correlation between certain subcategories and overlap between main categories. We argue that our algorithm achieved good performance compared to other methods reported in the literature.

Categories and Subject Descriptors: Arabic Language Processing

General Terms: Arabic Natural Language Processing, Arabic Wikipedia, Categorized Corpora, Text Categorization, Light Stemming, Text Analysis

1. INTRODUCTION

Over the world wide web, the continuous increase in content creation in general, and Arabic content in particular, comes with a great need for tools to overcome the many challenges facing the processing and retrieval of web content. For Arabic content these challenges include understating the content, efficient retrieval of useful information from this content, improving the quality and efficiency of searching Arabic data by providing tools such as spelling, correction, named entity extraction, document categorization, query optimization and filtration and more.

The process of categorizing (or classifying) documents by assigning one of a set of given categories to the document is an important challenge when it comes to Arabic. Being able to search pre-categorized documents helps improving search results due to the level of ambiguity in everyday Arabic text. For example if a user searched for the word ريال (Riyal) he might be looking for ريال مدريد (Real Madrid) team or ريال سعودي (Saudi Riyal) currency. If the search is done based on categorized documents then the user will have the option to search either under رياضة (Sports) or under علوم مالية ومصرفية(Finance). Moreover, text categorization can be used to define better spell-checking systems, as in our earlier work where we used categorization and categorized datasets to narrow the possibilities of outputs in a spell checker by first categorizing the input document then use a categorized dataset (dictionary) as the reference for the correction process. This improved the results and performance [Yahya and Salhi 2012]. Also categorization may help improve question answering tasks by resolving text ambiguities by reference to categories (As we said before ريال "Riyal" in economics is different from ريال "Riyal" in sports, تضخم as "inflation" in the economic context or "inflammation" in the medical context, …).

In this paper we are offering an algorithm for categorizing Arabic text, built by adopting a simple categorization idea, then moving forward to more complex one. The algorithm relies on a highly categorized corpus-based data sets obtained from the Arabic Wikipedia by using a combination of manual and automated processes to build and customize categories. Our algorithm is hierarchical and supports two levels of categorization of Arabic text. That is, our categories are grouped into a hierarchy of main categories and subcategories. This introduces challenges resulting from the correlation between certain subcategories and the overlaps between main categories, something we will also discuss in this paper.

The rest of the paper is organized as follows: in Section 2 will discuss some earlier and related work. In Section 3 we will discuss our corpora and how they were built. In Section 4 we will discuss the text filtration methods that we applied in some of our tests. Discussing our categorization process will be in Section 5, where we talk in detail about our algorithm and the different approaches and filters we tested. Comparisons with related work will be discussed in Section 6, followed by conclusions and pointers to future work in Section 7.

2. RELATED WORK

In this section we will summarize and discuss some earlier related work and then we will select some resources from early work for comparison tests. The comparison is done by comparing the results reported by others with the results of using their resources in training and testing our algorithm.

Work on Arabic text classification used several approaches with different data resources. Some use Manhattan and Dice measures on N-Gram sets extracted from a corpus of text documents covering four categories: *Sports, Economy, Technology and Weather* [Khreisat 2006]. The corpus there was collected from different online Arabic newspapers and was split into 60% training data and 40% testing data. The accuracy value (F1 score) for Manhattan measure was 60.7% and for Dice measure was 85.6%.

[Al-Harbi et al. 2008] used Chi-Squared statistics to select the best N (where N=30) terms to represent a certain category. They built their experiments using several sources of Arabic text obtained from different news agencies and websites. The data was split into 70% training and 30% testing and two classification algorithms were applied; the Support Vector Machines (SVM) algorithm and Clementine for the C5.0 decision tree algorithm with average success rates of 68.65% for SVM and 78.42% for C5.0.

Another study used Naïve Bayes algorithm and reported 68.78% accuracy [El-Kourdi et al. 2004]. Another study used kNN algorithm and reported 96% accuracy results based on six categories: *Politics, Economics, Health, Sports, Cancer and Agriculture* [Al-Shalabi et al. 2006].

[Syiam et al. 2006] built an intelligent system for Arabic text categorization by adopting machine learning algorithms, different stemming algorithms and feature selection and term weighting methods with kNN and Rocchio classifiers. The tests were done over 6 categories (*Arts, Economics, Politics, Sports, Woman and Information Technology*) and concluded that Rocchio classifier has an advantage over kNN classifier with accuracy of 98%.

[Alsaleem 2010] investigated Naïve Bayesian method (NB) and SVM algorithms on Arabic corpora of seven categories extracted from The Saudi Newspapers (SNP). His experiments reveal that SVM algorithm outperforms the NB, which agrees with the results in [Saad 2011].

[Saad 2011] studied the impact of text pre-processing on classification by analyzing input text, changing term weighting schemes and Arabic morphological analysis (stemming and light stemming) and using approaches such as *Decision Tree, k-Nearest Neighbors, Support Vector Machines, Naïve Bayes and its variations* to classify the input text. The researcher applied the classification algorithms to seven different corpora (splitting each corpus to training and testing texts). He concludes that light stemming with Support Vector Machines outperforms other algorithms.

[Yang et al. 2003] studied well-known categorization algorithms such as *Support Vector Machines, k-Nearest Neighbor, Ridge Regression, Linear Least Square Fit and Logistic Regression* by an investigation on the usage of those algorithms in a hierarchical setting for categorization. They proved that the scalability of a method depends on the topology of the hierarchy and the category distributions, in other words the distribution of categories and subcategories affects the categorization process which we will see in Section 5.4.

[Qiu et al. 2011] highlights three approaches for hierarchical text classification; *flat, local and global approaches*. The *flat approach* uses only the classes of end categories in the leaf nodes (categorize by subcategories only) and works without hierarchical class information. The *local approach* is based on a top-down fashion, which starts by categorizing the text into main categories on the top-level then re-categorizes the subcategories (low-level) under the main category (top-level). The *global approach* builds only one categorizer to discriminate all categories in a hierarchy. In our work we first will adopt a *local approach* and prove later in Section 5.4 that using a *flat approach* gives better results.

As noted, results vary from one experiment to another because of the data, algorithms, and measures used. In most of the work we reviewed authors use few distinct categories in their experiments and do not address the challenge of having a large set of categories or of having a set of highly correlated categories.

