Mining Textual Chinese News Articles

Mohammad Alkhaleefah[1], Shadi Alian[2] and Orabe Almanaseer[3]

[1] Graduate Institute of Networking & Multimedia, National Taiwan University
Taipei, 10617, Republic Of China (Taiwan)

[2] Department of Computer Information Systems, The University of Jordan
Aqaba, 77110, Jordan

[3] Department of Business Information Technology, The University of Jordan
Aqaba, 77110, Jordan

Abstract

The purpose of this paper is to implement a system for analyzing unlabeled news documents that are written in Chinese Language to retrieve at most (100) relevant documents from document collection in response to a specific query, then the retrieved documents are grouped by authors. To reach this purpose, the implementation of the designed system framework consists of two phases. In phase 1, an Information Retrieval system is implemented, while in phase 2, a classification system is implemented. Each phase consists of several steps, and different algorithms, tools and techniques are used in each step. The system results in each phase are evaluated according to evaluation schema.

Keywords: *Chinese news documents analysis, Information Retrieval, Documents Classification.*

1. Introduction

Nowadays, an increasing number of newspapers have a place on the internet and most of them provide an electronic version of their printed articles. The reason for a newspaper doing this is to attract readers as the number of Internet users has rapidly increased over the past decades. In fact, the overwhelming volume of textual news articles available in digital form has made it difficult to find information of interest to specific readers or be useful for particular purposes. In other words, this amount of text is simply too large to read and analyze easily. Accordingly, the need for automated extraction of useful knowledge from huge amounts of textual data in order to assist human analysis is fully apparent [1]. Furthermore, this massive amount of textual articles has provided computer and information system specialists with many opportunities for research in the area of Natural Language Processing (NLP), Text Mining, and machine learning Techniques. For the purpose of this paper we will focus on the areas of NLP, Information Retrieval and Classification.

2. Background

2.1 Natural Language Processing

Natural language processing (NLP) is the attempt of getting computers to understand the languages that people use for everyday communication in order to change the way that we interact with them [2]. Moreover, NLP is the process of extracting a fuller meaning representation from text by using information from a formal grammar and a lexicon, the resulting information is then interpreted semantically to figure out what was said (who did what to whom, when, where, how and why). NLP has various techniques that typically make use of linguistic concepts, It includes techniques like word stemming (removing suffixes), lemmatization (replacing an inflected word with its base form), multiword phrase grouping, synonym normalization, part-of-speech (noun, verb, adjective, etc.), word-sense disambiguation, anaphora resolution (what previous noun does a pronoun or other back-referring phrase correspond to), and role determination (e.g. subject and object) [3].

2.2 Information Retrieval (IR)

Information retrieval deals with the structure, analysis, organization, storage, searching, and the main task of information retrieval is to retrieve relevant documents in response to a query [4]. Fig. 1 illustrates the objectives of information retrieval of documents, where (a) a general description is given of the query, (b) the document collection is searched, and (c) a subset of relevant documents is returned [5].

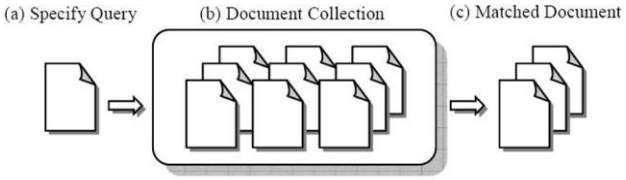

Fig. 1 Information Retrieval Objectives.

2.3 Classification

Perhaps the most common theme in analyzing data is text classification. The main task of text classification is to classify a collection of document into a given set of categories. And when a new document is presented, the objective is to place this document in the appropriate category. Feldman and Sanger [6] point out that there are two main approaches to text classification. The first is the knowledge engineering approach in which the expert's knowledge about the categories is directly encoded into the system. The other is the machine learning approach in which a general inductive process builds a classifier by learning from a set of pre-classified documents, and the characteristics of the categories. However, the most of recent work on classification is concentrated on machine learning approach. The advantages of this approach over the knowledge engineering approach are a very good effectiveness, considerable saving in terms of costs, time, expert labor power, and straightforward portability to different domains [7].

3. Problem Formulation

The purpose of this project lies on Information Retrieval and Text Classification techniques in order to design a system for textual analysis to automatically analyze structure of unlabeled news documents that are written in Chinese Language. The main task of this system is to retrieve at most (100) relevant documents from document collection in response to a specific query, then the retrieved documents are grouped by authors.

