# Locating knowledge sources through keyphrase extraction

This article was published in the journal, Journal of Knowledge and Process Management 13(2), pp. 100-107.

# Locating Knowledge Sources through Keyphrase Extraction

Sara Tedmori[1], Thomas W. Jackson[2], Dino Bouchlaghem[3]

[1]Department of Civil and Building Engineering, Loughborough University, Loughborough, Leicestershire, UK, LE11 3TU, Tel: +44 (0) 1509, Fax: +44 (0) 1509 223982,

Email: S.M.J.Tedmori@lboro.ac.uk

[2] Research School of Informatics, Loughborough University

[3]Department of Civil and Building Engineering, Loughborough University

**Abstract:**

There are a large number of tasks for which keyphrases can be useful. Manually identifying keyphrases can be a tedious and time consuming process that requires expertise, but if automated could save time and aid in creating metadata that could be used to locate knowledge sources. In this paper, the authors present an automated process for keyphrase extraction from email messages. The process enables users to find other people who might hold the knowledge they require from information communicated via the email system. The effectiveness of the extraction system is tested and compared against other extraction systems and the overall value of extracting information from email explored.

## 1. INTRODUCTION

Keywords and keyphrases are useful for a variety of purposes (throughout this paper, the authors use the latter to subsume the former). They can be used to summarise, index, label, categorise, cluster, highlight, browse, and search information (Turney, 2003). They can be used in many text-mining and knowledge management related applications. 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). Numerous techniques have been proposed to automatically extract keypharses from documents. However, these techniques mainly focus on extracting keyphrases from journal articles. Many other types of documents would also benefit from having keyphrases, including web pages, email messages, news reports, magazine articles, and business papers (Turney, 2003).

A relatively new area of research is trying to extract keyphrases from email messages to aid in determining who knows what within an organisation (Jackson and Tedmori, 2003). The keyphrases that are extracted should give some sort of indication of skills and experience exchanged in emails. Such keyphrases ought to disclose skills such as technical expertise, management skills, industry knowledge, education and training, work experience, professional background, knowledge in subject areas, etc. However, extracting keyphrases that describe the individual's expertise from an email body poses an immense challenge. Emails are freestyle text, not always syntactically well formed, domain independent, of variable length, and on multiple topics (Tzoukermann et al. 2001). Commercial systems for expert identification using emails include: *Tacit's ActiveNet* (Tacit, 2005), *AskMe Enterprise* (Ask Me, 2005) and *Corporate Smarts' Intelligent Directory* (Corporate Smarts,

2005). Figure 1 shows how such systems can be used to analyse emails to identify individuals or groups that have specific expertise. When an email is sent (step 1 in Figure 1), keyphrases are extracted (step 2 in Figure 1). The extracted keyphrases are then sent back to the user (step 3 in Figure 1) and placed into an expertise profile that the user can edit (step 4 in Figure 1). The expertise profile contains information about 'who knows what' within the organisation. This information is then distilled into a searchable database (step 5 in Figure 1) which users can query to find relevant people. Not all systems perform steps 3 and 4 and in this particular case these steps are specific to the system the authors are developing. Users are provided with an interface to rank their knowledge in the extracted keyphrases. With regards to similar extraction systems and how they work, most of the system information is only available in the form of white papers serving as a marketing tool to promote an organisations product and point of view which potentially could be biased.