It is not easy to compare our results with others. However after checking the earlier work discussed above, we considered the work of [Saad 2011] due to the nature of the results obtained and availability of the used corpora. [Saad 2011] is not a single approach to categorization but rather an application of several categorization algorithms with the use of different corpora. So we thought the best way to compare our work with others is simply by comparing our work with this author's work since he already compared well known algorithms and highlighted the best of them. Thus using his corpora with our algorithm is just like comparing our work with the algorithms he tested. Considering the corpora mentioned in [Saad 2011] and our predefined corpora, we did some tests to compare his results with ours. That is, we applied his corpora (most of it is available online) to our categorizing algorithm and compared our results with the results he obtained. This will be discussed in Section 6.

3. CATEGORIZATION CORPORA

In this section we will talk about our training and testing data for the developed categorizing algorithms.

3.1 Training data

In this work we focus on building different categorized data sets of words. The idea is to provide a wide range of categories, forming a hierarchy where some are subcategories of others, and use them in building and testing different categorizing approaches.

We built our corpora using the Arabic Wikipedia, by applying our own dynamic category extraction algorithm which will be discussed next.

Wikipedia-based categories were built using a partial copy of the Arabic Wikipedia that holds around 96,128 titles with their content. The copy is not directly obtained

from Wikipedia dumps. Rather, it was obtained from "*The Arabic online content indications project*".[1]

In the Wikipedia each article is associated with a set of manual tags. The overlapping tags are not well defined. That is, one can find tags such as: رؤساء و وزراء فلسطين , وزراء فلسطين رؤساء السلطة الوطنية (In English: Presidents and Ministers of Palestine, Ministers of Palestine, Presidents of the Palestinian Authority). These tags can be merged in one major tag such as قادة فلسطينيين (Palestinian Leaders) or in the more general tag أخبار فلسطين (Palestine News) or the general tag سياسة (Politics). The tags found in Wikipedia may be too specific on one hand and on another can be repeated using different words.

To build the Wikipedia-based categories we need first to define the categories and then add as many as possible articles under each category. To do that, an automated process of connecting related articles based on manual tags was built, followed by a manual verification process.

Using Wikipedia manual tags we can link articles based on the shared tags between the articles; the more shared tags the more the articles are related. Also this means that there is a possible relation between tags if the tags appear jointly in different articles. For example if text A (in the Wikipedia) is tagged under: قوانين نيوتن (Newton Laws) and ميكانيكا (Mechanics) and text B is tagged under: ميكانيكا (Mechanics) and طاقة حركية (Kinetic Energy) then we can conclude that these three tags are related as they have ميكانيكا (Mechanics) in common. However, if we go deep in this relation analysis we may end up connecting all tags in the Wikipedia (which is not desirable). So one should be wise in selecting the limit of relation depth and interfere manually to have control over how deep the tags/articles relation goes. For that we developed the *Related Tags Approach*.

The approach can be illustrated by the following steps:

(a) We start by defining the category we want to build (starting point), say فيزياء (Physics). Now the goal is to collect Wikipedia articles that talk about topics in فيزياء (Physics).

(b) We parse our list of Arabic Wikipedia articles to extract the articles that contain فيزياء (Physics) as a tag.

(c) For each extracted article, the tags found while parsing the article are added to a queue (Q), for example if an extracted article (that already includes فيزياء " Physics" tag) has also ميكانيكا (Mechanics) and طاقة حركية (Kinetic Energy) tags, then both of the tags will be added to the queue with a variable (frequency) that indicates how many times a certain tag is seen. In the current example, since both tags are seen for the first time the frequency of each will be 1. This variable also indicates the number of articles that contain a certain tag.

(d) After repeating step (c) for all articles, we move to the next tag in the queue. In our example it's ميكانيكا (Mechanics).

[1] The Arabic online content indications project. 2010. Computer Research Institute. King Abdul-Aziz City for Science and technology. Retrieved from http://cri.kacst.edu.sa/en/cri-products/current-projects

(e) The process repeats itself here, step (b) and (c) and each repeated tag will only increment its frequency variable.
(f) When the total number of Wikipedia articles processed reach N, the process stops. (We set N to 50)[2]
(g) The queue now will hold a set of tags and their frequencies., The tags are sorted based on their frequency and then a manual process to detect the best tags to use is adopted. The manual process is to make sure that the tags in the queue (which is added due to the parsing of the N = 50 articles) are truly related and do not cause major problems in categorization.
(h) The selected final set of tags will be used to parse all the articles containing any of those tags related to فيزياء to get the categorized corpus for فيزياء (Physics).

Table I shows some categories with their top 10 related tags.

Table I. Categories with top 10 related tags, for the related tags approach

Selected tags related to: **فيزياء (Physics)** with English translation			
نظريات فيزيائية	Physics Theories	علم الكون	Cosmology
فيزياء	Physics	أعداد الكم	Quantum numbers
نسبية	Relativity	ضوء	Light
صواريخ	Rockets	أمثلة كونية	Examples of Cosmic
إلكترونيات	Electronics	اتصالات	Communication
Selected tags related to: **طب (Medicine)** with English translation			
طب	Medicine	صحة	Health
علم الأدوية	Drug Science	منظمة الصحة العالمية	World Health Org.
أمراض	Diseases	أمراض وراثية	Hereditary diseases
مصطلحات طبية	Medical Terms	مضادات حيوية	Antibiotics
صيدلة	Pharmacy	علم الوراثة	Genetics
Selected tags related to: **علم حاسوب (Computer Science)** with English translation			
علم الحاسوب	Computer Science	حوسبة	Computing
شبكات عصبونية	Neural Networks	برمجة	Programming
تنقيب البيانات	Data Mining	إنترنت	Internet
تحليل رياضي	Mathematical Analysis	معلوماتية عصبونية	Neural Information
أمن المعلومات	Information Security	أمن شبكة الحاسوب	Network Security
Selected tags related to: **دين (Religion)** with English translation			
دين	Religion	فقه إسلامي	Islamic jurisprudence
فقه عبادات	Jurisprudence of Worship	ديانات آسيا	Religion in Asia
أركان الإسلام	Pillars of Islam	جهاد	Jihad
شريعة إسلامية	Islamic law	طوائف إسلامية	Islamic sects
نصوص دينية	Religious texts	كونفوشيوسية	Confucianism

Using this technique, we built a Wikipedia-based categorized corpus. Table II gives statistics about the categories we adopted; of course the data in this corpus is subject to change due to continuous data processing. So far, we defined 25 categories. In Table II, *"# of distinct words*" represents the total number of distinct words in each

[2] The value of N determines how much manual work will be needed at step (g). Increasing N will increase the tags and thus the need for manual check for the added tags. The appearance possibility of unrelated tags in the queue will become higher when increasing N. To maintain control over the quality and value of the corpus and to limit the manual check and to output a reasonable size categorized corpus, we set N to 50. We believe the resulting corpora were of good quality. However, this value can be subject to more experiments.

corpus. Words are extracted from the text by splitting on "white space" characters. Please note that some earlier and related work refer to word(s) as "*term(s)"* [Sarkar et al. 2004; Goweder and Reock 1998].[3]

Using the *Related Tags Approach* we can create new categories by starting from the desired category. For example if a user is interested in the category الطيور (birds) then adopting the same steps as for فيزياء (Physics) will output a new corpus specialized in الطيور (Birds) from the Wikipedia articles.

