ABHATH AL-YARMOUK: "*Basic Sci. & Eng.*" Vol. 21, No. 1, 2012, pp. 85-105

N-Gram-Based Techniques for Arabic Text Document Matching; Case Study: Courses Accreditation

Mohammad A. Al-Ramahi* and Suleiman H. Mustafa**

Received on Feb. 28, 2011 **Accepted for publication on Oct. 10, 2011**

Abstract

Measuring text similarity has been studied for a long time due to its importance in many applications in natural language processing and related areas such as Web-based document searching. One such possible application which is investigated in this paper is determining the similarity between course descriptions of the same subject for credit transfer among various universities or similar academic programs. In this paper, three different bi-gram techniques have been used to calculate the similarity between two or more Arabic documents which take the form of course descriptions. One of the techniques uses the vector model to represent each document in a way that each bi-gram is associated with a weight that reflects the importance of the bi-gram in the document. Then the cosine similarity is used to compute the similarity between the two vectors. The other two techniques are: word-based and whole document-based evaluation techniques. In both techniques, the Dice's similarity measure has been applied for calculating the similarity between any given pair of documents. The results of this research indicate that the first technique has demonstrated better performance than the other two techniques as viewed with respect to the human judgment.

Keywords: N-Gram Similarity Measures, Cosine Similarity Measure, Vector Model, Dice's Similarity Measure, Course Description Matching, Course Credit Transfer.

Introduction

According to Ethan M, et al., [1], text similarity is the measure of how alike two documents are, or how alike a document and a query are. Measures of text similarity have been used for a long time in applications in natural language processing and related areas [2]. One of the earliest approaches of text similarity is perhaps the vectorial model in information retrieval, where the document most relevant to an input query is determined by ranking documents in a collection in reversed order of their similarity to the given query.

* Department of Management Information Systems, Faculty of Information Technology and Computer Sciences, Yarmouk University, Irbid, Jordan.

** Department of Computer Information Systems, Faculty of Information Technology and Computer Sciences, Yarmouk University, Irbid, Jordan.

In this model, each document is represented as a vector of words. More formally, each word is associated with a weight w_i that reflects its importance in the document as follows:

$$d_j = (w_{1,j}, w_{2,j}, w_{3,j}, \ldots\ldots\ldots, w_{n,j})$$

Where

d_j : Document j

$w_{1,j}$: Weight of the first word in document j

The weight of a word in a document can be calculated by the frequency of the word in the document normalized by the maximum frequency in that document and multiplied by the inverse document frequency (*idf*) of the word as follows:

$$w_{i,j} = f_{i,j} \times idf$$

Where

$w_{i,j}$: The weight of word i in document j

$f_{i,j}$: The frequency of the word i in the document j normalized by the maximum frequency in that document

$$idf \text{ (the inverse document frequency)} = \log \frac{N}{n_i}$$

N : The total number of documents in the collection

n_i : The number of documents contain word i

Such term-weighting strategies are called *tf-idf* (term frequency inverse document frequency). Many similarity measures can be used to calculate the similarity between any two vectors (cosine, dice, jaccard or inner product) [3].

Another approach of text similarity is clustering, in which we group the given set of documents according to their similarity into different clusters. Benjamin C.M. et al., [4] used a method built on this idea that similar documents can be identified by some common words, called frequent item sets. By finding the frequent item sets that are common to a number of documents we can put these documents in one cluster.

In order to calculate the similarity between two documents or between a document and a query, the first challenge is to decide what will be used to represent a document or a query. Since a document is a sequence of words, a very common way is to view a

document as a bag of words. After indexing a document, the common words such as “or”, “and”, and "the" should be removed.

Generally, the effectiveness of the matching process that views a document as sequence of words (bag of words) depends on the number of identical words in the two documents. Because of the word variation in the documents, this will decrease the similarity value among two similar documents unless you have an accurate stemmer. So, the second challenge in calculating the similarity between two texts is word variation. That is, a word can exist in several forms. For example, the word “connect” may exist as “connects”, “connected” or “connection”. The most common types of variation that are encountered in textual databases are affixes, multiword concepts, spelling errors, alternative spellings, transliteration, and abbreviations [5]. So we need an additional step to handle the word variation problem. Conflation stemming is the act of bringing together nonidentical textual words that are semantically related and reducing them to a controlled or single form for retrieval purposes [5].

The primary goal of conflation is to allow matching of different variants of the same word; in terms of standard information retrieval quality measures, conflation improves recall (the quotient of the number of retrieved relevant documents and the total number of relevant documents). In addition to that, precision (quotient of the number of retrieved relevant and number of retrieved documents) can be positively affected, as several terms in the same documents can be conflated to the same index term, which can lead to a change in similarity to the query and thus the ranking. Furthermore, conflation can reduce the size of the document index significantly, because there are fewer distinct index terms that need to be stored [6].