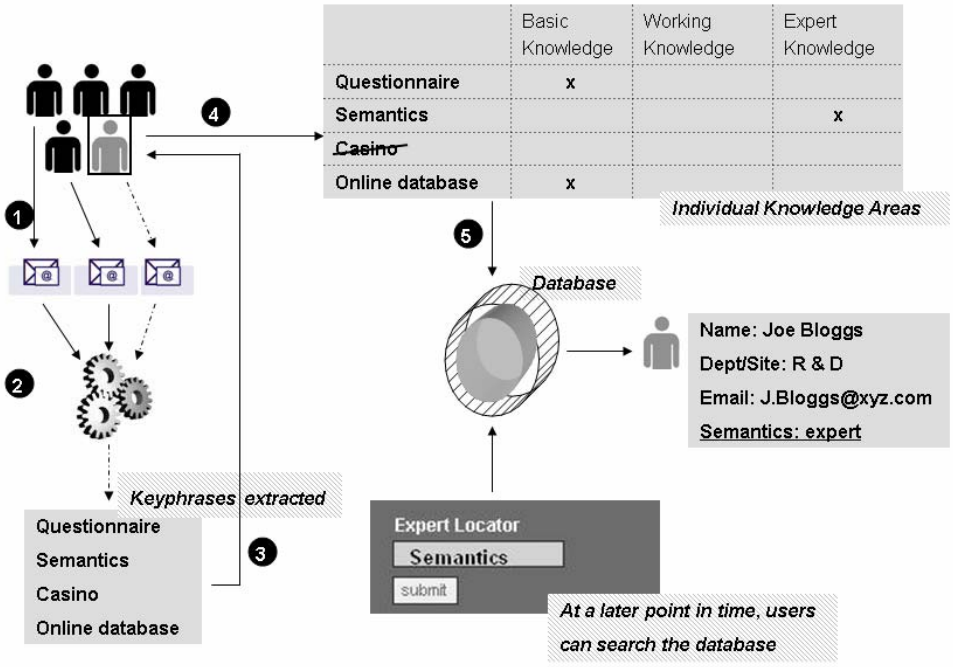


**Figure 1:** System Overview

In this paper, the authors review current keyphrase extraction techniques and present an automatic email message keyphrase extractor that will extract keyphrases and convey them to the user, by combining machine learning techniques and linguistics. The paper evaluates the proposed technique and concludes with a discussion of the proposed technique, including suggestions for future research.

## 2. EXTRACTION TECHNIQUES

Numerous papers explore the task of producing a document summary by extracting key sentences from the document (Luhn, 1958; Edmundson, 1969; Marsh et al., 1984; Paice, 1990; Paice and Jones, 1993; Johnson et al., 1993; Salton et al., 1994; Kupiec et al., 1995; Brandow et al., 1995; Jang and Myaeng, 1997; Tzoukermann, 2001). While similar to keypharse extraction, Truney (1999) argues that document summarisation is more difficult than keyphrase extraction. The end result is a set of sentences that often lack cohesion because anaphoric references are not resolved (Johnson et al., 1993; Brandow et al., 1995). Anaphors are pronouns (e.g., “it”,“they”), definite noun phrases (e.g., “the car”), and demonstratives (e.g., “this”, “these”) that refer to previously discussed concepts. Truney (1999) continues by saying that it may be impossible or very difficult for the reader of the summary to determine the referents of the anaphors. Johnson et al. (1993) tried to automatically resolve anaphors, however this resulted in overly long summaries. The problem of resolving anaphors does not arise in keyphrase extraction tasks, because anaphors are not keyphrases. Moreover, unlike a list of sentences, a list of keyphrases has no structure; a list of keyphrases can be randomly permuted without significant consequences. (Truney, 1999). There have been a number of techniques proposed for extracting keyphrases from text (Krulwich and Burkey, 1996; Frank et al., 1999; Turney,

1999; Barker and Cornacchia, 2000). Some of these techniques are domain specific while others are domain independent. Domain dependent techniques use machine learning, and require a collection of documents with keyphrases already attached for training purposes. Furthermore, these techniques (both domain dependent and domain independent) typically have some kind of connection to linguistics and/or use pure statistical methods. A number of applications have been developed using these techniques and their merits and pitfalls are discussed in the following paragraphs in order to determine the most effective way of extracting keyphrases from email.