Table II. Wikipedia categories current statistics.

Category (Arabic)	Category (English)	Total number of words	#of distinct words	Average Frequency*
كرة قدم	Football	47,704	5,046	9.45
كرة سلة	Basketball	36,290	3,479	10.43
تنس	Tennis	53,400	4,426	12.07
سباقات سيارات	Racing	18,498	2,385	7.76
أولمبياد	Olympics	50,193	4,630	10.84
اقتصاد	Economics	52,069	6,894	7.55
إسلامي	Islam	144,484	17,176	8.41
مسيحي	Christianity	36,971	5,628	6.57
كهربائية وإلكترونية	Electronics	23,733	3,419	6.94
ميكانيكية	Mechanics	23,815	3,799	6.27
كمبيوتر و شبكات	Computers	55,686	7,303	7.63
كيمياء	Chemistry	37,703	5,196	7.26
فيزياء	Physics	13,765	2,342	5.88
رياضيات	Mathematics	22,745	3,337	6.82
أحياء	Biology	19,412	3,683	5.27
طب بشري	Medicine	46,596	6,262	7.44
صيدلة	Pharmacy	9,253	2,003	4.62
تاريخ حديث	New History	50,959	6,409	7.95
تاريخ قديم	Old History	63,413	8,800	7.21
شعر وأدب	Literature	21,533	4,559	4.72
موسيقى وغناء	Music	27,740	4,931	5.63
سينما ومسرح	Cinema & Theater	30,795	5,694	5.41
ديانات أخرى	Other Religions	17,699	3,444	5.14
سياسة	Politics	93,945	11,409	8.23
موضة	Fashion	9,574	2,212	4.32

* Total_number _of_words/#_distinct_words

3.2 Testing data

Before we discuss the categorization process, let us introduce the testing sample that will be used in testing our algorithm. The testing was done on a sample of 400 documents distributed among 10 categories with 40 documents in each category. Table III shows the categories used in the tests, and their sources (web sites). The documents were pre-categorized manually by human experts as in Table III, thus when testing our categorizing algorithms we compare the output category for each test document with the original category of the test document set by human experts. To calculate the success rates which will be reported for our experiments.

Our testing samples were not derived from the same source as the training data. That is, the testing source is not the Wikipedia: rather, it was collected from random Web Pages. We believe that it makes more sense to have training and testing data

[3] Also it is worth mentioning that it's our intention to make our categorized corpus available to the research community with different features and better characterization. For more information readers and researchers are advised to contact the authors.

come from different sources in order for the tests to be more credible and indicative of performance in real life environments. We needed a way to make sure that our testing data sources don't cross directly with Wikipedia articles used in building the training data. To do that, we applied a simple test. We used Google search engine to search the Arabic Wikipedia for pages that include (1) the domains of the test documents sources and (2) the category of each set of test documents. For example in Table III we have a category كرة قدم (Football) with "mbc.net" as one source of testing documents for كرة قدم (Football), we applied the query shown in Figure 1.

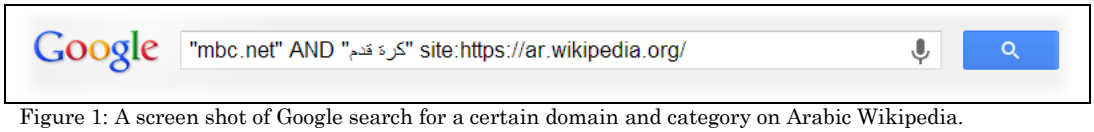

Figure 1: A screen shot of Google search for a certain domain and category on Arabic Wikipedia.

This was done for all the websites for each category in the testing documents and none of the testing documents gave a match with any Wikipedia page included in our training data.

Table III. Testing data sources.

Category	# of documents	Web Sites
كرة قدم (Football)	40	http://www.mbc.net http://www.yallakora.com/ http://www.kooora.com/ http://www.syrian-soccer.com
كرة سلة (Basketball)	40	http://www.as7apcool.com http://www.kooora.com/ http://www.yallakora.com/
سباقات (Racing)	40	http://www.bbc.co.uk http://www.yallakora.com/
هندسة كهربائية والكترونية (Electronics)	40	http://olom.info/ http://www.alhandasa.net http://aafaq.4t.com/components.htm
فيزياء (Physics)	40	http://www.physicsacademy.org/ http://hazemsakeek.com/ http://phys.olom.info/ http://www.schoolarabia.net http://www.marefa.org
كيمياء (Chemistry)	40	http://www.ksa-teachers.com/ http://www.schoolarabia.net http://www.bytocom.com/
أحياء (Biology)	40	http://www.sehha.com http://www.asnanak.net http://www.csmc.edu/6757.html
كومبيوتر وشبكات (Computers)	40	http://www.bramjnet.com http://www.boosla.com/
اقتصاد (Economics)	40	http://www.aljazeera.net http://www.bbc.co.uk/arabic/ http://ara.reuters.com/news/business
سياسة (Politics)	40	http://www.maannews.net http://www.aljazeera.net

4. PREPROCESSING TECHNIQUES

In this section we will discuss some filtration tools used in some of our testing. The filtration tools are not a basic part of the categorization process. However we test their effect on the performance of the categorization process.

4.1 Root Extraction (RE)

Root extraction is based on our earlier work [Yahya and Salhi 2011] and the idea is to process the training and testing data to extract word roots and use roots rather than words in the categorizing algorithm.

Our root extraction filter, recursively removes prefixes, suffixes and infixes then attempts to find a root for the stripped form. Affix removal is based on input word length to judge what, which and when to remove a certain affix. For more information about the extraction process, see [Yahya and Salhi 2011].

4.2 Light Stemming (LS)

Similar to root extraction, however we only filter prefixes and suffixes, and keep infixes, if present. One of the best light stemmers is called light10 [Larkey et al. 2007]. However we used our own light stemmer which employs nouns/verbs/adjectives lists obtained from [Attia 2011].