4. Methodology

In this project, a proposal framework based on IR and Classification techniques is adopted in order to design and develop a system for retrieving and classifying Chinese news documents (see Fig. 2).

The overall system is implemented through two phases. In phase 1, the IR System is designed and implemented to retrieve news documents by a query, while in phase 2, our task is to implement a classifier to classify the retrieved news documents by agency. Then the systems results are evaluated according to evaluation schema.

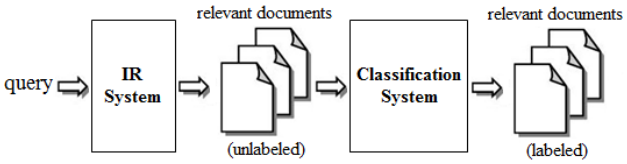

Fig. 2 The System Framework

5. Evaluation Schema

For the IR system in Phases 1, the classic IR notations of precision and recall are adapted to evaluate the performance of the system result. The Precision (P) is the fraction of retrieved documents that are relevant, and the Recall (R) is the fraction of relevant documents that are retrieved, see the following equations.

$$\text{Precision} = \frac{\#(\text{relevant items retrieved})}{\#(\text{retrieved items})} = P(\text{relevant}|\text{retrieved}) \quad (1)$$

$$\text{Recall} = \frac{\#(\text{relevant items retrieved})}{\#(\text{relevant items})} = P(\text{retrieved}|\text{relevant}) \quad (2)$$

However, the performance measures (Precision and Recall) may be misleading when examined alone. Therefore, another measure called precision at n (P@n) is considered for evaluating our system, whereas P@5 & P@10 results returned by the system is calculated [8].

For the Classification system in Phases 2, the decision made by the classifier can be represented in a structure known as a global contingency table (see Table 1) [9] [10].

Table 1: The Global Contingency Table

Category set $C=\{c_1, c_2, c_3, \ldots, c_3\}$		Actual Results	
		TRUE	*FALSE*
Classifier Results	*TRUE*	$TP=\sum_{i=1}^{m} TP_i$	$FP=\sum_{i=1}^{m} FP_i$
	FALSE	$FN=\sum_{i=1}^{m} FN_i$	$TN=\sum_{i=1}^{m} TN_i$

This table contains information about actual and predicted classification done by a classification system, and it has four main categories which are:

- True positive (TP) = positive examples correctly labeled as positive.
- False positive (FP) = negative examples incorrectly labeled as positive.
- True negative (TN) = negative examples correctly labeled as negative.
- False negative (FN) = positive examples incorrectly labeled as negative.

The global contingency table is used for calculating the micro-average measurements (see Table 2). For calculating the macro-average measurement first the micro-average is calculated and then will be divided by total number of categories (see Table 3) [10].

Table 2: The Micro-average Measures

Precision	$\pi = \frac{TP}{TP + FP} = \frac{\sum_{i=1}^{\|C\|} \mathrm{TP}_i}{\sum_{i=1}^{\|C\|} (TP_i + FP_i)}$
Recall	$\rho = \frac{TP}{TP + FN} = \frac{\sum_{i=1}^{\|C\|} \mathrm{TP}_i}{\sum_{i=1}^{\|C\|} (TP_i + FN_i)}$

Table 3: The Macro-average Measures

Precision	$\pi = \frac{\sum_{i=1}^{\|C\|} \pi_i}{\|C\|} = \frac{\sum_{i=1}^{\|C\|} \frac{\mathrm{TP}_i}{\mathrm{TP}_i + FP_i}}{\|C\|}$
Recall	$\rho = \frac{\sum_{i=1}^{\|C\|} \rho_i}{\|C\|} = \frac{\sum_{i=1}^{\|C\|} \frac{\mathrm{TP}_i}{\mathrm{TP}_i + FN_i}}{\|C\|}$

As mentioned previously, the contents of Tables 3 are used to extract a number of measures to evaluate the performance of a text categorization system. A common evaluation strategy is to consider classification accuracy or its complement error rate [11]. Accuracy refers to the proportion of correctly assigned documents to the total number of documents assigned and error is the proportion of incorrectly assigned documents to the total number of documents assigned. The better the classifier the higher would be its accuracy [12]. Based on contingency table, the Accuracy (Acci) and Error rate (Erri) can be calculated as shown in Table 4.