Peter Truney (1999) devised *GenEx*, a hybrid genetic algorithm for keyphrase extraction. GenEx has two components: *Genitor*, a genetic algorithm, and *Extractor*, a keyphrase extraction algorithm. After stopword removal, candidate keyphrases (unigrams, bigrams, and trigrams) from the input document are scored based on a number of parameters. These parameters include frequency of the stemmed words in the phrase, length of the phrase, position of the phrases, etcetera. To maximise the performance on the training data, the *Genitor* genetic algorithm tunes the parameters of *Extractor*. *Genitor* is no longer needed after the training process. When the optimal set of parameters are found, *Extractor* can extract the best set of keyphrases, that is the one that has the most matches to the known keyphrase set in the training document set (Truney, 1999).

*KEA* has been developed by Frank et al. (1999). *KEA* is based on the naïve Bayes machine learning method. *KEA* uses a simpler set of features than Truney's *GenEx* algorithm. The two feature values that *KEA* calculates for each candidate keyphrase 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 phases first appearance. The

machine-learning scheme first builds a prediction model using training documents where the author's keyphrases are known, and then uses the model to find keyphrases in new documents. *KEA* chooses candidate keyphrases using lexical methods, calculates feature values for each candidate, and uses machine learning to predict which candidates are good keyphrases. The length of candidate phrases is limited to three. Frank et al. (1999) evaluate the *KEA* algorithm in relation to *GenEx* algorithm. The experiments show that *KEA*'s performance is statistically equivalent to *GenEx*. Initially, *KEA* was used by the authors (Jackson and Tedmori, 2003) to extract keyphrases from electronic messages. However, after testing the system it was apparent that the keywords extracted by *KEA* were inappropriate for the task of extracting keyphrases from email messages. As a result, GenEx was deemed inappropriate for the task at hand and alternatives had to be explored.

Peter Turney (2003) argues that a limitation of previous automatic keyword extraction algorithms is that the output keyphrases are at times incoherent. That is, that the majority of the extracted keywords may fit well together however, there will be a minority of outliers with no semantic relation to the majority or each other and he continues to argue that discarding this minority might improve the quality of the machine-extracted keyphrases**.** He suggests a different approach which is to use the degree of statistical association among candidate keyphrases as evidence that they may be semantically related, and thus avoiding them tends to improve the quality of the extracted keywords. These coherence features are not domain specific, and his experiments show that their use improves the quality of extracted keywords even when the testing domain is different than the training domain (Truney, 2003). Hulth (2003) pinpoints two common drawbacks *of GenEx* and *KEA* algorithms. The first drawback is that the number of words in a keyphrase is limited to

three knowing that in the training data 9.1% of the manually assigned keywords consist of four or more words. The second drawback is that the user must state how many keywords to extract from each document (Hulth, 2003).

Common to these systems is the approach of extracting keyphrases from text as a supervised learning task (Truney, 1999; Frank et al., 1999). These systems require a separate training document set with keyphrases already assigned in order to function properly. Email messages with pre-assigned keyphrases, to be used as a training set, are 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.

A common approach to extracting keyphrases when no machine learning is involved is by means of parts-of-speech (POS) patterns. Barker and Cornacchia (2000) describe a system where noun phrases are chosen from a document as keyphrases. The system first skims the input document for base noun phrases (non-recursive structure consisting of a head noun and zero or more premodifying 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. Barker and Cornacchia (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 (Barker and Cornacchia, 2000).

Hulth (2003) reports that keyword extraction from abstracts can be achieved by using simple statistical measures as well as syntactic information from the documents as input to the machine-learning algorithm. His experimental results show that extracting noun phrase

chunks gives better precision than n-grams (sequence of 1…N words), and by adding POS tag(s) assigned to the term so that all words or sequences of words matching any of a set of POS are extracted and a dramatic improvement is achieved. By using phrases, the length of the potential words is not restricted, rather potential terms are treated as units. When inspecting manually assigned keywords, the vast majority turn out to be nouns or noun phrases (Hulth, 2003).