Several conflation techniques have been used to handle the word variation problem. As in [6], the usual approach to conflation in information retrieval is the use of a stemming algorithm that tries to find the stem of a word, which is the basic form from which inflected forms are derived. For example, the stem of both “connection” and “connects” would be “connect”.

Frakes [7] distinguishes between four types of stemming strategies: affix removal, table lookup, successor variety, and n-grams. Table lookup consists simply of looking for the stem of a word in a table. Successor variety stemming is based on the determination of morpheme boundaries, uses knowledge from structural linguistics, and is more complex than affix removal stemming algorithms. N-grams stemming is based on the identification of digrams and trigrams and is more a term clustering procedure than a stemming one.

The stemming algorithms only address the problem of morphological variants of words, ignoring the problem of misspellings. One simple method for automatic spelling correction is to use N-gram technique as stemming technique. This technique breaks up a text document into several n-character long unique grams, and produces a vector whose components are the counts of these grams [8]. The issue is further complicated by the fact that the stemmer might not be efficient for the matching process in some

applications [9]. For example, electronic documents that are produced by scanning and optical character recognition OCR can contain errors due to misrecognition.

This paper uses the N-gram matching approach to compute the similarity between two Arabic texts in three different techniques: word-based using Dice's similarity measure, word-based using cosine similarity and whole document-based using Dice's similarity measure. N-gram techniques have been widely investigated for a number of text processing tasks. According to [10], a character N-gram is an N-character slice of a longer string. For example, the word " INFORM " produces the 5-grams "_INFO", "INFOR", "NFORM", and "FORM_" where the underscore represents a blank. The key benefit of N-Gram-based matching derives from its very nature: since every string is decomposed into small parts, any errors that are present tend to affect only a limited number of those parts leaving the remainder intact. The N-Grams for related forms of a word (for instance, "information", "informative"," informing", etc.) automatically have a lot in common. If we count N-Grams that are common in two strings, we get a measure of their similarity that is resistant to a wide variety of grammatical and typographical errors.

In this paper, the N-gram similarity approach has been applied for course credit transfer between universities. A common practice in most universities is allowing students who transfer from a university to another to count some of the courses taken before transfer to the new institution. The process of credit transfer is carried out by comparing the descriptions of the courses that have been studied in the old university with those that exist in the degree plan of the new university. Two courses are considered equivalent, and hence can be considered for credit transfer, if a high degree of similarity exists between the descriptions of the two courses. The main objective of this research was to investigate how N-gram-based matching techniques can be used to handle the course transfer problem in the case of Arabic course descriptions and how well the performance of these techniques compare with the human-based similarity judgment.

Related Work

The word-based N-gram technique was used by Miller et al. [1] in information retrieval for English texts in similar way to vector space model. They used N-grams rather than words as index terms. Each document was represented by a vector of N-grams rather than words. Each N-gram was associated with a weight w_{ik} that reflects the importance of N-gram in the document as follows:

$$w_{ik} = f_{ik} - a_k$$

Where, the frequency (f_{ik}) of N-gram$_k$ is its count normalized by the total number of N-grams in document I, and a_k is the average normalized frequency overall documents. The similarity between two document vectors is then calculated as the cosine of the two representation vectors.

$$Sim_{(d_J, d_K)} = \frac{\sum_{i=1}^{t} W_{i,j} W_{i,k}}{\sqrt{\sum_{i=1}^{t} W^2_{i,j}} \times \sqrt{\sum_{i=1}^{t} W^2_{i,k}}}$$

where

$w_{i,j}$: The weight of a bi-gram (*i*-th bi-gram) in the document (j).

$w_{i,k}$: The weight of a bi-gram (*i*-th bi-gram) in the document (k).

.Miller et al. [1] showed that it is possible to build a text information retrieval engine using N-grams rather than words as terms that can handle gigabyte-sized corpora. They also adapted techniques that had been used for word-based systems to N-gram-based information retrieval, making adjustments as necessary to account for the different term distributions exhibited by N-grams.