Our approach extracts three lists of reference stems (nouns, verbs, adjectives) from [Attia 2011] datasets. Each list is related to a map of prefix/suffix strings. When the stemmer receives an input word, it removes prefix and suffix by applying the three maps (one at a time), then normalize letters such as {أ ا إ آ} to {ا} and {ة,ه} to {ه}. Then the result will be three suggested stems for each word, then the suggested stems are compared with the lists of nouns, verbs and adjectives. If a match is found, it will be considered the stem of the word, if more than one stem is found, then the first match will only be considered, if no stem in the reference lists matches the suggested stems, then the largest suggested stem with less prefix/suffix removal is considered to be the stem of the word. If the input word is a stop word it will be returned as is.

Table IV shows a sample comparison result between our approach and light10 in terms of the stems returned for an input word. Later in this paper (Section 5.7) and based on experimental results of categorization we will prove that the use of our light stemmer gives better results than the use of Light10.

Table IV. Some Results for comparison between our stemmer and light 10.

Word	Our Approach	Light 10
هاجمت	هاجم	هاجمت
الرئاسة	رئاسه	رئاس
الفلسطينية	فلسطيني	فلسطين
بشده	شده	بشد
وشرطة	شرطه	شرط
الحكومة	حكومه	حكوم
غزة	غزة	غز
باتت	بات	باتت
أزمة	ازمه	ازم
لنادي	نادي	لناد

4.3 Special Expressions Extraction

The tool starts by extracting lists of single, double and triple expressions from the input list. Then each expression in each list is checked by a stop words filter to remove any expression with a stop word, then the expressions in each list are checked again to select expressions that start with (1) the definite article Al Ta'reef "ال" (in all words), and (2) the expressions with nouns or adjectives (using the lists in [Attia 2011]). Expressions with verbs are dropped.

The final filtered expressions are ordered by frequency of appearance and considered to be the representation of the input text.

Please note that we are not building or using any NER system here. We are only filtering single, double, and triple expressions based on the simple filters mentioned above.

5. CATEGORIZATION PROCESS

Our categorization process is based on the idea of categorizing the input text in two phases. In phase one, we categorize the text into one of the main categories, and in phase two, we further categorize the input text based on subcategories. For example if a text was assigned رياضة (Sports) in phase one, in phase two it will be further categorized into one of رياضة (Sports) categories (football, basketball, tennis ... etc). Table V shows the adopted main categories and their subcategories.

Table V. Main and Subcategories.

#	Main Categories	Subcategories
1	رياضة Sports	كرة قدم, كرة سلة, كرة مضرب, سباقات, العاب أولمبية Football, Basketball, Racing, Tennis, Olympics
2	علوم Sience	فيزياء, كيمياء, أحياء, رياضيات Physics, Biology, Mathematics, Chemistry
3	هندسة Engineering	هندسة ميكانيكية, هندسة كهربائية, حاسوب Electronics, Mechanics, Computers
4	صحة Health	طب عام, صيدلة Pharmacy, Medicine
5	أدبيات Literatures	موسيقى, سينما, أدب, موضى Music, Cinema & Theatre, Fashion, Literature
6	تاريخ History	تاريخ حديث, تاريخ قديم New History, Old History
7	دين Religion	الديانة الإسلامية, الديانة المسيحية, ديانات أخرى Islam, Christianity, Other Religions
8	اقتصاد Economic	-----
9	سياسة Politics	-----

We started our categorizing process by adopting a simple categorization idea BCA, and then moved to more complex one. During that we applied a testing sample to make sure that the results are improving by the modifications we adopt. We start by introducing our basic categorization approach, then introducing the more complex one which we named *Percentage and Difference Categorization (PDC)* Algorithm.

5.1 Basic Categorization Algorithm (BCA)

The basic categorization algorithm assigns the input text represented by vector Z to a category by calculating a weight for Z in each reference category X: $W_X(Z)$. The category with the highest weight is considered to be the correct category for the input text. The weighting function $W_X(Z)$ is based on Equation (1), where z_i are words in Z, m is the number of distinct words in Z found also in X, n is the number of words (x_i) in X and $\omega(t)$ is the frequency of word t. $W_X(Z)$ is defined by the total number of words found in both the input text Z and category X to the total number of category words normalized by the relative sizes of Z and X.

$$W_X(Z) = \left[\frac{\sum_{f_j \neq 0} \omega(z_j)}{\sum_{i=1}^{n} \omega(x_i)} \right] * \left[\frac{m}{n} \right] \quad (1)$$

To understand "Eq(1)" take the following example: assume that an input text Z has words $\{z_1, z_2, z_3, z_4, z_5\}$ and $\{z_1, z_2, z_5\}$ are also found with (nonzero) frequencies f_1, f_2 *and* f_5, respectively in reference category X with n words in the form of $\{x_1, x_2, x_3, \ldots, x_n\}$, m is equal 3 (number of found words) and j will range over $\{1,2,5\}$. This means that "Eq(1)" will generate Equation (1').

$$W_X(Z) = \left[\frac{f_1 + f_2 + f_5}{\sum_{i=1}^{n} \omega(x_i)} \right] * \left[\frac{3}{n} \right] \quad (1')$$

The Category with the highest weight is considered to be the correct category for the input text Z. Of course the categorization process is done after removing stop words from the input text, using stop words filtration method discussed in [Yahya and Salhi 2011].

5.2 Percentage and Difference Categorization (PDC) Algorithm

This algorithm focuses on the relation between ratios in the input text words and the corresponding ratios in the reference texts (our training data) to decide to which category to assign each word in the input text. This means it will calculate the percentage of each word (word frequency/total words) in the input and compare it with that word percentage in each category (if it exists), then find the difference between the two values and assign to the word the category with smallest difference.

The difference will give us an idea of how much a word z in the input Z is close to the frequency percentage for z in category X. We can say (in general) that this algorithm works as if it is deciding for each word of the input text how closely it is related to each of the given categories.

For example if word z has frequency 7 in the 300 word input text Z, then the percentage of z in the input table is $7/300 = 0.023333$. Next z ratio is calculated in each category in the reference data (if it exists), for example z has a frequency of 500 in category X_1 with $10{,}000$ *words*, then z in X_1 has the ratio $500/10000 = 0.05$. Then the relation between z in X_1 and z in the input text Z will be the absolute value of $(0.023333 - 0.05)$ which is 0.026667, this is done for all categories $(X_1, X_2, \ldots, X_m)$ and the category with minimum difference is assigned to the word z (not the input text).

The word z flag of category X is set to 1, where category X gives the minimal distance between the frequencies of z in the input text and its counterparts in all categories.

This process is repeated for all words of Z, after removing stop words from input text using stop words filtration method in [Yahya and Salhi 2011]. Basically we are categorizing each nonstop word of the text separately. After the processing of all

input words, a flag matrix similar to Table VI will be generated. Note that each row has a single 1 in the column representing the word assigned category.