Tzoukermann et al. (2001) present *GIST-IT*, a system for automatic extraction of salient information from email messages, for the purpose of providing an informative, generic, 'at-a-glance' summary. *GIST-IT* follows a process similar to *KEA* in that a set of candidate noun phrases are built up and assigned features that are then used to decide on the keyphrases. *GIST-IT* offers two significant improvements on *KEA*. Firstly, *GIST-IT* is intended for single email messages, and the training for feature selection takes place largely on an email corpus. This implies that *GIST-IT* is much more specific to email keyphrase extraction than *KEA*. *GIST-IT* uses some linguistic filtering which include: removing unimportant modifiers (i.e. most, more, etc), removing common words, and removing empty nouns (i.e. lot, group, set).

Common to these systems (Tzourkman et al., 2001; Hulth, 2003) is the extraction of noun phrases from text. However, the downside is that in spite of the filtering employed, many of the extracted keyphrases are common words that are likely to occur in numerous emails in a whole range of contexts. Therefore, it is important to distinguish between more general nouns 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.

## 3. KEYPHRASE EXTRACTION FROM EMAIL MESSAGES

This section describes a keyphrase extraction algorithm for email text. The algorithm is fully implemented and embedded in Email Knowledge Extraction (*EKE)*, an agent developed by the authors that enables its users to find other people who hold the required knowledge of a specific domain. The extraction algorithm has two stages. The first stage involves training in which a model for POS tagging is created. 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. The input to the system is a single email message. After the email text is obtained, the text is split into tokens using regular expression rules. In order to discover patterns in text, the next step is to tag the words in the email message by their parts of speech. The Brill rule-based tagger is used to assign the most likely single part of speech tag (noun, verb, adjective, etc.) to each word in the email. Brill tagging is a type of transformational-based learning. It is a supervised learning method since it needs annotated training data. It compiles a list of transformational correction rules. This tagger works by automatically recognising and remedying its weakness, thereby incrementally increasing its performance. The Brown corpus was used as the annotated training document set. The Brown corpus consists of 500 texts, each consisting of just over 2,000 words. In total, it contains 1,014,312 words sampled from 15 text categories. The result of the supervised learning is a prediction model that will be used to tag new text.

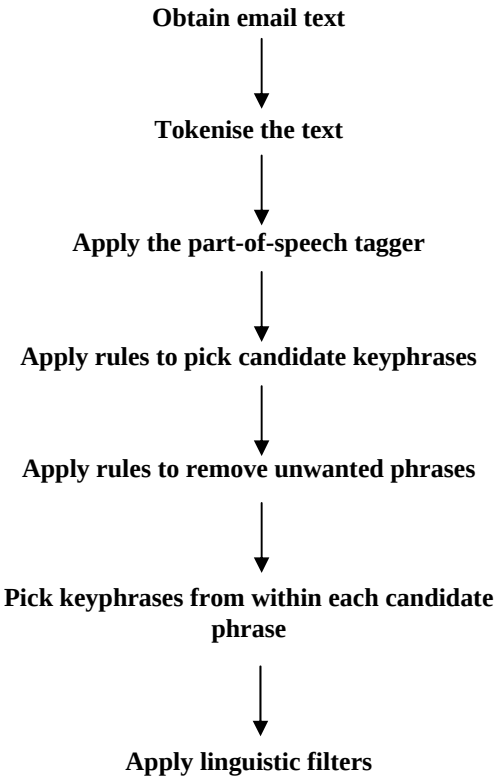


**Figure 2:** Stages of the extraction process

Following POS text tagging, rules are applied to select candidate keyphrases by grouping all occurrences of specific sequences of tags together. 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. Keyphrases are then selected from the identified candidate keyphrases. Finally, the system uses linguistic filtering to extract more important keyphrases. The result is a set of lines, each a sequence of tokens containing at least one letter. Table 1 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 primary advantage of this technique is that it is domain and genre independent.