In [11], a vector processing model was used for documents and queries, but by using N-gram frequencies as the basis for the vector element values instead of more traditional term frequencies. For documents, this study used term weights of the form:

$$w_j = \frac{\left(\log_2 \left(tf_j\right) + 1\right)}{\sqrt{\sum_i w_i}}$$

Where

w_j : is the weight of *j*th quad-gram in the document

tf_j : is the term frequency of the *j*th quad-gram in the document

w_i : is the weight of *i*th quad-gram in the document

For queries, the study used similar term weights, but with *idf* (the inverse document frequency) defined follows:

$$w_j = \frac{\left(\log_2 \left(tf_j\right) + 1\right) \cdot idf_j}{\sqrt{\sum_i w_i}}$$

where

$$idf_j = \log_2\left(\frac{N}{n_j}\right)$$

where

N: is the number of the documents in the collection

n_j : is the number of documents in the collection containing at least one occurrence of the jth quad-gram.

According to [11], this approach has many advantages, including:

- It provides a robust retrieval system that can tolerate spelling errors in both documents and queries.
- It requires no linguistic pre-processing of documents or queries to perform word-stemming or stopword removal. Thus it is also inherently language independent.
- It allows the system to accrue all of the benefits of the vector processing model, including being able to manipulate documents and queries in a uniform way. For example, it easy to use a retrieved document as a query for a more refined search, such as is necessary for relevance feedback systems.

The work in [5] used an N-gram conflation method to convert a word into a sequence of N-grams and apply it within the context of Arabic textual retrieval systems. In this study the similarity between two words was calculated through dividing the number of unique identical N-grams between them by total number of unique N-grams in the two words (Dice's similarity coefficient).

$$S = \frac{2C}{A + B}$$

where S is the similarity value, A and B are the respective numbers of unique N-grams in word one and word two, and C is the total number of unique N-grams that are common for both words being compared.

The results of this study indicate that the digram method offers a better performance than trigram with respect to conflation precision/recall ratios. Also the study indicates that the N-gram approach does not appear to provide an efficient conflation approach within Arabic context.

Another formula is used in [12] that can also be used to calculate the similarity between two N-grams sets, and is:

$$S(w_1, w2) = \frac{\left|M(w_1) \cap M(w_2)\right|}{\sqrt{\left|M(w_1)\right| * \left|M(w_2)\right|}}$$

where $\left|M(w_1)\right|$ is the number of different N-grams in $w1$.

The work in [10] produced a system capable of discovering rules based on a rich and varied set of features that are useful to the task of discriminating between text documents. In this study, a novel method for using Genetic Programming to create compact classification rules based on combinations of N-Grams is described. According to this study, the frequency of a particular N-Gram could be a simple count of the occurrence of an N-Gram in a document or a more sophisticated measure such as the term frequency inverse document frequency (*tf-idf*).

Methodology

Three different N-gram-based techniques are used in this paper, including word-based using dice's similarity measure, word-based using cosine similarity measure and whole document-based. In each technique stop words are removed as a first preprocessing step. In addition, the Arabic language differs from English in that the article (ال التعريف) is not separated from the word and this might affect the accuracy of the techniques. So, it has been decided to remove it with all variations during this step. For example, "بالتعويض" is preprocessed into "تعويض".

In order to explain how each technique works, the following are two short texts have been extracted from the problem domain which represent two different descriptions of a calculus course.

D1= { التكامل المحدود، التكامل بالأجزاء، التكامل بالتعويض، التكامل بالكسور الجزئية، المتتاليات والمتسلسلات}

D2={التكامل المحدود، التكامل بالأجزاء، التكامل بالتعويض، التكامل بالكسور الجزئية، الإحداثيات القطبية}

Word-Based Using Dice's Similarity Coefficient

In this technique, bi-gram will be applied on each individual word in the document in such a way that each word will be represented by a set of bi-grams. So Dice's similarity coefficient can be applied to calculate the similarity between the two sets of bi-grams. A predefined threshold is used to decide if the two words are similar or not according to similarity value S.

By taking the unique words only in each document and calculating the similarity between each two words in the two documents and counting the similarity values of S that above predefined threshold (i.e. common words in two documents); the similarity between the two documents can be calculated using Dice's Coefficient after modifying it to be applied on document level rather than word level, where *A* and *B* become the respective numbers of unique words in two documents rather than unique bi-grams in

two words and *C* becomes the total number of unique words that are common for both documents (i.e. similarity between them above threshold) being compared rather than the total number of unique N-grams that are common for both words being compared.