Table VI. Percentage and difference categorization algorithm process.

Input Word/Category	X_1	X_2	X_3 ...	X_i	X_m
z_1	1	0	0	0	0
z_2	0	0	1	0	0
z_3	0	1	0	0	0
. . .	1	0	0	0	0
z_j . . z_n	. . . 1	. . . 0	. . . 0	c(i,j) . . 0	. . . 0
Category Sum	∑c(1,k)	∑ c(2,k)	∑ c(3,k)	∑ c(i,k)	∑ c(m,k)

The category with the highest sum of flag values (as seen in Table VI) is considered to be the best match for the input text.

5.3 PDC algorithm vs BCA

Before we move on with our categorization process, we need to select which algorithm is better, in order to adopt for further processing.

To do that, we applied both algorithms to the testing samples discussed in Section 3. As mentioned earlier, two phases of categorization were applied for both algorithms: first into the main categories (رياضة, علوم, هندسة, دين, تاريخ, ادب, طب, سياسة, اقتصاد) in English (Sports, Science, Engineering, Religion, History, Medicine, Economics, Politics) then, as the second phase, into the subcategories of the main category selected in phase one.

One may argue that the subcategories might be highly correlated, and some words such as مباراة (Game) in رياضة (Sports) might not help differentiate between inner subcategories (such as كرة سلة, كرة قدم, كرم مضرب ... الخ) in English (Basketball, Football, Tennis … etc). That's true; but for the current test we didn't apply any inner word (within the same major category) filtration. However, this issue will be discussed later in Section 5.5.

Table VII shows the comparison between PDC Algorithm and BCA. As can be seen in Table VII, the PDC algorithm gives better success rates, thus we will adopt this algorithm in our further processing.

Table VII. Basic categorization algorithm Vs PDC algorithm

Categories/ Subcategories	Basic Categorization Algorithm		PDC Algorithm	
	Success % (Subcategory)	Success% (Category)	Success% (Subcategory)	Success % (Category)
رياضة – كرة قدم Sport- Football	80%	97.5%	95%	98.34%
رياضة – كرة سلة Sport-Basketball	95%		95%	
رياضة – سباقات Sport- Racing	87.5%		92.5%	
هندسة – حاسوب Engineering- Computers	62.5%	85%	70%	85%
هندسة – كهرباء Engineering – Elec & Electronics	85%		82.5%	
علوم – فيزياء Science- Physics	57.5%	72.5%	65%	78.7%
علوم – كيمياء Science – Chemistry	72.5%		82.5%	
سياسة Politics	75%	75%	90%	90%
اقتصاد Economics	72.5%	72.5%	90%	90%
Overall Average	**76.39%**	**80.5%**	**84.72%**	**88.41%**

5.4 Enhancing Main/Subcategories Grouping

When it comes to categorizing using main/sub categories one of the main problems is the possible high correlation between subcategories of different main categories. For example هندسة كهربائية (Electrical Engineering) and فيزياء (Physics) are highly related. However فيزياء (Physics) comes from main category علوم (Science), not هندسة (Engineering), thus if a فيزياء (Physics) document was categorized as هندسة (Engineering) in phase one, then it will never be categorized as فيزياء (Physics) in phase two since فيزياء (Physics) is not a subcategory of هندسة (Engineering) in the hierarchy. To solve this we adopted the two approaches discussed next.

5.4.1. Overlapping main categories for phase two: The idea is to allow main categories to overlap by having shared subcategories (from other categories) that are related to the inner subcategories. This is to preserve the ability to correctly categorize in phase two even when phase one categorization fails due to common features of subcategories from different main categories. For example if an input هندسة كهربائية (Electrical Engineering) text was categorized as علوم (Science) in phase one (say due to the presence of many physics terms), then in phase two it will not only be categorized under one of علوم (Science) subcategories but also with the added هندسة (Engineering) subcategories and صحة (Health) subcategories. That is, we add subcategories (from other main categories) related to each subcategory in علوم (science).

5.4.2. Replacing main categories by groups of related categories: We believe that main categories used so far are not adequately related internally. It is not clear that أحياء (Biology) is closer to فيزياء (Physics) than to طب (Medicine). Since these divisions are transparent to the final categorizing process, one may modify the first phase main

categories to assist the process. The idea here is to redefine the main categories and replace them by groups of related subcategories (dummy/working categories), as shown in Table VIII.

Table VIII. New definition of main and subcategories

#	Major Categories (Groups)	Subcategories
1	Group1: رياضة (Sport)	كرة قدم, كرة سلة, كرة مضرب, سباقات, العاب أولمبية Football, Basketball, Racing, Tennis, Olympics
2	Group2: فيزيائيات (Physics Related)	فيزياء, هندسة كهربائية, هندسة ميكانيكية Physics, Elec & Electronics, Mechanics
3	Group3: حاسوبيات (Computing)	حاسوب, رياضيات Computers, Mathematics
4	Group4: طب و مختبرات (Medicine Related and Labs)	أحياء, كيمياء, طب, صيدلة Biology, Medicine, Pharmacy, Chemistry
5	Group5: أدبيات (Literatures)	موسيقى, سينما, أدب, موضة Literature, Cinema & Theatres, Fashion, Music
6	Group6: تاريخ (History)	تاريخ حديث, تاريخ قديم New History, Old History
7	Group7: ديانات (Religions)	الديانة الإسلامية, الديانة المسيحية, ديانات أخرى Islam, Christianity, Other Religions
8	Group8: اقتصاد (Economics)	-----
9	Group9: سياسة (Politics)	-----

In phase one, an input text will be categorized under one of the nine groups shown in Table VIII, then it will be subcategorized within the selected group. We re-did the testing on PDC algorithm using the same test sample and using the discussed two approaches. Table IX shows the results.

So what we did here is very simple, we re-defined the main categories in a way that the inner subcategories of those new main categories (groups) are highly correlated, then we applied hierarchical categorization by first categorizing an input text into one of the groups, then the subcategories the selected group (as an output) will be used to categorize the text into a subcategory under that group.