| **A working example of an email sent through the keyphrase extraction system** |
|---|
| **>>> Obtain email text**<br>Hi Dany, I've had some experience with online surveys. I usually use html to design the survey and php to process the html and store the results in a database. I know there are alternative languages that you can use, but php is easy to learn and you can find a lot of material on the web. I recommend you start with designing your survey in html! Sara |
| **>>> Tokenise the text**<br><hi>, <dany>, <,>, <i've>, <had>, <some>, <experience>, <with>, <online>, <surveys>, <.>, <i>, <usually> and so on…. |
| **>>> Apply POS Tagger**<br><hi/NN>, <dany/NN>, <,/,>, <i've/NN>, <had/hvd>, <some/dti>, <experience/nn>, <with/in>, <online/NN>, <surveys/nns>, <./.>, <i/nn>, <usually/rb> and so on…. |
| **>>> Apply rules to pick candidate keyphrases and to remove unwanted phrases**<br>(S: <hi/NN> <dany/NN> <,/,> <i've/NN> <had/hvd> <some/dti> <experience/nn> <with/in> **(Key phrase: <online/NN> <surveys/nns>)** <./.> <i/nn> <usually/rb> and so on…. |
| >>> **Pick Keyphrases from within each candidate phrase**<br><online/NN><surveys/nns>, <php/NN>, <html/NN>, <database/NN>, <php/NN>, <html/NN> and so on…. |
| **>>>Apply linguistic filters**<br><online/NN><surveys/nns>, <html/NN>, <database/NN>, <php/NN> and so on…. |
| *For the complete set of tags used in the Brown corpus please refer to http://www.comp.leeds.ac.uk/amalgam/tagsets/brown.html* |

Table 1: A Working Example

## 4. EVALUATION AND RESULTS:

In this section, the authors firstly describe the test corpus used to measure the performance of the keyphrase extraction process. The authors then describe the evaluation criteria that will be used to measure the performance of the keyphrase extraction application.

### 4.1 Test corpus

The experiments in this report 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*. For each *sampling unit*, there is a target set of keyphrases that have been generated by hand.

Table 2 below details the three corpuses used. The *sampling units* are collected from 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.

| Corpus Name | Description | Size |
|---|---|---|
| Corpus 1 | Emails form various academic domains | 45 |
| Corpus 2 | Employee E From Company XYZ | 19 |
| Corpus 3 | Enron | 50 |

Table 2: details of the 3 email collections

All the *sampling units* were outgoing mail. The authors believe that *sampling units* are representative of typical messages that are sent out in institutional and corporate environments. The *sampling units* of the *sample*, Corpus 1, were collated from various academic disciplines (computer science, information science, building and construction engineering). The *sampling units* of the second *sample*, Corpus 2, are specific to one employee from a large supplier of total office solutions in the UK & Ireland, which for confidentiality reasons in is referred to as Employee E from Company XYZ. The *sampling units* of the final sample, Corpus 3, are collated from the Enron email dataset, which is freely available on the net.

**4.2 Evaluation Approach**

There are two basic approaches to evaluating automatically generated keyphrases (Jones and Paynter, 1999). The first adopts 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 usually used as the set of relevant phrases, or the 'Gold Standard'. Author phrases stand for the list of phrases usually found at the beginning of many articles such as academic journals.

The second approach is to gather subjective keyphrase assessments from human readers. Previous studies involving human phrase assessment (Barker and Cornacchia, 2000; Chen, 1999; Turney, 2000) follow essentially the same methodology. Subjects are provided with a document and a phrase list and asked to assess in some way the relevance of the individual phrases (or of sets of phrases) to the given document. A study by Jones and Paynter (1999), shows that authors do provide good quality keyphrases and thus can be used as the 'Gold Standard' against which other keyphrases can be compared.

The work in this paper adopts the first approach. However, the authors of the emails need to highlight the phrases that they think are relevant. The authors of the emails need to highlight keyphrases that appear in the body of the email text. Keyphrases consisting of more than one word should be in the same order as in the body of the email text. At occasions, when the authors of emails were not accessible, the authors of this paper had to manually assign keyphrases to the emails. These keypharses were then used as the 'Gold Standard'.

The task is to take an email message as input and automatically generate a list (containing no duplicate keyphrases) of keyphrases as output. The output keyphrases always appear somewhere in the body of the input email document. The performance measure is based on the number of matches between the machine-generated phrases and the human generated phrases. In the following subsections, the authors will define what matching keyphrases means and how the performance measure is calculated from the number of matches.