Given the two documents D1 and D2 in Table1:

Table 1: Words appearing in the two documents D1 and D2

Document Number	Text (Course description)
D1	التكامل المحدود، التكامل بالأجزاء، التكامل بالتعويض، التكامل بالكسور الجزئية، المتتاليات والمتسلسلات
D2	التكامل المحدود، التكامل بالأجزاء، التكامل بالتعويض، التكامل بالكسور الجزئية، الإحداثيات القطبية

After article "ال" and stop words removal, each text will be tokenized into words as illustrated Table2

Table 2: the words of D1 and D2

D1\Word	D2\Word
تكامل	تكامل
محدود	محدود
تكامل	تكامل
أجزاء	أجزاء
تكامل	تكامل
تعويض	تعويض
تكامل	تكامل
كسور	كسور
جزئية	جزئية
إحداثيات	متتاليات
قطبية	متسلسلات

Table 3 shows the Bi-grams set for document D1 obtained by taking the unique words in the document (Table 2) and applying Bi-grams on them. The Table 4 shows the same thing for document D2

Table 3: Bi-grams for each word in D1

Word	Bi-grams
تكامل	{تك، كا، ام، مل}
محدود	{مح، حد، دو، ود}
أجزاء	{أج، جز، زا، اء}
تعويض	{تع، عو، و، ،يض}
كسور	{كس، سو، ور}
جزئية	{جز، زئ، ئي، ية}
متتاليات	{مت، تت،، ا،ال، لي، يا، ات}
متسلسلات	{مت، تس، سل، لس، سل، لا، ات}

Table 4: Bi-grams for each word in D2

Word	Bi-grams
تكامل	{تك، كا، ام، مل}
محدود	{مح، حد، دو، ود}
أجزاء	{أج، جز، زا، اء}
تعويض	{تع، عو، وي، يض}
كسور	{كس، سو، ور}
جزئية	{جز، زئ، ئي، ية}
إحداثيات	{إح، حد، دا، اث،ث، ،يا، ات}
قطبية	{قط، طب، بي، ية}

Using Dice's Coefficient, the similarity measure between the two words <تكامل> and <تكامل> would be (2 * 4) / (4 + 4) =1 and between the two words <تكامل> and <محدود> would be (2 * 0) / (4 + 4)=0 and so on. By using 0.7 as a threshold value to

decide if two words are similar or not, the number of similar words in the two documents will be 6. So, the similarity measure between the two documents D1, D2 would be: (2 * Common unique words in two documents) / (Total number of unique words in two documents) (2 * 6) / (8 +8) = 0.75

Word-Based Using Cosine Similarity

In this technique, bi-grams are computed for each word in the document as in the first technique to address the word variation problem so that each document will be represented by vector of bi-grams, and then the equation

$$w_{i,j} = f_{i,j} \times idf$$

which has been cited above is used with modifications to assign a weight ($w_{i,j}$) to each bi-gram rather than each word, where: $f_{i,j}$ represents the frequency of a bi-gram (*i*-th bi-gram) in the document (*j*) normalized by the maximum frequency in that document**,** *N* represents the total number of documents in the collection, n_i represents the total number of documents contains *i*-th bi-gram. The similarity between two vectors is calculated according to cosine similarity measure given above:

$$Sim_{(d_j, d_K)} = \frac{\sum_{i=1}^{t} W_{i,j} W_{i,k}}{\sqrt{\sum_{i=1}^{t} W^2_{i,j}} \times \sqrt{\sum_{i=1}^{t} W^2_{i,k}}}$$

where $w_{i,j}$ represents the weight of a bi-gram (*i*-th bi-gram) in the document (*j*) and $w_{i,k}$ represents the weight of a bi-gram (*i*-th bi-gram) in the document (*k*).

As an example, suppose we have a collection of two documents D1 and D2 where D1= <تعريب البرامج> and D2= <البرامج الحاسوبية>. Each document will be represented as vector of bi-grams by applying bi-gram on each word as follow:

D1=< تع، عر، ري، يب، ال، لب، بر، را، ام، مج >

D2=< ال ، لب ، بر، را ، ام ، مج ، ال ، لح ، حا ، اس ، سو ، وب ، بي ، ية >

Now, the formula mentioned above (*tf-idf*) will be used to assign a weight for each bi-gram. The weight of (ال) bi-gram in D2, for example, can be computed as follows:

$$w_{ال,2} = f_{ال,2} \times \log \frac{N}{n_{ال}}$$

$$= \frac{2}{2} \times \log \frac{2}{2} = 0$$

Note that the maximum frequency in D2 is 2.

Table 5 contains the weights of all bi-grams in the two documents (D1 and D2). The first ten bi-grams are from D1 and the rest come from D2.