Table IX. PDC algorithm with overlapping and modified grouping

Subcategory	PDC Algorithm Success Percentage	
	(Original Grouping - Overlapping)	(Modified Grouping)
كرة قدم (Football)	97.5%	97.5%
كرة سلة (Basketball)	92.5%	92.5%
سباقات (Racing)	90%	90%
فيزياء (Physics)	80%	82.5%
كهرباء (Elec & Electronics)	85%	90%
حاسوب (Computers)	75%	77.5%
كيمياء (Chemistry)	85%	87.5%
سياسة (Politics)	90%	90%
اقتصاد (Economics)	90%	90%
Average	**87.22%**	**88.61%**

As can be seen in Table IX, the results improved under approach two from 84.72% to 88.61%, so we will adopt approach two and the groups in Table VIII as our new main reference categories. Note, that here we are comparing under the second level of categorization (using subcategories results). The output is not comparable regarding

main categories because we re-defined the main categories into groups in approach two, but under subcategories it's still comparable.

5.5 Word filtration techniques within categories

Next we try to remove or reduce the effect of correlation between subcategories of each group. For example the word مبارة (Match) can help categorizing a document as رياضة (Sports), however it might not help when deciding between subcategories of رياضة (Sports), that is the word is used in most of the subcategories, thus might be treated like a stop, non discriminating word in phase two.

We investigated three techniques to filter out such words and to check the filtering effect on the results. Those techniques depend on the definition of inverse document frequency (idf): a measure of whether the word (term) is common or rare across all documents. It is obtained by dividing the total number of documents by the number of documents containing the word, and then taking the logarithm of that quotient[4].

Technique 5.5.1. Remove any word seen in two or more subcategories of the given main category. For example if word w is seen in subcategories xi and xj in main category X, then remove w from all subcategories of X. It's as if we are removing all words for which Equation (2) applies. n is the number of all subcategories in category X and m is the number of all subcategories that have the word w.

$$\text{idf}(w,X) = \log(n/m) \text{ ; where m in the range of } (2,n) \quad (2)$$

Technique 5.5.2. Remove any word that is shared in all subcategories of a main category. For example if a word w is seen in all subcategories of X, then remove w from all subcategories of X during the test. It's as if we are removing all words for which Equation (3) applies.

$$\text{idf}(w,X) = \log(n/m) = 0 \text{ ; where } m = n \quad (3)$$

Technique 5.5.3. Detect any word that is seen in two or more subcategories in a given category, and then only keep the word in the subcategory in which it has the highest percentage. For example if a word w is seen in subcategories x_i, x_j and x_k of a main category X with frequencies P_i, P_j and P_k respectively and $max(P_i, P_j, P_k)=P_j$ then keep w in the subcategory x_j and remove it from the rest of the subcategories of X. Same as technique 5.5.1 but we keep the word in the subcategory with highest percentage.

Table X shows the result of applying the three techniques with the testing sample.

It is seen that adopting *Technique 4.5.3.* outperforms others. So using PDC algorithm with re-defined categories (as groups) and the third filtration technique gave the best results so far.

[4] Retrieved from http://en.wikipedia.org/wiki/Tf%E2%80%93idf

Table X. The results with different filtration of inner stop words techniques on PDC

Category	Plain Algorithm	Technique 1	Technique 2	Technique 3
رياضة – كرة قدم Sport - Football	97.5%	95%	97.5%	97.5%
رياضة – كرة سلة Sport-Basketball	92.5%	92.5%	92.5%	95%
رياضة – سباقات Sport- Racing	90%	90%	87.5%	92.5%
حوسبة – حاسوب Computing- Computers	82.5%	75%	75%	77.5%
علوم فيزيائية – كهرباء Physics Related – Elec & Electronics	90%	85%	90%	90%
علوم فيزيائية – فيزياء Physics Related - Physics	77.5%	27.5%	70%	85%
صحة – كيمياء Health –Chemistry	87.5%	85%	87.5%	87.5%
سياسة Politics	85%	85%	85%	85%
اقتصاد Economics	90%	90%	90%	90%
Average:	**88.06%**	**80.56%**	**86.11%**	**88.89%**

5.6 Modified PDC with N Scales

To investigate our categorization algorithm more we edited the measurements in the PDC algorithm to allow for multi-valued instead of binary scaling. In subsection 5.2, we mentioned that when categorizing a word *z*, it will be assigned to the category with minimum difference and for each category there is a flag that is set to 1 if the category holds the minimum difference for *z* and is set to zero for all other categories. We investigated the behavior of the algorithm when the assignment can have more values, such as three values [1 or 0.5 or 0], or five values [0, 0.25, 0.5, 0.75, and 1].

In order to define a scaling of [1 or 0.5 or 0] we need to introduce the following rules:

Rule 5.6.1 (Minimum Value). This will be the minimum value found after calculating all differences between each of the categories and the given word.

Rule 5.6.2 (Maximum Value). This will be the maximum value found after calculating all differences between each of the categories and the given word.

Rule 5.6.3 (Middle Value). This will be the result of (Minimum Value + Maximum Value)/ 2 (the midpoint of the range).

Rule 5.6.4 (Break1 Value). [Minimum Value + Middle Value]/2.

Rule 5.6.5 (Break2 Value). [Maximum Value + Middle Value] /2.

And for [0, 0.25, 0.5, 0.75, and 1] scaling we need also:

Rule 5.6.6 (Value A). [Break1 Value+ Minimum Value] / 2.

Rule 5.6.7 (Value B). [Break1 Value + Middle Value] / 2.

Rule 5.6.8 (Value C). [Middle Value+ Break2 Value] / 2.

Rule 5.6.9 (Value D). [Maximum Value + Break2 Value] / 2.

Here is the line of values.

```
Min          Brk1         Mid          Brk2         Max
  |-------o-------x-------o-------|-------o-------x-------o-------|
Min    A             B     Mid    C             D    Max
```

Table XI shows which values will convert to what.

Table XI. PDC - Scales & Values

Difference value between input and reference	Value
PDC Scale of 3	
[Minimum – Break1]	1
(Break1 – Break2)	0.5
[Break2 – Maximum]	0
PDC Scale of 5	
[Minimum, A]	1
(A,B]	0.75
(B,C]	0.5
(C,D)	0.25
[D, Maximum]	0

The operation from here on is the same as the original algorithm, after assigning a category with a value for each word in the input depending on the interval, the category with highest sum is considered to be the category of the input text.

Table XII shows a comparison between PDC algorithm with and without scales. As can be noticed from Table XII the binary scale gave the best results, then the 3-valued scale followed by the 5-valued scale, so we can predict that if we continue dividing the scale to more points the results will not actually improve, thus having a continuous scale will not improve the results, so we will keep the algorithm as is. (with the binary scale).

Other experiments done on the PDC algorithm consisted of applying (on both the reference/training and testing sets) the preprocessing tools discussed in Section 4.