4.2.1 Criteria for Matching Phrases

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.

4.2.1 The Performance Measure

Researchers in information retrieval commonly use *precision* and *recall* to evaluate the performance of the returned results (e.g. search results returned). 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. *Recall* is an estimate of the probability that, if a phrase is a keyphrase, then a given system will output it as a keypharse. However, there is a well-known trade-off between *precision* and *recall*. One can be optimised at the expense of the other (Truney, 1999). For example, if it is guessed that all phrases are keyphrases, then recall is 100%, but precision will be close to 0%. On the other hand, if one relevant keyphrase is guessed as the only keyphrase then precision might be 100%, but recall would be close to 0%. What is required is a performance measure that yields a high score only when precision and recall are balanced. A measure that is widely used in information retrieval is the f-measure, defined as

$$f-measure = \frac{2 \times precision \times recall}{precision + recall} \quad \text{..... (Formula 1)}$$

**4.3 Results**

In Table 3, precision, recall, and the f-measure results are shown. The highest precision (59.6), recall (63.1), and f-measure (61.3) were achieved on the smallest sample (19 messages). Since only three sets were evaluated, one cannot determine the coloration between size of the sample and performance of the extractor.

| Corpus Name | Precision | Recall | f-measure |
|---|---|---|---|
| Corpus 1 | 53.3 | 57.6 | 55.4 |
| Corpus 2 | **59.6** | **63.1** | **61.3** |
| Corpus 3 | 41.7 | 48.3 | 44.8 |

Table 3: Shows the precision, recall and f-measure values for each of the collections.

Turney (1997) evaluates four keyphrase extraction algorithms using 311 email messages collected from 6 employees, and in which 75% of each employee's messages was used for training and 25% (approximately 78 messages) was used for testing. His evaluation approach is similar to the authors of this paper and the highest f-measure reported was that of the NRC, the extractor component of GenEx, which uses supervised learning from examples. The f-measure reported is 22.5, which is, as expected, significantly less than the f-measures shown in Table 3. Moreover, Hulth (2003) reports results from three different term selection approaches. The highest f-measure reported was 33.9 from the n-gram approach with POS tags assigned to the terms as features. All unigrams, bigrams, and trigrams were extracted, after which a stop list was used where all terms beginning or ending with a stopword were removed. Again, the result reported is less than the authors lowest f-measure (44.76). The system Hulth (2003) reports, limits itself by limiting the number of tokens in the keyphrases to three.

## 5. CONCLUSION

In this paper, the authors presented a process for keyphrase extraction from email messages. The method uses machine learning to tag new text by its part of speech, then extracts keyphrases purely based on POS tags that surround these phrases. The system was evaluated using three samples. The highest f-measure obtained was 61.3. If comparing with other reported performance measurements from other algorithms, the f-measure obtained by the authors is higher. The f-measure results detailed in this paper are higher than previously reported findings and the keyphrases extracted have provided an effective means of determining who knows what within an organisation. However, the efficiency of the system still requires refining as the end user still has to delete a large number of irrelevant

keyphrases (noise) that do not depict their expertise. Therefore, future research should be conducted into exploring ways to improve the process detailed in this paper in order to obtain higher performance measurements.

## 6. REFERENCES

AskMe. http://www.askmecorp.com [16 December 2005]

Barker K, Cornacchia N. 2000. Using Noun Phrase Heads to Extract Document Keyphrases. *In Proceedings of the Thirteenth Canadian Conference on Artificial Intelligence* (LNAI 1822), Montreal, Canada, 40-52.

Brandow R, Mitze K, Rau LR. 1995. The automatic condensation of electronic publications by sentence selection. *Information Processing and Management*, 31 (5), 675-685.

Chen KH. 1999. Automatic Identification of Subjects for Textual Documents in Digital Libraries. *Los Alamos National Laboratory*, Los Alamos, NM, USA.

