

# OPTIMISING THE EMAIL KNOWLEDGE EXTRACTION SYSTEM TO SUPPORT KNOWLEDGE WORK

Tedmori, Sara, Department of Civil and Building Engineering, Loughborough University, Loughborough University, Loughborough, Leicestershire, UK, LE11 3TU, S.M.J.Tedmori@lboro.ac.uk

Jackson, Thomas, Research School of Informatics, Loughborough University, Loughborough University, Loughborough, Leicestershire, UK, LE11 3TU, T.W.Jackson@lboro.ac.uk

Bouchlaghem, Dino, Department of Civil and Building Engineering, Loughborough University, Loughborough University, Loughborough, Leicestershire, UK, LE11 3TU, N.M.Bouchlaghem@lboro.ac.uk

## Abstract

*Although employees' expertise has for some time been seen as a vital knowledge asset in organisations, it is only lately that it started to attract researchers' attention. As a result, interest in automated systems that aim at enhancing the visibility and traceability of employees with particular expertise is growing. This research focuses on one critical everyday organisational business tool - email, as an information source to help locate employees with particular expertise within the organisation. This paper presents the process for keyphrase extraction from email messages. The process uses machine learning to tag new text by its part of speech, then extracts keyphrases purely based on part-of-speech (POS) tags that surround these phrases. The system has been evaluated using three datasets. Results show that the use of the linguistic tool, WordNet, improves to some extent the precision, recall, and f-measure metrics. The goal of this work is to advance our understanding of what may (or may not) be effective in extracting information from email to help identify experts.*

*Keywords: Email, Expertise identification, Knowledge management, Keyphrase extraction, Performance Measurement.*

# 1 INTRODUCTION

In an age of information abundance, employees are often overwhelmed with information that varies in both quality and format. Selecting from the information sources available can be both daunting and time consuming.

What's the solution? People are valuable sources of information. Research has consistently found that people often seek information by asking other people even when they have access to reservoirs of information such as the Internet and libraries (Bannon 1986, Kraut et al. 1995). People naturally rely on the experience and opinions of others, seeking recommendations from people who are familiar with the choices they face, who have been helpful in the past, whose perspectives they value, or who are recognised experts (Terveen et al. 2001). Therefore, finding the right person to turn to when you have an information need is important.

Expertise locating systems have emerged as an attempt to decrease workload on the key people in organisations whose task is to make referrals (McDonald et al. 2000) and to support people in benefiting from each other's experience. This paper explores the development of the expertise recommender system EKE (short for *Email Knowledge Extraction*). The successful development of EKE depends largely on the performance of a subsystem that uncovers interest areas by picking out keyphrases from employees' email messages. Several automatic keyphrase extraction techniques have been proposed in previous studies (Turney 1999, Frank et al. 1999, Barker et al. 2000). However, these techniques are limited because the extracted phrases are often ill formed or inappropriate (Medelyan et al. 2006).

The work is a continuation of the research started earlier by the authors (Tedmori et al. 2006), towards the development and optimisation of a keyphrase extraction system from email messages to aid in determining who knows what within the organisation. The challenge is to extract relevant phrases, whilst minimising non-relevant phrases. Relevant phrases are those that disclose skills and experiences traded in the exchange of emails such as technical expertise, management skills, industry knowledge, education and training, work experience, professional background, knowledge in subject areas, etcetera. The system has to achieve both high precision and recall, two standard measures of performance that are primarily used in information retrieval. This work is a direct response to the need to improve further the efficiency of EKE through adding a secondary process (WordNet) to improve precision and recall.

## 2 OVERVIEW OF EKE

The use of email has grown, making it the most intensively used knowledge work tool. Over the last three years the authors of this paper have been trying to develop a solution to support knowledge work. The tool which has been named EKE, mines information contained in employees' emails. EKE automatically finds interest areas by picking out keyphrases from employees' email messages. EKE is designed in such a way that it is tightly integrated into the employees' email client and fits well into the work they do in a natural manner.