Table XII. The results with different Scales on PDC

Category	PDC-Binary Scale	PDC 3-valued Scale	PDC 5- valued Scale
رياضة – كرة قدم Sport – Football	97.5%	95%	97.5%
رياضة – كرة سلة Sport-Basketball	95%	95%	95%
رياضة – سباقات Sport- Racing	92.5%	87.5%	87.5%
حوسبة – حاسوب Computing- Computers	77.5%	75%	77.5%
علوم فيزيائية – كهرباء Physics Related – Elec & Electronics	90%	82.5%	90%
علوم فيزيائية – فيزياء Physics Related – Physics	85%	70%	62.5%
صحة – كيمياء Health –Chemistry	87.5%	87.5%	87.5%
سياسة Politics	85%	70%	55%
اقتصاد Economics	90%	87.5%	87.5%
Average:	**88.89%**	**83.33%**	**82.22%**

5.7 Further Testing on PDC algorithm

In order to study the effect of other preprocessing tools (both on the input and reference data), we applied the following four tools:

Tool 5.7.1. (Root Extraction). Extracting the roots of both the input text and the reference categorized data before applying the categorizing algorithm.

Tool 5.7.2. (Light Stemming & Light10). Light stemming both the input text and reference categorized data before applying the categorizing algorithm.

Tool 5.7.3. (Double Words). Processing both the input and reference text as expressions of double (not as single) words, before applying the categorizing algorithm.

Tool 5.7.4. (Expressions Extraction). Filter expressions from both input and reference data and use them with the categorizing algorithm.

Table XIII shows the results after applying the above tools.

Table XIII. Test results with extraction tools

Categories	Overall Average Pass Percentage
No Tools	88.9%
Root Extraction	75.1%
Double Words	84.9%
Light Stemming	87.5%
Light 10	81.9%
Expressions	86.9%

As seen in Table XIII the algorithm with no additional tools gives the best results, thus we will keep the PDC without these tools.

Also it can be noticed that using our light stemmer with PDC gave better results than using Light10, and since the categorization algorithm is not changed while testing, we can conclude that the impact of using our stemmer on categorization is better than the impact of using Light10. It seems to be the case that using a reference set of stemmed words in a stemmer as we do, is better than not using one as is the case for Light10, of course at the expense of the added cost of the lookup step for checking the generated stem in the reference list.

5.8 Using testing data from the reference categories

Our testing was based on external testing data sets, as we explained in Section 3. However we did investigate the results when the training data and testing data came from the same source (Arabic Wikipedia). This was done by splitting our corpus data to 66% training and 34% testing; the selection of the testing data and training data from the same corpus was done three times by selecting the first 34%, the middle 34% and the last 34% as test. Also by using 100% of the corpus for training and the overlapping last 34% as testing data (last 34% gave best results). Table XIV shows the results.

Table XIV. Test results witin same refrence

#	Training/Testing data Splitting	Pass Percentage for : PDC-Binary Scale	Training: x Testing: _
1	66% Training, 34% Testing (First)	94.1%	____xxxxxxxx
2	66% Training, 34% Testing (Middle)	96.1%	xxxx____xxxx
3	66% Training, 34% Testing (Last)	96.3%	xxxxxxxx____
4	100% Training, 34% Testing (Last)	99.0%	xxxxxxxxxxxx
5	100% Training, External Testing	88.9%	xxxxxxxxxxxx ____

Note that when the training and testing data came from same source (Arabic Wikipedia) the results will be better (96.3% vs 88.9%). Also one can note that the location (first, mid or last third) of selected testing data in the corpus can result in slightly different results.

6. COMPARISON WITH RELATED WORK

As mentioned in Section 2, we want to compare our results with the work of [Saad 2011]. His work does not present a single approach but rather an application of several categorization algorithms, the author applied different algorithms on Arabic by applying also different Arabic corpora. So the best way to compare our work with others is simply comparing our work with this author's work since he already compared well known algorithms and highlighted the best of them, thus using his corpora on our algorithm is just like comparing our work with all the algorithms he tested[Saad 2011]. We will not need to re-implement the algorithms and methods used there, but rather we need only to use the available data resources he used and apply them to our algorithm. That is, we use his training data instead of the Arabic Wikipedia.

The author applied different classification algorithms such as: C4.5 Decision Tree (TD), K Nearest Neighbors (KNN), Support Vector Machines (SVMs), Naïve Bayes (NB), Naïve Bayes Variants (Naïve Bayes Multinomial -NBM), Complement Naïve

Bayes (CNB) and Discriminative Multinomial Naïve Bayes (DMNB) in his testing, and concluded that light stemming with SVMs outperforms other algorithms.

The author tested seven corpora differing in size and the number of categories in each and most of them are available for free online as raw data. Thus they need processing to be ready for use with our algorithm by removing stop words, punctuation marks, non-Arabic characters, extraction into database and calculating frequencies of appearance in order to fit the needs of our algorithm. Table XV shows the characteristics of the training data (66% of the total documents) of the corpora we used from this author (after processing them to fit our algorithm).

Table XV. Training data characteristics

	Categories	# of documents	# of total words	# of distinct words
1	OSAC (Open Source Arabic Corpus)			
	Economic	2047	1066188	63429
	History	2134	3680098	210451
	Education & Family	2381	2267196	163685
	Religious	2093	1101102	60370
	Sport	1596	610091	49286
	Health	1515	1142330	30642
	Astronomy	367	197944	26683
	Law	623	614731	29449
	Stories	470	666194	87100
	Cooking Recipes	1566	273558	18001
2	BBC Corpus			
	Economics	195	68147	11865
	Technologies	153	53134	11743
	Middle East News	1555	572747	46076
	World News	983	339217	36054
	Newspapers	32	30713	9374
	Sports	145	50421	8616
	Miscellaneous	81	33796	6960
3	CNN Corpus			
	Business	552	229487	23841
	Entertainment	313	133067	29260
	Middle East	695	340459	39116
	Science and Technology	526	124776	21739
	Sport	347	191866	22174
	World News	667	270964	30367
4	Aljazeera Corpus			
	Art	198	56615	17123
	Economics	198	44557	10160
	Politics	198	60195	15230
	Science	198	48691	11524
	Sport	198	49561	10549
5	Khaleej Corpus			
	Economics	600	281149	31948
	International News	629	344962	37909
	Local News	1582	637811	60844
	Sport	944	365079	38150

We used the corpora (one at a time) with our algorithm, by using each of the corpora (66% of total documents) as reference data (categories) and the rest of the documents (34%) of each of the corpora as the testing data. Table XVI compares our results with the best results of [Saad 2011].