Table 5: Weights of all bi-grams in the two documents (D1 and D2)

N-gram	Document number	Freq.	n_i	Weight ($w_{i,j}$)
ال	D_1	1	2	0
ام	D_1	1	2	0
بر	D_1	1	2	0
تع	D_1	1	1	0.30103
را	D_1	1	2	0
ري	D_1	1	1	0.30103
عر	D_1	1	1	0.30103
لب	D_1	1	2	0
مج	D_1	1	2	0
يب	D_1	1	1	0.30103
اس	D_2	1	1	0.150515
ال	D_2	2	2	0
ام	D_2	1	2	0
بر	D_2	1	2	0
بي	D_2	1	1	0.150515
حا	D_2	1	1	0.150515
را	D_2	1	2	0

N-gram	Document number	Freq.	n_i	Weight ($w_{i,j}$)
سو	D_2	1	1	0.150515
لب	D_2	1	2	0
لح	D_2	1	1	0.150515
مج	D_2	1	2	0
وب	D_2	1	1	0.150515
ية	D_2	1	1	0.150515

The similarity between D1 and D2 can be calculated using cosine similarity given above as follows:

$$Sim_{(D_1, D_2)} = \frac{\sum_{i=1}^{t} W_{i,1} W_{i,2}}{\sqrt{\sum_{i=1}^{t} W^2_{i,1}} \times \sqrt{\sum_{i=1}^{t} W^2_{i,2}}}$$

$$= \frac{(0)(0)+(0)(0)+(0)(0)+(0)(0)+(0)(0)+(0)(0)}{\sqrt{0.362476} \times \sqrt{0.158583}}$$

$$= 0$$

Let us take a more practical example from the problem domain of the paper. Suppose we have the documents D1 and D2 that mentioned in first technique, in addition of other two courses description D3 and D4 as illustrate in table 6:

Table 6: Four course descriptions

Document Number	Text (Course description)
D1	التكامل المحدود، التكامل بالأجزاء، التكامل بالتعويض، التكامل بالكسور الجزئية، المتتاليات والمتسلسلات
D2	التكامل المحدود، التكامل بالأجزاء، التكامل بالتعويض، التكامل بالكسور الجزئية، الإحداثيات القطبية
D3	التكامل، التكاملات المنتهية و غير المنتهية، تطبيقات هندسية وفيزيائية
D4	مقدمة إلى نظرية المجموعات، الاحتمالات المنفصلة، الدوال، المخططات

According to this technique, the similarity values between the document D1 and the other documents are shown in Table 7:

Table 7: The similarity values between D1 and the other documents (D2, D3, and D4)

	Document Name	Similarity
D1	D2	0.466
	D3	0.063
	D4	0.057

Note that the similarity between the documents D1 and D2 is 0.466 and this is relatively small when compared with the similarity value result from the first technique.

Whole Document-Based

In the last technique, each document is viewed as a sequence of characters rather than a sequence of words (bags of words) so that each document will be represented as a vector of bi-grams and Dice's Coefficient is used to measure the similarity between the two vectors where *A* and *B* represent the respective numbers of unique N-grams in string one and string two. *C* represents the total number of unique N-grams that are common for both strings being compared.

Given the same two course descriptions in table 1, initially each document is preprocessed as illustrated in table 8:

Table 8: The documents after preprocessing

Document Number	Text (Course description)	After Preprocessing
D1	التكامل المحدود، التكامل بالأجزاء، التكامل بالتعويض، التكامل بالكسور الجزئية، المتتاليات و المتسلسلات	تكامل محدود تكامل أجزاء تكامل تعويض تكامل كسور جزئية متتاليات متسلسلات
D2	التكامل المحدود، التكامل بالأجزاء، التكامل بالتعويض، التكامل بالكسور الجزئية، الإحداثيات القطبية	تكامل محدود تكامل أجزاء تكامل تعويض تكامل كسور جزئية إحداثيات قطبية

Each preprocessed document will be converted into a vector of bi-grams as in table 9 and table 10 respectively:

Table 9: List of bi-grams of document D1

After Preprocessing	Bi-grams
تكامل	{تك، كا، ام، مل، ل ،
محدود	م، مح، حد، دو، ود،د،
تكامل	ت، تك، كا، ام، مل، ل،
أجزاء	أ، أج، جز، زا، اء، ء ،
تكامل	ت، تك، كا، ام، مل، ل ،
تعويض	ت، تع، عو، وي، يض، ض،
تكامل	ت، تك، كا، ام، م، ،ل،
كسور	ك، كس، سو، ور، ر،
جزئية	ج، جز، زئ، ئي، ية، ة،
متتاليات	م، مت، تت، تا، ال، لي، يا، ات، ت،
متسلسلات	م، مت، تس، سل، لس، سل، لا، ات}