How does it work? EKE intercepts email messages before they are sent to the remote email server and retrieves the email content. An Outlook plug-in developed by the authors is used for this task. As soon as the email content is captured, the plug-in issues an http request to a web service passing to it the email content, along with the sender's email address. On the server, a web service runs extracting keyphrases from the email content and storing them in a temporary buffer. At a certain point in time, a server side application collates all of the extracted keyphrases and displays them to the user for their approval. For ethical and privacy reasons, the user has to specify the extracted keywords as private or public and rank them using a scale of three to denote their expertise in that field (e.g. basic knowledge, working knowledge, or expert). The keywords accepted by the user are then stored on a main database

on the server. The keywords in the database can then be retrieved based on user's queries. The result returned is a list of experts in the organisation ranked by their suitability to answer the user's query.

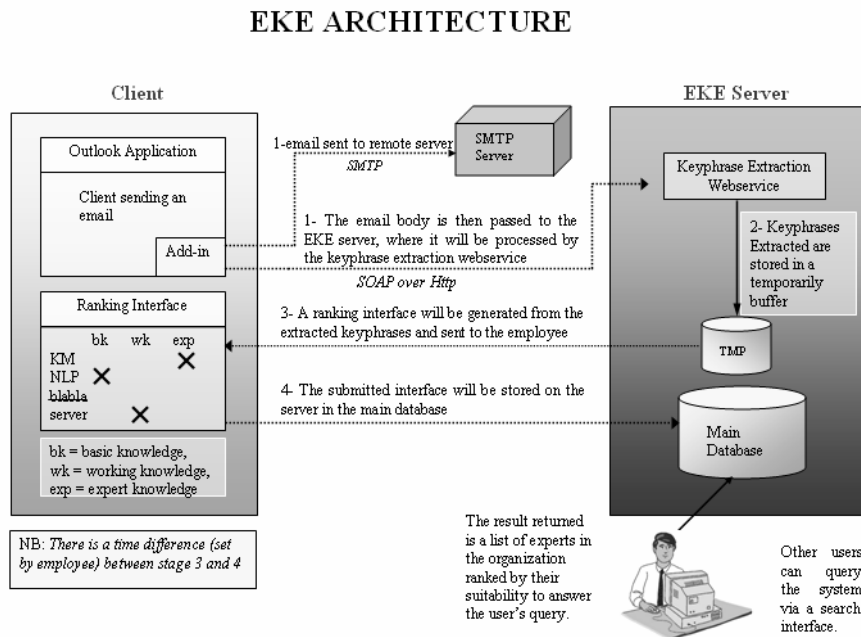


Figure 1. EKE Generic Architecture

### 3 BACKGROUND

The purpose of this section is to provide a brief background and information on roles of keyphrases. This section will also outline the existing keyphrase extraction techniques.

#### 3.1 Keyphrases

Keywords and keyphrases are useful for a variety of purposes (throughout this paper, the authors use the latter to subsume the former). Keyphrases are a useful type of summary information. They can also be used to index, label, classify, cluster (Jones et al. 2000, Zamir et al. 1999), highlight, search (Jones 1999), retrieve (Arampatzis et al. 1998, Croft et al. 1991, Jones et al. 1999b), and browse (Gutwin et al. 1999, Jones et al. 1999a) information. Knowing that many types of documents would benefit from having keyphrases, including journal articles, web pages, email messages, news reports, magazine articles, and business papers (Turney 2003), unfortunately, the great majority of documents come without keyphrases, and manually assigning keyphrases is a tedious process that requires knowledge of the subject matter (Witten et al. 1999).

Regarding email messages, automatically extracting keyphrases from emails can aid in determining who knows what within an organisation. Commercial systems for expert identification using emails include: Tacit's ActiveNet (Tacit 2005), AskMe Enterprise (AskMe 2005), and Corporate Smarts' Intelligent Directory (Corporate smarts 2006). All of which extract keyphrases from users' emails and electronic documents. The information is placed into an expertise profile and distilled into a searchable database in order to enable users to query the system and find relevant people. The challenge is to extract keyphrases that give an indication of skills and experience exchanged in emails. With regards to evaluating such commercial systems and how they work, most of the information is only available in the form of white papers serving as a marketing tool to promote a product and point of view, which potentially could be biased.