Corporate smarts. 2006. http://www.corporatesmarts.com [22 January 2006]

Edmundson HP. 1969. New methods in automatic extracting. *Journal of the Association for Computing Machinery*. 16 (2), 264-285.

Frank E., Paynter G.W., Witten I.H., Gutwin C. and Nevill-Manning C.G. 1999. Domain-specific keyphrase extraction, *Proc. Sixteenth International Joint Conference on Artificial Intelligence*, Morgan Kaufmann Publishers, San Francisco, CA, pp. 668-673 http://www.nzdl.org/Kea/Frank-et-al-1999-IJCAI.pdf

Hulth A. 2003. Improved Automatic Keyword Extraction Given More Linguistic Knowledge. *In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'03)*. Sapporo.

Jackson TW, Tedmori S. 2004. Capturing and Managing Electronic Knowledge: The Development of the Email Knowledge Extraction. *Innovations Through Information Technology*, Khosrow-Pour, M. (ed.), Idea Group, IRMA, New Orleans, USA, pages 463-466.

Jang DH, Myaeng SH. 1997. Development of a document summarization system for effective information services. *RIAO 97 Conference Proceedings: Computer-Assisted Information Searching on Internet*; 101-111. Montreal, Canada.

Johnson FC, Paice CD, Black WJ, Neal AP. 1993. The application of linguistic processing to automatic abstract generation. *Journal of Document and Text Management* 1, 215-241.

Jones S, Paynter G. 1999. Topic-based browsing within a digital library using keyphrases. *Proceedings of the fourth ACM conference on Digital libraries*; 114-121, Berkeley, California, United States

Krulwich B, Burkey C. 1996. Learning user information interests through the extraction of semantically significant phrases. *In AAAI 1996 Spring Symposium on Machine Learning in Information Access.*

Kupiec J, Pedersen J, Chen F. 1995. A trainable document summarizer. *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 68-73, New York: ACM.

Luhn HP. 1958. The automatic creation of literature abstracts. *I.B.M. Journal of Research and Development*, 2 (2), 159-165.

Marsh E, Hamburger H, Grishman R. 1984. A production rule system for message summarization. *In AAAI-84, Proceedings of the American Association for Artificial Intelligence*, pp. 243-246. Cambridge, MA: AAAI Press/MIT Press.

Paice CD. 1990. Constructing literature abstracts by computer: Techniques and prospects. *Information Processing and Management*, 26 (1), 171-186.

Paice CD, Jones PA. 1993. The identification of important concepts in highly structured technical papers. *SIGIR-93: Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 69-78, New York: ACM.

Salton G, Allan J, Buckley C, Singhal A. 1994. Automatic analysis, theme generation, and summarization of machine-readable texts. *Science*, 264, 1421-1426.

Tacit. 2005. http://www.tacit.com [10 December 2005]

Turney PD. 1997. Extraction of Keyphrases from Text: Evaluation of Four Algorithms, National Research Council, Institute for Information Technology, *Technical Report ERB-1051*. (NRC #41550)

Turney P. 1999. Learning to Extract Keyphrases from Text. *Technical Report ERB-1057*, Institute for Information Technology, National Research Council of Canada.

Turney PD. 2000. Learning Algorithms for Keyphrase Extraction. *Information Retrieval* 2, 4; 303-336.

Turney P. 2003. Coherent Keyphrase Extraction via Web Mining. *In Proceedings Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03),* 434-439, Acapulco, Mexico

Tzoukermann E, Muresan S, Klavans JL. 2001. GIST-IT: Summarizing Email using Linguistic Knowledge and Machine Learning. *In Proceeding of the HLT and KM Workshop*, EACL/ACL.

Witten IH, Paynter GW, Frank E, Gutwin C, Nevill-Manning CG. 1999. KEA: Practical automatic keyphrase extraction. *Proc. DL '99*, pp. 254-256. (Poster presentation.) http://www.nzdl.org/Kea/Frank-et-al-1999-IJCAI.pdf