Table 10: List of bi-grams of document D2

After Preprocessing	Bi-grams
تكامل محدود تكامل أجزاء تكامل تعويض تكامل كسور جزئية إحداثيات قطبية	{تك، كا، ام، مل، ل م، مح، حد، دو، ود، د ت، تك، كا، ام، مل، ل أ، أج، جز، زا، اء، ء ت، تك، كا، ام، مل، ل ت، تع، عو، وي، يض، ض ت، تك، كا، ام، مل، ل ك، كس، سو، ور، ر ج، جز، زئ، ئي، ية، ة إ، إح، حد، دا، اث، ثي، يا، ات، ت ق، قط، طب، بي، ية}

Then unique Bi-grams are extracted from each vector of the two documents as described in table 11 and table 12 respectively:

Table 11: List of unique bi-grams of document D1

Unique Bi-grams
{تك، كا، ام، مل، ل، م، مح، حد، دو، ود، د ، ت، أ، أج، جز، زا، اء، ء، تع، عو، وي، يض، ض، ك، ك، ،سو، ور، ر، ج، زئ، ئي، ية، ة ، مت، تت، تا، ال، لي، يا، ات، ت، تس، سل، لس، لا}

Table 12: List of unique bi-grams of document D2

Unique Bi-grams
{تك، كا، ام، مل، ل، م، مح، حد، دو، ود، د، ت، أ، أج، جز، زا، اء، ء، تع، عو، وي، يض، ض، ك، كس، سو، ور، ر، ج، زئ، ئي، ية، ة ، إ، إح، دا، اث، ثي، يا، ات، ت ، ق، قط، طب، بي}

The number of unique Bi-grams in D1 is 45 and in D2 is also 45 and the number of common Bi-grams in the two documents is 36. Table 13 shows the common Bi-grams between the two documents.

Table 13: Common bi-grams between D1 and D2

Common Bi-grams
{تك، كا، ام، مل، ل ،
م، مح، حد، دو، ود، د ، ت، أ، أج، جز،
زا، اء، ء ، تع، عو،
وي، يض، ض،
ك، كس، سو، ور، ر ، ج، زئ، ئي،
ية،ة،ياء،ات،ت }

By knowing the number of unique bi-grams in each document and the number of bi-grams that are common to the two documents, the similarity between them can be easily calculated using the formula mentioned above as follows:

(2 * Common Bi-grams) / (Total number of unique Bi-grams)

$$(2 * 36) / (45 + 45) = 0.80$$

Experiments and Results

To evaluate each technique, two sets of documents were used. The first set consist of a collection of 104 different course descriptions which were withdrawn from the course catalogs of different colleges at Yarmouk University, most of which were selected from the College of Information Technology and Computer Sciences and the College of Science. The second set contains 30 course descriptions that were selected from different Jordanian universities to be used as testing cases. Human judgments about the credit transfer of these courses as carried out at Yarmouk University were recorded as shown in Table14. Similarity checking was performed by comparing the testing cases against the course descriptions in the first set.

Table 14: List of testing cases and how they were treated by humans

Testing Case	Human Judgment (Transferred course)	Testing Case	Human Judgment (Transferred course)
Test 1	Arabic 100	Test 16	Physics 101
Test 2	Arabic 102	Test 17	Physics 102
Test 3	Chemistry 101	Test 18	Physics 105, 101
Test 4	Biology 101	Test 19	Physics 106,102
Test 5	Physics 101	Test 20	Biology 101
Test 6	Physics 102	Test 21	Biology 102
Test 7	Computer science 101	Test 22	Biology 105
Test 8	Computer science 250	Test 23	Biology 106
Test 9	Calculus 101	Test 24	Calculus 102
Test 10	Calculus 102	Test 25	Calculus 101
Test 11	Calculus 201	Test 26	Statistics 101

Testing Case	Human Judgment (Transferred course)	Testing Case	Human Judgment (Transferred course)
Test 12	Calculus 203	Test 27	Calculus 241
Test 13	Statistics 201	Test 28	Computer science 333
Test 14	Computer science 100	Test 29	Computer science 100
Test 15	Computer science 101	Test 30	Computer science 101

Each N-gram technique reported in this paper was evaluated by calculating the similarity of each course description from 30 test cases in the second set with all existing courses in the first set of 104 documents and then the course with the highest similarity value is recorded. After that we compare between the course with the highest similarity and the accredited course of the test case in table14. If they are matched then the accreditation done by the technique is correct, otherwise it is not. By doing that, the accuracy for each technique can be calculated by dividing the number of test cases that are accredited correctly on the total number of test cases.

Word-Based Using Dice's Similarity Coefficient

The threshold value used for determining if two words are similar was 0.75. As table 15 indicates, the number of courses that are correctly handled is 24 courses, which gives an accuracy rate of 0.8 (i.e., 24/30=80%).