### 3.2 Extraction Techniques

In keyphrase extraction, keyphrases that are considered important are selected from within the body of a given document. Several approaches have been proposed for the automatic extraction of keyphrases from text (Barker et al. 2000, Frank et al. 1999, Krulwich et al. 1996, Turney 1999). These approaches can be split into two main groups: supervised methods that require training documents and unsupervised methods that do not require training data.

According to Csomai et al. (2006), all supervised keyphrase extraction methods developed so far appear to share a common framework. They start with a preprocessing stage that handles the extraction and filtering of candidate phrases. This is followed by the actual ranking of the keywords using a set of contextual features (values of certain attributes for each candidate used in training and extraction) and a standard machine learning algorithm which can range from Naïve Bayes, to rule induction and genetic algorithms. Regarding features, several have been proposed. Some systems (Turney 1999) use word frequency and/or the word's position in the document (Frank et al. 1999) to indicate that a phrase is a keyphrase. Extractor scores candidate phrases based on a number of parameters which include frequency of the stemmed words in the phrase, length of the phrase, position of the phrase, etcetera. Extractor is part of GenEx, a hybrid genetic algorithm for keyphrase extraction, developed by Peter Turney (Turney 1999). GenEx has two components: Genitor genetic algorithm (Whitley 1989), and Extractor, the keyphrase extraction algorithm. Genitor tunes the parameters of Extractor until the optimal set of parameters is found and then it is no longer needed. Extractor generates candidate phrases, calculates a score for every candidate phrase using the parameter set and the top N (where N is desired number of output phrases specified by the user) highest scoring phrases are outputted.

KEA, a keyphrase extraction system, developed by members of The New Zealand Digital Library Project, a research programme at the University of Waikato, builds on Turney's work with GenEx's Extractor. KEA chooses candidate keyphrases using lexical methods, calculates feature values for each candidate, and uses naïve bayes machine learning to predict which candidates are good keyphrases. The way KEA selects keyphrases is by calculating for each candidate keyphrase feature values which are the TFXIDF, a measure of a phrases frequency in a document compared to its rarity in general use; and first occurrence, which is the distance into the document of the phrases first appearance.

Experimental results show that KEA's performance is statistically equivalent to GenEx (Frank et al. 1999). Initially (Jackson et al. 2004), KEA was used by the authors to extract keyphrases from email. However, after testing the system using KEA it was apparent that the keywords extracted were inappropriate. As a result, both KEA and GenEx were deemed inappropriate for the task at hand.

To extract keyphrases, Barker et al.'s (2000) system use a noun phrase approach, where firstly the system skims the input document for base noun phrases (non-recursive structure consisting of a head noun and zero or more pre-modifying adjectives and/or nouns). Then it uses the length of the phrase, the frequency of its use and the frequency of its head noun to assign scores to noun phrases, and finally it filters some noise from the set of top scoring keyphrases. Unfortunately, Barker et al. (2000) reported that there was no change in the performance of the system in comparison to the trained Extractor system in experiments involving human judges. Hulth's (2003) experimental results show that extracting noun phrase chunks gives better precision than extracting n-grams (sequence of 1...N words), and by adding part-of-speech (POS) tag(s) in which all words or sequences of words matching a given set of POS are extracted, a dramatic improvement in the results is achieved.

Disadvantages of systems that use supervised methods include the need for the training set (usually large) with pre-assigned keyphrases, which in this case is difficult to obtain. Moreover, these systems are intended for larger electronically stored documents such as journal articles, novels, and newspaper articles and not for emails which are considerably shorter.

On the other hand, unsupervised methods, which tend to be more domain-independent, depend on variations of TFxIDF or other similar methods (Csomai et al. 2006). They start by extracting candidate terms then ranking them based on some score. Keyphrases are then selected from the top ranked list. Using term frequency (TF) to determine a term's significance for instance is based on the idea that the more frequent a term occurs in a document the more important it is. This is not necessarily true when considering email messages as emails are usually short.

The downside of most keyphrase extraction systems is that most of the extracted keyphrases are common words that are not targeted thus resulting in a lot of noise. Therefore, it is important to distinguish between more general terms and those that are more likely to form keyphrases. In the following section, the authors present an approach for keyphrase identification from email text, which is purely based on the grammatical POS tags that surround these phrases.