Table 4: The Accuracy (Acci) and Error rate (Erri)

$Acc_i =$	$\frac{TP_i + TN_i}{TP_i + FP_i + TN_i + FN_i}$
$Err_i =$	$\frac{FP_i + FN_i}{TP_i + FP_i + TN_i + FN_i}$

6. System Implementation

In this project, the overall system implementation process is based on the proposed system framework. The process consists of many different courses of actions including document collection, text-document preprocessing, document representation and dataset generation, classifier learning, and system evaluation. In order to accomplish the overall system design process, different tools and algorithms have been used in each step. The Java programming language has been selected to implement the coding part for all algorithms in the system. The following sections address the details of the process and methods associated with each step.

6.1 Document Collection

The data set used in this project contains of (102005) unlabeled news documents that are stored in (XML) format. These documents are written in Chinese language by three different agencies. For evaluation purpose another extra document contains ground truth information (document id, author and release date) for each document in the collection.

6.2 Text Processing

As mentioned previously, the news document collection is written in Chinese language. We have to take into account that Chinese is standardly written without spaces between words. Therefore, the text document processing consists of two main steps including: segmentation and normalization. However, all XML tags are removed from documents before starting processing them.

In segmentation process, for each document in the collection, the Chinese text is tokenized into tokens (individual words) using Stanford-Word-Segmenter tools. These tools are a Java implementation of the CRF-based Chinese Word Segmenter [13]. We decided to include two models with two different segmentation standards, Chinese Penn Treebank (CTB) standard and Peking University standard (PKU) [14].

In the normalization process, all the punctuations and all non-Chinese single characters are removed from the list of

terms created in the previous step. This will reduce the storage size used for representing these terms in the main memory.

6.3 Document Representation

For efficient retrieval, an inverted index is used to index our documents. The inverted index is the essential data structure used in information retrieval systems.
In this System, an index is created as an object containing all terms in the documents collection, whereas each term in our index is an object holding the document frequency (df) of this term, and pointing to a posting list that contains the id of the document where the term occurs and the term frequency (tf) in each document. (see Fig. 3).

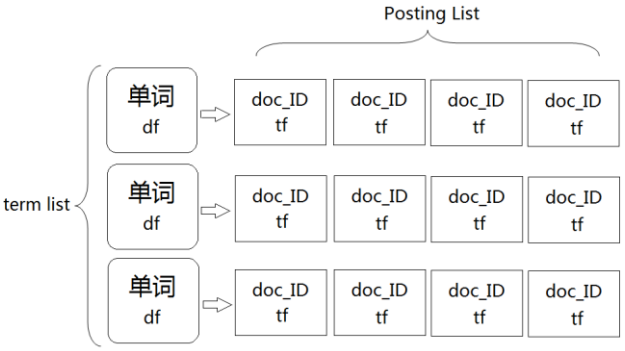

Fig. 3 Document Representation (Inverted Index)

6.4 Phase 1, IR system

As a result from previous step, all text documents (unstructured data) have been converted into structured data in a serialized object. Now we can easily retrieve the relevant documents for any one-term-query by retrieve its posting list. We have implemented an algorithm to handle queries contain more than one term, this algorithm finds the union between the posting lists for all terms in the query.

As mentioned previously, the main task of this system is to retrieve at most (100) relevant documents from documents collection in response to a specific query. Therefore, we have to give a rank for each retrieved document. However, another algorithm is implemented to assign a rank to the retrieved documents based on tf-idf weighting along with the vector space model for scoring [15]. The following equations have been used to score documents based on given query.

$$\text{Score}(q,d) = \sum_{t \in q} \text{tf-idf}_{t,d}. \tag{3}$$

$$\text{tf-idf}_{t,d} = \text{tf}_{t,d} \times \text{idf}_t. \tag{4}$$

$$\text{idf}_t = \log \frac{N}{\text{df}_t}. \tag{5}$$

The major Information Retrieval system process is based on the previous equations and is fully implemented in Java according to the following algorithm [15].

```
COSINESCORE(q)
1   float Scores[N] = 0
2   Initialize Length[N]
3   for each query term t
4   do calculate w_{t,q} and fetch postings list for t
5      for each pair(d, tf_{t,d}) in postings list
6      do Scores[d] += wf_{t,d} × w_{t,q}
7   Read the array Length[d]
8   for each d
9   do Scores[d] = Scores[d]/Length[d]
10  return Top K components of Scores[]
```

6.4.1 IR System Testing and Results

As a result of this phase, at most 100 documents are retrieved for each query. For the lack of space, it is not possible to show the entire result of this phase. Therefore, the result of query 1 is illustrated as a sample in Table 5.