Table15: Results of the word-based N-gram technique using Dice's Coefficient

Coming Courses	System judgment	Coming Courses	System judgment
Test 1	0.15	Test 16	0.37
Test 2	0.16	Test 17	0.65
Test 3	0.31	Test 18	0.55
Test 4	0.17	Test 19	0.64
Test 5	0.48	Test 20	0.36
Test 6	0.9	Test 21	0.125
Test 7	Fail(CS 352)	Test 22	0.38
Test 8	0.43	Test 23	0.17
Test 9	0.52	Test 24	0.64
Test 10	0.52	Test 25	0.45
Test 11	0.68	Test 26	0.57
Test 12	0.5	Test 27	0.49
Test 13	0.36	Test 28	Fail (CS 433)
Test 14	Fail(MIS250)	Test 29	Fail(MIS250)
Test 15	Fail(CIS 101)	Test 30	Fail(CIS 101)

Word-Based Using Cosine Similarity

According to the results shown in table 16, the number of courses that are correctly treated is 26 courses. So the accuracy rate of this technique is 0.87 (i.e., 26/30=87%).

Table16: Results of the word-based N-gram technique using Cosine similarity

Coming Courses	System judgment	Coming Courses	System judgment
Test 1	0.3	Test 16	0.5
Test 2	Fail(CS470)	Test 17	0.8
Test 3	0.55	Test 18	0.66
Test 4	0.48	Test 19	0.46
Test 5	0.52	Test 20	0.56
Test 6	0.93	Test 21	0.45
Test 7	Fail(CS352)	Test 22	0.29
Test 8	Fail(CS352)	Test 23	0.35
Test 9	0.38	Test 24	0.66
Test 10	0.65	Test 25	0.44
Test 11	0.64	Test 26	0.66
Test 12	0.65	Test 27	0.68
Test 13	0.42	Test 28	0.62
Test 14	0.35	Test 29	0.35
Test 15	Fail(CIS10)	Test 30	Fail(CIS10)

Whole Document-Based

According to the results shown in table 17, the number of courses that are correctly accredited is 21 courses. Hence, the accuracy rate of this technique is 0.7 (i.e., 21/30=70%).

Table17: Results of the whole document-based N-gram technique using Cosine similarity

Coming Courses	System judgment	Coming Courses	System judgment
Test 1	Fail CS 376	Test 16	0.57
Test 2	Fail CALC102	Test 17	0.68
Test 3	0.54	Test 18	0.65
Test 4	0.50	Test 19	0.58
Test 5	0.56	Test 20	0.60
Test 6	0.79	Test 21	0.54

Coming Courses	System judgment	Coming Courses	System judgment
Test 7	Fail(CS25)	Test 22	0.51
Test 8	0.57	Test 23	Fail(CHE105)
Test 9	Fail(Calc21)	Test 24	0.66
Test 10	0.58	Test 25	0.56
Test 11	0.79	Test 26	0.67
Test 12	0.60	Test 27	0.63
Test 13	0.58	Test 28	Fail(CS 433)
Test 14	Fail(MIS 482)	Test 29	Fail(CIS103)
Test 15	Fail(CIS 227)	Test 30	0.49

Putting the results of the three techniques as shown in table 18, we can see that the word-based technique that uses Cosine Similarity provides better accuracy rates than the other two N-gram matching techniques. It worth nothing that the similarity values of the first technique are smaller than the similarity values of the other two techniques. It is important to note also that most of the courses that all techniques failed to accredit are computer science courses. This is because these courses have general descriptions especially first year courses such as "CS 100". On the other hand, the science faculty courses have similar descriptions in the different universities, because of that they are correctly accredited by all the techniques.

Low accuracy of the third technique, compared with the others, does not necessary mean that it is not a good technique, because it has a chance to be a good one when it is used with trigram or more.

It should be mentioned here that we didn't consider evaluation of performance factors due to the fact that the whole process is not time consuming. Once indexers are built, the process of comparing documents is relatively short and hence evaluating the impact on time was insignificant.

Table 18: Summary of the accuracy results for the three techniques

Technique	Accuracy
Word-Based Using Dice's Similarity Coefficient	80%
Word-Based Using Cosine Similarity	87%
Whole Document-Based	70%

Conclusion and Future Work

In this paper, we focused on measuring documents' similarity for course credit transfer between Jordanian universities using Arabic course descriptions. Three N-gram-based matching techniques have been investigated and compared with human judgments, with the objective to evaluate which technique provides better results within an acceptable threshold value.

The analysis of the results indicates that the word-based N-gram technique using Cosine Similarity provides better accuracy rates than the word-based technique and than the whole document-based N-gram technique that use Dice's Coefficient.