## 4 KEYPHRASE EXTRACTION FROM EMAIL MESSAGES

Following the unpromising results obtained from KEA to extract keyphrases from emails the authors had to seek an alternative approach. The authors used the Natural Language ToolKit or more commonly known as NLTK, to build the keyphrase extractor. NLTK is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for the Python programming language. The keyphrase extractor is embedded in EKE (Email Knowledge Extraction). The extraction algorithm has two stages. The first stage involves training in which a model for POS tagging is created. Eric's Brill rule-based tagger was trained on the Brown corpus, resulting in the prediction model used in the second stage to tag new text. The second stage involves extraction in which keyphrases are extracted from email messages using the created model. Figure 2 shows the basic overview of the extraction stage.

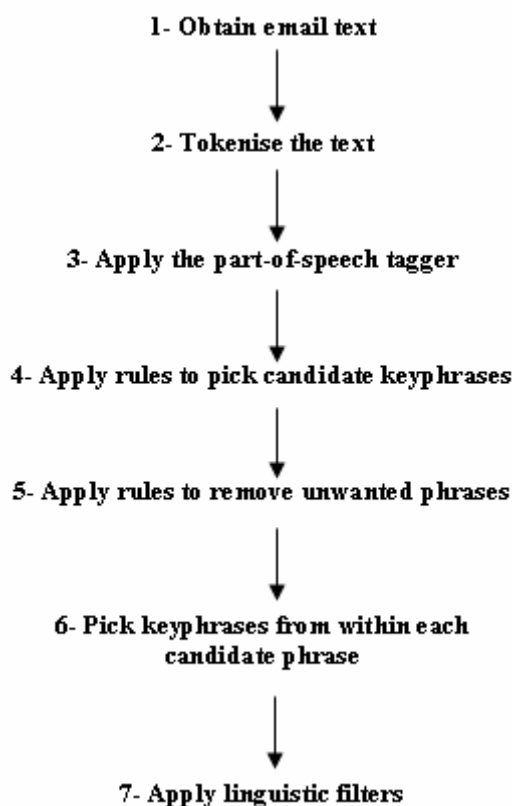


Figure 2. Stages of the extraction process

After the email text is obtained (1), the email body is split into tokens using regular expression rules (2). Regular expressions are a powerful tool for performing complex string searching, matching, and replacing. In order to discover patterns in text, individual tokens are tagged by their parts of speech using the POS tagging model (3). Following POS text tagging, rules are applied to select candidate keyphrases by grouping all occurrences of specific sequences of tags together (4). A rule is a sequence of grammatical tags that is most likely to contain words that make up a keyphrase. These rules were manually set by the authors by manually identifying keyphrases from an email sample consisting of 50 emails and looking at the grammatical properties that surround these phrases. After the sequences of tags are grouped together, rules are applied to remove a subset of phrases that are not relevant (5). Keyphrases are then selected from the identified candidate keyphrases (6). Finally, the system uses linguistic filtering to extract more important keyphrases (7). The result is a set of lines, each a sequence of tokens containing at least one letter.

## 5 EVALUATION APPROACH

A good keyphrase extraction system is able to extract relevant phrases without withholding non-relevant phrases. In this section, the authors describe the test corpus used to measure the performance of the keyphrase extraction process and the evaluation criteria used to measure the performance of the keyphrase extraction application.

### 5.1 Test Corpus

The experiments are based on three email collections. The authors refer to the email collection as the *sample* and to each individual email as the *sampling unit*. The *sampling units* belong to subjects from different backgrounds (people with English as their first language and people who can communicate in English, but is not their first language). All subjects belong to the age group 24-60. Table 1 below details the three samples used. The aim was to get all participants to take a greater part in the study. In practice however, this was very difficult. Sample 1 units consisted of email messages belonging to academic staff (i.e lecturers, research assistants, PHD candidates) from various academic domains. Participants were asked to provide the authors with a few knowledge-related knowledge email messages. A knowledge-related email is one that engages in the creation, identification, collection, organisation, sharing, adaptation and use of knowledge. The participants had to highlight the phrases that they think are relevant to their interest/knowledge. The sampling units in Sample 2 belonged to an employee working at company XYZ, a large company specialising in office solutions. Sampling units in Sample 3 were selected from the Enron online dataset. Since the authors of sampling units in Sample 2 and Sample 3 were not accessible, the authors of this paper had to manually assign keyphrases to the emails.