Table 5: The Relevant Documents Retrieved for Query 1
(IR System Sample Result)

1
56877 77894 73046 21328 43742 62650 51265 77154 97844 65640 73538 7333 52273 11259 35554 76592 78064 17035 55836 68942 39334 65768 81464 40629 78300 9558 35524 86464 99535 91914 100308 13522 41337 83642 34257 20904 45852 62788 45670 66790 92373 925 5697 72725 8191 98140 101621 67459 60292 67260 90028 31746 28867 1205 2488 62963 86764 35603 47183 92170 7873 31335 1382 59710 98743 531 74130 19705 99875 83803 90251 30305 4036 47138 64980 31558 39484 80470 64077 605 427 70053 25424 31051 47114 28597 99358 44987 40407 56181 49912 30514 1102 27874 98221 55155 1815 63923 55017

6.4.2 IR System Evaluation

The IR system performance is evaluated according to Precision-Recall evaluation criteria that are induced from the contingency table based on IR system results. We have implemented a Java application that takes these results, in order to calculate the average of Precision-Recall and (P@5 and P@10) for the 20 queries given to the system (see Table 6).

Table 6: The IR System Evaluation

Recall	Precision
0.00	0.831363636363636
0.10	0.693203168137631
0.20	0.590004733641134
0.30	0.505625749816590
0.40	0.458497377440170
0.50	0.365829471838338
0.60	0.313554213795717
0.70	0.143064322503194
0.80	0.111995554035567
0.90	0.031617647058824
1.00	0.00

P@5	0.660
P@10	0.655

6.5 Phase 2, Classification System

The main goal in this phase is to build a classifier that can classify all the documents are retrieved from the IR system built in phase 1. For the classification System in this phase, a dataset has been generated by using the ground truth document which contains information (document id, author and release date) for each document in the collection except the retrieved documents (phase 1, result). The dataset is created as a one-dimensional array of three elements or classes (0, 1, and 2), we have randomly chosen 90% of the dataset as the training set and 10% as the testing set. Each element in the two arrays (training and testing sets) is a hashmap object containing all documents IDs related to a specific class along with the released data for each document. The main reason for dividing the dataset is that the training set is used to build the classifier and then the testing set is used to evaluate the performance of the classifier. (see Fig. 4)

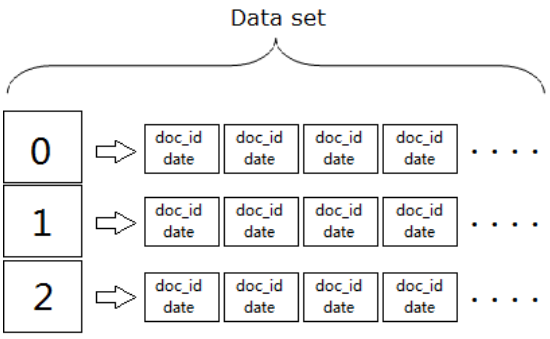

Fig. 4 Training and Testing Datasets Structure

The relationship between the contents of each class (0, 1, and 2) in the training set of documents is learned through an algorithm based on Naive Bayes text classification model for probability estimation. The following equation is the most implementations of Naive Bayes multinomial model [16].

$$c_{\text{map}} = \arg\max_{c \in \mathbb{C}} [\log \hat{P}(c) + \sum_{1 \le k \le n_d} \log \hat{P}(t_k|c)]. \quad (6)$$

For calculating the parameter P^(c), we have used the following equations [16]:

$$\hat{P}(c) = \frac{N_c}{N}, \quad (7)$$

Where Nc is the number of documents in class c and N is the total number of documents. We calculate the conditional probability P^(t|c) as the relative frequency of term t in documents belonging to class c:

$$\hat{P}(t|c) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct'}}, \quad (8)$$