From Table XVI it can be noted that, on average, our algorithm gives better results. One important note here is that we don't have information about how the

66% and 34% splitting (in the compared work) is done, it's likely that our 66% training data might not hold the same content as the 66% training data of the compared work and the same is said about testing data. So there is a possibility that if we selected a different 66% of the content (selecting different documents), we may end up with different results (we proved that in Table XIV on own our data). which may explain why we had a lower (but close) pass percentage (96.6% vs 99.3%) in the test of OSAC corpus.

Table XVI. Comparing our results with the best results of [Saad 2011]

#	Corpus	Our Algorithm	SVM-Light Stemming
1	OSAC " Open Source Arabic Corpus"	96.6%	99.3%
2	BBC Corpus	90%	90%
3	CNN Corpus	91.3%	86%
4	Aljazeera Corpus	93.76%	89%
5	Khaleej Corpus	93.7%	88.5%
	Average	**93.07**	**90.56**

7. CONCLUSIONS AND FUTURE WORK

In this paper we presented our experiments on categorizing Arabic text with a focus on a hierarchy of main categories and subcategories. For that we employed categorized corpora obtained from the Arabic Wikipedia which we built using a combination of automated and manual processes.

We introduced a categorization algorithm that started with a simple weighting idea and progressed to a more complex one that considers the relation of weights in input text and training data.

We designed and implemented different text pre-processing tools such as root extractor, light stemmer and expression extractor and tested their effect on the performance of our categorizing algorithm. We noted that light stemming with a reference list of pre-stemmed words is a better approach for light stemming even though light stemming (in general) didn't give the better results when incorporated into our categorization algorithm.

Another important conclusion is that there are two methods for testing categorization algorithms: the first is to use training and testing data from same source by splitting the corpus into test and training components. This consistently gives better results than the second method in which training and testing data come from different sources. Most of the early work use the first method. However we believe that the second method makes more sense as the tests will be more credible and indicative of performance in real life environments.

Regarding future work on categorization, we are interested in investigating the manual tags found in articles in the Arabic Wikipedia (tags that are added by editors at the end of each article). We plan to group related tags into more general tags to end up with well defined major tags, and those tags will be used with the article titles in the process of further categorization.

REFERENCES

S. Al-Harbi, S. Almuhareb, A. Al-Thubaity, and M. -S. Khorsheed. 2008. Automatic Arabic Text Classification. 9es Journées internationales d'Analyse statistique des Données Textuelles. Retrieved from http://lexicometrica.univ-paris3.fr/jadt/jadt2008/pdf/harbi-almuhareb-thubaity-khorsheed-rajeh.pdf.

R. Al-Shalabi, G. Kanaan, and G. –H. Manaf. 2006. Arabic Text Categorization Using kNN Algorithm, in Proc. 4th International Multi-conference on Computer Science and Information Technology. Vol. 4. Amman, Jordan, (2006). Retrieved from http://www.uop.edu.jo/download/research/members/CSIT2006/vol4%20pdf/pg20.pdf

S. Alsaleem. 2011. Automated Arabic Text Categorization Using SVM and NB. International Arab Journal of e-Technology, Vol2, No.2. Retrieved from http://www.iajet.org/iajet_files/vol.2/no.2/Automated%20Arabic%20Text%20Categorization%20Using%20SVM%20and%20NB_doc.pdf

M. Attia. 2011. Word Count of Modern Standard Arabic. Retrieved from : http://sourceforge.net/projects/arabicwordcount/

M. El-Kourdi, A. Bensaid, and T. Rachidi. 2004. Automatic Arabic Document Categorization Based on the Naïve Bayes Algorithm. 20th International Conference on Computational Linguistics, Genev. Retrieved from http://acl.ldc.upenn.edu/W/W04/W04-1610.pdf

A. Goweder and A. De Roeck. 1998. Assessment of a Significant Arabic Corpus. Retrieved from http://www.abdelali.net/ref/ACL-EACL%202001_goweder.pdf

L. Khreisat. 2006. Arabic Text Classification Using N-Gram Frequency Statistics: A Comparative Study. 6th Conference on Data, Monte Carlo Resort, Las Vegas, Nevada, USA. Retrieved from http://ww1.ucmss.com/books/LFS/CSREA2006/DMI5552.pdf

L. S. Larkey, L. Ballesteros, and M. E. Connell. 2007. Light Stemming for Arabic Information Retrieval. Arabic Computational Morphology Text, Speech and Language Technology. Volume 38. pp 221-243, (2007). Retrieved from http://link.springer.com/content/pdf/10.1007%2F978-1-4020-6046-5_12

X. Qiu, X. Huang, Z. Liu and J. Zhou. 2011. Hierarchical Text Classification with Latent Concepts. HLT '11 Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers. Volume 2. pp. 598-602. Retrieved from https://www.aclweb.org/anthology/P/P11/P11-2105.pdf

M. Saad. 2011. The Impact of Text Preprocessing and Term Weighting on Arabic Text Classification. Faculty of Engineering, The Islamic University, Gaza, Palestinian Territories. Retrieved from http://library.iugaza.edu.ps/thesis/91986.pdf

A. Sarkar, A. De Roeck and P. Garthwaite. 2004. Easy measures for evaluating non-English corpora for language engineering: Some lessons from Arabic and Bengali. Technical report, Technical Report 2004/05, Department of Computing, Open University. Retrieved from http://computing-reports.open.ac.uk/2004/2004_05.pdf

M. M. Syiam, Z. T. Fayed and M. B. Habib 2006. An intelligent system for Arabic text categorization. International Journal of Intelligent Computing and Information Sciences. Volume 38. pp 221-243, (2007). Retrieved from http: http://eprints.eemcs.utwente.nl/19190/01/IJICIS2006.pdf

A. Yahya, and A. Salhi. 2011. Enhancement Tools for Arabic Web Search : A Statistical Approach. 7th International Conference on Innovations in Information Technology, Abu Dhabi, United Arab Emirates. Retrieved from http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5893871&isnumber=5893793

A.Yahya and A. Salhi. 2012. Arabic Text Correction Using Dynamic Categorized Dictionaries: A Statistical Approach. Linguistica Communicatio Journal (Selected Papers from CITALA 2012). Volume 5, 2013.

Y. Yang, J. Zhang and B. Kisiel. 2003. A scalability analysis of classifiers in text categorization. Proceedings of SIGIR-03, 26th ACM International Confereumnce on Research and Development in Information Retrieval. eds. J. Callan,G. Cormack, C. Clarke, D. Hawking & A. Smeaton, ACM Press, New York, US: Toronto, CA, pp. 96–103. Retrieved from http://nyc.lti.cs.cmu.edu/yiming/Publications/yang-sigir03.pdf