Sample name	Description	Size
Sample 1	Emails from various academic domains	45
Sample 2	Emails sent by employee E at company XYZ	19
Sample 3	Emails from Enron	50

Table 1. Details of the 3 email collections

All the *sampling units* were outgoing mail. The authors believe that the sampling units are representative of typical messages that are sent out in institutional and corporate environments.

### 5.2 Evaluation Metrics

When it comes to the evaluation framework and the choice of metrics, the authors propose to use the standard information retrieval metrics of precision and recall to reflect how well generated phrases

match phrases, which are considered to be ‘relevant’. Author phrases are typically used as the set of relevant phrases, or the ‘Gold Standard’. The evaluation of the automatic email keyphrase extraction system will consist of a comparison between the automatically extracted entries and the entries in the gold standard.

### 5.2.1 Criteria for matching keyphrases

A match occurs, if for example an author suggests the keyphrase “wordnet relation” and a keyphrase generation algorithm suggests the keyphrase “wordnet relations”. Yet, if the author suggests “wordnet relation” and the algorithm suggests “relation”, this is not counted as a match, since there are many different kinds of “relations”. However, if the authors suggest “wordnet” and the algorithm suggests “wordnet relations”, this is counted as a match because the algorithm is specifying the term. To summarise, a human selected keyphrase matches a machine-generated keyphrase when they either correspond to the same sequence of stems or when the machine generated keyphrase makes the human selected phrase more specific.

### 5.2.2 The performance measure

To evaluate the performance of the returned results, the so-called *f*-measure, derived from the standard *precision* and *recall*, is used here.

In the keyphrase extraction context, *precision* is the estimate of the probability that if a given system outputs a phrase as a keyphrase, then it is really a keyphrase. It is defined as the fraction of extracted entries that are relevant. *Recall* is an estimate of the probability that, if a phrase is a keyphrase, then a given system will output it as a keyphrase. Recall is defined as the fraction of relevant entries that are extracted.

The *f*-measure, which combines *p* and *r* with an equal weight, is defined in the following formula:

$$f\_measure = \frac{2 \times precision \times recall}{precision + recall} \dots\dots\dots \text{(Formula 1)}$$

## 6 EXPERIMENTS

Initially (Tedmori et al. 2006), the linguistic filtering (step 7, Figure 2) used involved removing only proper names (i.e. Rob, John) and common words specifically names of months and days (i.e. Monday, October). Table 2 shows a working example of an email sent through the keyphrase extraction system based on the stages of the extraction process shown in Figure 2. The results reported back were higher than other reported performance measurements from other algorithms. However, having acknowledged that the efficiency of the system requires further refining as the end user still has to delete a large number of irrelevant keyphrases (noise) that do not depict their expertise, that authors had to explore ways to improve the process detailed in order to optimise performance measurements.

After reviewing the results, the authors detected a number of general phrases that are common and do not communicate expertise and experience. The direction of the work then was to attempt to discard some of the candidate phrases through the use of more filtering techniques. More specifically, the approach was to use *WordNet*, a well recognised linguistic tool, at the filtering level to help decide whether a phrase is a keyphrase or not. WordNet is a lexical knowledgebase for English language that was created at Cognitive Science Laboratory of Princeton University in 1990. After keyphrases are picked from candidate keyphrases (step 6, Figure 2), each one word length keyphrase is looked up in WordNet. If it is found in WordNet, it is simply removed from the list of keyphrases. Similar to Zakos et al.’s (2005) work, this decision was made based on the intuition that if the term does not appear in

WordNet then it is most probably a reasonably specific term, thus making it fairly important. However, if a term appears in WordNet, then it is most likely a general term that does not disclose any skills. Table 3 is an update of table 2. It shows the same email being processed, however using WordNet at the filtering level. Notice how WordNet has helped to reduce the amount of noise extracted. Initially, when WordNet was not used (Table 2), “site” (which is not a useful keyphrase) was extracted as a keyphrase.