Where Tct is the number of occurrences of t in training documents from class c, including multiple occurrences of a term in a document. The major learning and classification process is fully implemented in Java programming language according to the following algorithm [16].

```
TRAINMULTINOMIALNB(ℂ, 𝔻)
 1  V ← EXTRACTVOCABULARY(𝔻)
 2  N ← COUNTDOCS(𝔻)
 3  for each c ∈ ℂ
 4  do N_c ← COUNTDOCSINCLASS(𝔻, c)
 5     prior[c] ← N_c / N
 6     text_c ← CONCATENATETEXTOFALLDOCSINCLASS(𝔻, c)
 7     for each t ∈ V
 8     do T_ct ← COUNTTOKENSOFTERM(text_c, t)
 9     for each t ∈ V
10     do condprob[t][c] ← (T_ct+1) / Σ_t'(T_ct'+1)
11  return V, prior, condprob

APPLYMULTINOMIALNB(ℂ, V, prior, condprob, d)
1  W ← EXTRACTTOKENSFROMDOC(V, d)
2  for each c ∈ ℂ
3  do score[c] ← log prior[c]
4     for each t ∈ W
5     do score[c] += log condprob[t][c]
6  return arg max_c∈ℂ score[c]
```

6.5.1 Classification System Testing and Result

Due to the huge volume of the retrieved documents, it is not possible to show the entire result. Hence, a sample of the result is illustrated in Table 7.

Table 7: The Relevant Documents Retrieved for Query 1 (Classification System Sample Result)

1
56877-0 77894-0 73046-2 21328-2 43742-0 62650-0 51265-0 77154-2 97844-2 65640-0 73538-1 7333-0 52273-0 11259-1 35554-1 76592-1 78064-2 17035-0 55836-1 68942-1 39334-1 65768-2 81464-1 40629-2 78300-1 9558-1 35524-2 86464-0 99535-2 91914-2 100308-2 13522-0 41337-1 83642-0 34257-2 20904-1 45852-1 62788-0 45670-0 66790-1 92373-2 925-1 5697-1 72725-2 8191-2 98140-1 101621-2 67459-2 60292-0 67260-0 90028-0 31746-1 28867-2 1205-2 2488-2 62963-2 86764-1 35603-2 47183-1 92170-1 7873-1 31335-0 1382-2 59710-2 98743-2 531-2 74130-1 19705-2 99875-1 83803-0 90251-0 30305-0 4036-0 47138-0 64980-0 31558-2 39484-2 80470-2 64077-2 605-1 427-2 70053-0 25424-0 31051-1 47114-2 28597-0 99358-0 44987-2 40407-0 56181-1 49912-2 30514-0 1102-1 27874-1 98221-1 55155-0 1815-0 63923-0 55017-2

6.5.2 Classification System Evaluation

The Classification system performance is evaluated to different evaluation criteria that are induced from the contingency table given by Naive Bayes classifier. We have implemented a Java application that takes the classifier results, in order to calculate the Precision-Recall, Accuracy-Error, and F-measure using the Micro-Averaging and the Macro-Averaging measurements (see Tables 8 and 9).

Table 8: Evaluation using Micro-Averaging

Evaluation using Micro-Averaging	
Recall:	0.68620000000
Precision:	0.68620000000
Accuracy:	0.79080000000
Error:	0.20920000000
F-measure:	0.68620000000

Table 9: Evaluation using Macro-Averaging

Evaluation using Macro-Averaging	
Recall:	0.68484186940
Precision:	0.68651615065
Accuracy:	0.79080000000
Error:	0.20920000000
F-measure:	0.68557772182

7. Challenges

One of the challenges we have faced in phase 1 is the segmentation accuracy or ambiguity. Native Chinese speakers will sometimes disagree about the proper segmentation for a sentence, as in many cases a Chinese character can be either a term by itself or part of a compound terms.

Another challenge regarding the segmentation part is that the dictionary we used in our segmentation algorithm is not up to date. Therefore, the segmentation processing might be inaccurate.

8. Conclusion

In this project, a framework was proposed based on IR and Classification techniques in order to design and develop a system for retrieving and classifying Chinese news documents. The system was implemented through two phases. In phase 1, the IR System is designed and implemented to retrieve news documents by a query, while in phase 2, classifier was implemented to classify the retrieved news documents by authors.