The results of this investigation show that N-gram document matching techniques can be applied to automate the matching process of course descriptions for credit transfer between universities within an accuracy level that goes beyond 80%. However, it important to note that similar courses at various universities seem to have similar descriptions, because they represent commonly adopted requirements at the national level.

This study applied N-gram approach to document matching at two levels: word level and whole document level. The same approach can be applied also at phrase level. In addition, all the techniques discussed in the paper applied Bi-gram computation, applying Tri-gram computation might improve all or some of the strategies used, because Tri-grams put more constraint on the matching process than Bi-grams.

مطابقة الوثائق النصية العربية باستخدام أساليب المقاطع النصية: دراسة تطبيقية على وصف المساقات لغايات احتساب الساعات المعتمدة بين الجامعات

محمد عارف الرمحي و سليمان حسين مصطفى

ملخص

لقد تمت دراسة موضوع التشابه بين النصوص لمدة طويلة نظراً لأهميته في العديد من التطبيقات في مجال معالجة اللغات الطبيعية والمجالات المتصلة بها مثل البحث عن النصوص على الانترنت. ومن بين هذه التطبيقات تحديد مدى التشابه بين وصف المساقات في المجال الأكاديمي الواحد لغايات احتساب الساعات المعتمدة في حالة الإنتقال من جامعة إلى أخرى، وهو موضوع هذه الدراسة. وقد تمت في هذه الدراسة مقارنة ثلاثة أساليب مختلفة باستخدام المقاطع الزوجية (bi-gram) لحساب درجة التشابه بين نصين أو أكثر من نصوص وصف المساقات باللغة العربية. الأسلوب الأول يستخدم نموذج المصفوفة لتمثيل النص بحيث يكون كل مقطع زوجي مصحوباً بوزن يعكس أهميته في النص. ومن ثم يستخدم معيار كوساين (Cosine) لحساب التشابه بين المصفوفات. الأسلوبان الآخران هما: التقييم المبني على الكلمة الواحدة والتقييم المبني على كامل الوثيقة، حيث استخدم في كليهما مقياس دايس (Dice) لحساب التشابه بين زوجين من النصوص. وقد أشارت النتائج إلى أن الأسلوب الأول أظهر أداءً أفضل من الأسلوبين الآخرين عند مقارنة كل منهما بما يقوم به الإنسان لتحديد التشابه.

References

[1] Miller E., Shen D., Liu J., and Nicholas C., 'Performance and Scalability of a Large-Scale N-gram Based Information Retrieval System', *Journal of Digital Information*, 1(5), 2000**.**

[2] Corley C. and Mihalcea R., 'Measuring the Semantic Similarity of Texts', *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, 2005.

[3] Baeza-Yates R. and Ribeiro-Neto B. (eds). *Modern Information Retrieval.* Addison Wesley/ACM Press, New York, 1999.

[4] Fung B. C., Wang K. and Ester M., 'Hierarchical Document Clustering Using Frequent Itemsets', *Proceedings of the SIAM International Conference on Data Mining*, 2003

[5] Mustafa S. H. and AL-Radaideh Q. A., 'Using N-grams for Arabic Text Searching', *Journal of The American Society for Information Science and Technology - JASIS*, 55(11), 2004

[6] Kosinov S., 'Evaluation of N-Grams Conflation Approach in Text-Based Information Retrieval', *Proceedings of International Workshop on Information Retrieval*, 2001.

[7] Frakes W. B. and Baeza-Yates R. A. (eds). *Information Retrieval: Data Structures and Algorithms.* Prentice-Hall, 1992.

[8] Zwa A., Ebert D. S. and Miller E. L., 'Multiresolution Document Analysis with Wavelets', *Proceedings of the 1996 Conference on Information and Knowledge Management (CIKM '96) Workshop on New Paradigms in Information Visualization and Manipulation*, Nov. 1996.

[9] Cavnar W. B. and Gillies A. M. 'Data Retrieval and the Realities of Document Conversion', *Proceedings of the First Annual Conference on the Theory and Practice of Digital Libraries, College Station (Texas)*, June 1994

[10] Hirsch L., Saeedi M., and Hirsch R., 'Evolving Rules for Document Classification', *European Conference on Genetic Programming-EuroGP*, 2005.

[11] Cavnar W. B., 'Using an N-Gram-Based Document Representation with a Vector Processing Retrieval Model', *Text REtrieval Conference - TREC*, 1994**.**

[12] Haslinger A., 'Data Matching for the Maintenance of the Business Register of Statistics', *Austrian Journal of Statistics*, 33(1&2), 2004.