A working example of an email sent through the keyphrase extraction system
<p>&gt;&gt;&gt; Obtain email text  Mary &amp; Mike, I spoke to John today who is working on trying to construct a simple version of the email trainer. Mike, it might be worth you mentioning to John the web site that re-writes text so it has a better structure. Thomas</p> <p>&gt;&gt;&gt; Tokenise the text  &lt;mary&gt;, &lt;&amp;&gt;, &lt;mike&gt;, &lt;, &gt;, &lt;i&gt;, &lt;spoke&gt;, &lt;to&gt;, &lt;john&gt;, &lt;today&gt;, &lt;who&gt;, &lt;is&gt;, &lt;working&gt;, &lt;on&gt;, &lt;trying&gt;, &lt;to&gt;, &lt;construct&gt;, &lt;a&gt;, &lt;simple&gt;, &lt;version&gt;, &lt;of&gt;, &lt;the&gt;, &lt;email&gt;, &lt;trainer&gt;, &lt;.&gt; and so on....</p> <p>&gt;&gt;&gt; Apply POS Tagger  &lt;mary/NN&gt;, &lt;&amp;/cc-tl&gt;, &lt;mike/NN&gt;, &lt;,/&gt;, &lt;i/nn&gt;, &lt;spoke/vbd&gt;, &lt;to/to&gt;, &lt;john/vb&gt;, &lt;today/nr&gt;, &lt;who/wps&gt;, &lt;is/bez&gt;, &lt;working/vbg&gt;, &lt;on/in&gt;, &lt;trying/vbg&gt;, &lt;to/to&gt;, &lt;construct/vb&gt;, &lt;a/at&gt;, &lt;simple/jj&gt;, &lt;version/nn&gt;, &lt;of/in&gt;, &lt;the/at&gt;, &lt;email/NN&gt;, &lt;trainer/NN&gt;, &lt;.&gt; and so on....</p> <p>&gt;&gt;&gt; Pick Keyphrases from within each candidate phrase  S: &lt;mary/NN&gt; &lt;&amp;/cc-tl&gt; &lt;mike/NN&gt; &lt;,/&gt; &lt;i/nn&gt; &lt;spoke/vbd&gt; &lt;to/to&gt; &lt;john/vb&gt; &lt;today/nr&gt; &lt;who/wps&gt; &lt;is/bez&gt; &lt;working/vbg&gt; &lt;on/in&gt; &lt;trying/vbg&gt; &lt;to/to&gt; &lt;construct/vb&gt; &lt;a/at&gt; &lt;simple/jj&gt; &lt;version/nn&gt; &lt;of/in&gt; (Key phrase: &lt;the/at&gt; &lt;email/NN&gt; &lt;trainer/NN&gt;) &lt;.&gt; and so on....</p> <p>&gt;&gt;&gt;Apply linguistic filters (initial filtering technique)  (&lt;email/NN&gt; &lt;trainer/NN&gt;), (&lt;site/nn&gt;)</p>
<p><i>For the complete set of tags used in the Brown corpus please refer to <a href="http://www.comp.leeds.ac.uk/amalgam/tagsets/brown.html">http://www.comp.leeds.ac.uk/amalgam/tagsets/brown.html</a></i></p>

Table 2. A working example

A working example of an email sent through the keyphrase extraction system
<p>&gt;&gt;&gt; Obtain email text  Mary &amp; Mike, I spoke to John and so on....</p> <p>&gt;&gt;&gt; Tokenise the text  &lt;mary&gt;, &lt;&amp;&gt;, &lt;mike&gt;, &lt;, &gt;, &lt;i&gt;, &lt;spoke&gt; and so on....</p> <p>&gt;&gt;&gt; Apply POS Tagger  &lt;mary/NN&gt;, &lt;&amp;/cc-tl&gt;, &lt;mike/NN&gt;, &lt;,/&gt; and so on....</p> <p>&gt;&gt;&gt; Pick Keyphrases from within each candidate phrase  S: &lt;mary/NN&gt; &lt;&amp;/cc-tl&gt; &lt;mike/NN&gt; &lt;,/&gt; &lt;i/nn&gt; and so on....</p> <p>&gt;&gt;&gt;<b>Apply linguistic filters (with wordNet used)</b>  (&lt;email/NN&gt; &lt;trainer/NN&gt;)</p>
<p><i>For the complete set of tags used in the Brown corpus please refer to <a href="http://www.comp.leeds.ac.uk/amalgam/tagsets/brown.html">http://www.comp.leeds.ac.uk/amalgam/tagsets/brown.html</a></i></p>