The systems results of phase 1 are evaluated according to evaluation schema Recall and Precision (P@5, P@10) of the retrieved results (P@5 is 0.660 and P@10 is 0.655). In phase 2, we have implemented a Java application that takes the classifier results, in order to calculate the Precision-Recall, Accuracy-Error, and F-measure using the Micro-Averaging and the Macro-Averaging measurements. The accuracy of both measurements is 0.79.

References

[1] H. Karanikas and B. Theodoulidis, *Knowledge Discovery in Text and Text Mining Software*. Technical Report Centre for Research in Information Management (CRIM). Manchester, UK: Department of Computation, UMIST, 2002.

[2] Gobinda G. Chowdhury, Natural Language Processing, *Annual review of information science and technology*, vol. 37, pp. 51-89, 2003.

[3] A. Kao, and Steve R. Poteet, *Natural Language Processing and Text Mining*. London, Springer Science & Business Media, 2007.

[4] R. Baeza-Yates and . B. Ribeiro-Neto, *Modern Information Retrieval: The Concepts and Technology Behind Search*, Addison Wesley, 2011.

[5] M. W. Sholom, I. Nitin, Z. Tong, and D. Fred, *Text Mining: Predictive Methods for Analysing Unstructured Information*. New York: Springer-Verlag, 2005.

[6] R. Feldman and J. Sanger, *The Text Mining Handbook: Advanced Approaches in Analysing Unstructured Data*. London: Cambridge University Press, 2007.

[7] F. Sebastiani, Machine Learning in Automated Text Categorisation, *ACM Computing Survey (CSUR)*, vol. 34, no. 1, pp. 1-47, 2002.

[8] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, Addison-Wesley, 1999.

[9] R. Kohavi and P. Foster, Glossary of terms, *Machine Learning*, vol. 30, no. 2-3, pp. 271-274, 1998.

[10] F. Sebastiani, A Tutorial on Automated Text Categorisation, Proc. *1st Argentinean Symposium on Artificial Intelligence (ASAI)*, pp. 7-35, 1999.

[11] M. Goadrich, L. Oliphant, and J. Shavlik, Gleaner: Creating Ensembles of First-Order Clauses to Improve

Recall-Precision Curves, *Journal of Machine Learning*, vol. 64, no. 1-3, pp. 231-261, 2006.

[12] Wikipedia, Evaluation Methods in Text Categorisation, http://datamin.ubbcluj.ro/wiki/index.php/Evaluation_methods_in_text_categorization, 2011.

[13] T. Huihsin , P. Chang, G. Andrew, D. Jurafsky and C. Manning, A Conditional Random Field Word Segmenter, *In Fourth SIGHAN Workshop on Chinese Language Processing*, 2005.

[14] C. Manning, D. Jurafsky and P. Liang, *The Stanford Natural Language Processing Group*, [Online]. Available: http://nlp.stanford.edu/-software/segmenter.shtml. [Accessed 30 January 2016].

[15] C. Manning, D. Jurafsky and P. Liang, *Computing vector scores*, [Online]. Available: http://nlp.stanford.edu/IR-book/html/htmledition/computing-vector-scores-1.html. [Accessed 30 January 2016].

[16] C. Manning , P. Raghavan and H. Schütze, *Introduction to Information Retrieval*, Cambridge University Press, 2008.

Mohammad Alkhaleefah received the Msc degree from the University of Manchester, United Kingdom, in 2008. He is a PhD candidate in Graduate Institute of Networking and Multimedia at National Taiwan University. He is a lecturer with the Faculty of Rahma College at the Al-Balqa Applied University, Jordan. His research interests include e-government, e-services, Web2.0, knowledge management, and data mining.

Shadi Alian received the B.Sc. degree in Computer Science from Yarmouk University, Irbid, Jordan, in 2004, then he was awarded a merit-based scholarship to continue his M.Sc. degree in Computer Science from Northeastern Illinois University, Chicago, Illinois, USA, in 2007. He is, at present, lecturer at The University of Jordan, Aqaba, Jordan, since 2010. His research interests include Multi-agent algorithms, Mutation testing, automatic test data generation and natural language processing.

Orabe Almanaseer received the Msc degree from the University of Manchester, United Kingdom, in 2008. He is a lecturer with the Faculty of information Technology and Systems at the University of Jordan, Aqaba, Jordan. His research interests include Web 2.0 and social media for e-government, ICT for development, and document interoperability. Mr. Almanaseer, is an IEEE Student Member.