Table 3. A working example(Update of Table 2)

## 7 RESULTS

Table 4 shows the evaluation results obtained from the two approaches. The first row shows the results obtained without filtering using WordNet and the second row shows the results obtained when using WordNet. Experimental evaluation showed that the use of WordNet did improve the precision, recall, and f-measure values in the three testing samples. However, the results are rather mixed. For example, in relation to *Corpus 1*, the system achieved when using WordNet a precision of 62.5% which is an improvement to the 53.3% precision that it achieved when it did not use WordNet. However, there was no noticeable improvement in recall (from 57.6% to 57.9%). This resulted in an improvement to f-measure (from 55.4% to 60.1%).

On *Corpus 2*, the system achieved a greater performance due to the notable improvement to both precision and recall values, a precision increase from 59.6% to 70.1% and a recall increase from 63.1% to 73.6%. This resulted in a considerable improvement to f-measure (from 61.3% to 71.8%). This is the highest f-measure achieved. On *Corpus 3*, the system achieved when using WordNet some improvement to precision with very little improvement to recall, resulting in an f-measure improvement from 44.8% to 48.1%. The reason behind the observed differences in performance of the system when tested on Corpus 1 and 3 and when tested on Corpus 2 is not clear. It is possible however, that the improvement directly proportional to the corpus size.

Corpus Name	Precision	Recall	f-measure
Corpus 1	53.3	57.6	55.4
with Wordnet	62.5	57.9	60.1
Corpus 2	59.6	63.1	61.3
with Wordnet	70.1	73.6	71.8
Corpus 3	41.7	48.3	44.8
with Wordnet	46.4	49.9	48.1

Table 4. Shows the precision, recall, and f-measure metrics for each of the collections

## 8 CONCLUSION

This paper sheds light onto the authors' attempts towards optimising the keyphrase extraction process from email messages for the purpose of supporting people in benefiting from each other's experience. The purpose of this paper is to contribute to the understanding of the problem of what may (or may not) be effective in extracting information from email to help identify experts. The idea was to use WordNet at the filtering stage to minimise the non relevant phrases extracted and optimise the process. The system was evaluated using three samples. Evaluation results indicate an improvement; however the results are rather mixed. In light of this, future research should be either conducted into exploring alternative approaches or ways to further improve on the process detailed in this paper in order to obtain higher performance metrics. One possible way of improving the detailed process is to use human input. The system simply learns based upon the user's choices of whether an extracted keyphrase is correct or incorrect. The authors are aware that this will add extra workload to the user at the initial set-up of the software, but believe that as time progresses this approach will greatly optimise the performance metrics.

## References

- Arampatzis, A.T., Tsoiris, T., Koster, C.H.A. and Van der Weide, T.P. (1998). Phrase-based information retrieval. *Information Processing & Management*, 34 (6), 693-707.
- AskMe. (2005). <http://www.askmecorp.com>, 16 December 2005.
- Bannon, L. J. (1986). Helping Users help each other. In D. A. Norman & S. W. Draper (Eds.), *User Centered System Design, New Perspectives on Human-Computer Interaction*, Hillsdale, NJ, London, , Lawrence Erlbaum Associates, 339-410.
- Barker, K. and Cornacchia, N. (2000). Using noun phrase heads to extract document keyphrases. In *Proceedings of the Thirteenth Canadian Conference on Artificial Intelligence*, Montreal, Canada, 40-52.
- Corporate smarts. (2006). <http://www.corporatesmarts.com>, 22 January 2006.
- Csomai, A. and Mihalcea, R.(2006). Creating a Testbed for the Evaluation of Automatically Generated Back-of-the-Book Indexes, In *Proceedings of the Conference on Computational Linguistics and Intelligent Text Processing (CICLing)*, Mexico City, 429-440.
- Croft, B., Turtle, H. and Lewis, D. (1991). The Use of Phrases and Structured Queries in Information Retrieval. In *Proceedings of SIGIR Conference*, ACM Press, 32-45.
- Frank, E., Paynter, G.W., Witten, I.H., Gutwin, C. and Nevill-Manning, C.G. (1999). Domain-Specific Keyphrase Extraction. In *Proceedings of Sixteenth International Joint Conference on Artificial Intelligence*, San Francisco, CA, 668-673.
- Gutwin, C., Paynter, G.W., Witten, I.H., Nevill-Manning, C. and Frank, E. (1999). Improving Browsing in Digital Libraries with Keyphrase Indexes. *Journal of Decision Support Systems* 27, 1-2, 81-104.
- Hulth, A. (2003). Improved Automatic Keyword Extraction Given More Linguistic Knowledge. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Sapporo, 216-223.
- Jackson, T.W. and Tedmori, S. (2004), Capturing and Managing Electronic Knowledge: The Development of the Email Knowledge Extraction, In *Proceedings of the Information Resources Management Association Conference: Innovations through Information Technology*, New Orleans, Louisiana, USA, 463-466.
- Jones, S. (1999). Design and Evaluation of Phrasier, an Interactive System for Linking Documents Using Keyphrases, In *Proceedings of Human-Computer Interaction: INTERACT'99*, Edinburgh, UK, IOS Press, 483-490.
- Jones, S. and Mahoui, M. (2000). Hierarchical Document Clustering Using Automatically Extracted Keyphrases. In *Proceedings of the Third International Asian Conference on Digital Libraries*, Seoul, Korea, 113-120.
- Jones, S. and Paynter, G. (1999a). Topic-based Browsing Within a Digital Library Using Keyphrases. In *Proceedings of the Fourth ACM Conference on Digital Libraries*, Berkeley, CA, ACM Press, 114-121.
- Jones, S. and Staveley, M. (1999b). Phrasier: a System for Interactive Document Retrieval Using Keyphrases. In *Proceedings of the 22nd International SIGIR Conference on Research and Development in Information Retrieval*, Berkeley, CA, ACM Press, 160-167.
- Kraut, R. E. and Streeter, L. A. (1995). Coordination in Software Development. *Communications of the ACM*, 38(3), 69-81.
- Krulwich, B., and Burkey, C. (1996). Learning user information interests through the extraction of semantically significant phrases. In *Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access*, Stanford, CA, 1996.
- Mcdonald, D.W. and Ackerman, M.S. (2000). Expertise Recommender: A Flexible Recommendation System and Architecture, In *Proceedings of the ACM Conference on Computer supported cooperative work*, PA USA, 231-240.
- Medelyan, O. and Witten I. H. (2006). Thesaurus Based Automatic Keyphrase Indexing. In *Proceedings of the Joint Conference on Digital Libraries*, Chapel Hill, NC, USA, pp. 296-297.
- Tacit. (2005). <http://www.tacit.com>, 10 December 2005.

- Tedmori, S. M., Jackson, T. W., and Bouchlaghem, N. M.(2006). Locating Knowledge Sources through Keyphrase Extraction. *Knowledge and Process Management*, 11(2), 100-107.
- Terveen, L. and Hill, W. (2001). Beyond Recommender Systems: Helping People Help Each Other. In *Human Computer Interaction in the new millenium*, Carroll, J. ed. Addison-Wesley, ACM Press, New York, 487-509.
- Turney, P. (1999). Learning to extract keyphrases from text. Technical Report ERB-1057, Institute for Information Technology, National Research Council of Canada.
- Whitley, D. (1989). The GENITOR Algorithm and Selective Pressure: Why Rank-Based Allocation of Reproductive Trials is Best, *Proceedings of the Third International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, California, 116-121.
- Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C. and Nevill-Manning, C.G. (1999). KEA: Practical automatic keyphrase extraction. In *Proceedings of DL '99*, 254-256. (Poster presentation.).
- Zakos, J. and Verma, B. (2005). Concept-based Term Weighting for Web Information Retrieval, in *Proceedings of the International Conference on Computational Intelligence and Multimedia Applications*, Las Vegas, USA, 173-178.
- Zamir, O. and Etzioni, O. (1999). Grouper: A Dynamic Clustering Interface to Web Search Results. *Computer Networks and ISDN Systems* 31, 11-16, 1361-1374.