Automated categorisation of e-journals by synonym analysis of n-grams

Article

Published Version

Automated Categorisation of E-Journals by Synonym Analysis of *n*-grams

Richard Hussey, Shirley Williams, Richard Mitchell
School of Systems Engineering
University of Reading
Reading, United Kingdom
{r.j.hussey, shirley.williams, r.j.mitchell}@reading.ac.uk

Abstract—**Automatic keyword or keyphrase extraction is concerned with assigning keyphrases to documents based on words from within the document. Previous studies have shown that in a significant number of cases author-supplied keywords are not appropriate for the document to which they are attached. This can either be because they represent what the author *believes* a paper is about not what it actually is, or because they include keyphrases which are more classificatory than explanatory e.g., "University of Poppleton" instead of "Knowledge Discovery in Databases". Thus, there is a need for a system that can generate an appropriate and diverse range of keyphrases that reflect the document. This paper proposes two possible solutions that examine the synonyms of words and phrases in the document to find the underlying themes, and presents these as appropriate keyphrases. Using three different freely available thesauri, the work undertaken examines two different methods of producing keywords and compares the outcomes across multiple strands in the timeline. The primary method explores taking *n*-grams of the source document phrases, and examining the synonyms of these, while the secondary considers grouping outputs by their synonyms. The experiments undertaken show the primary method produces good results and that the secondary method produces both good results and potential for future work. In addition, the different qualities of the thesauri are examined and it is concluded that the more entries in a thesaurus, the better it is likely to perform. The age of the thesaurus or the size of each entry does not correlate to performance.**

Keywords- Automatic Tagging; Document Classification; Keyphrases; Keyword Extraction; Single Document; Synonyms; Thesaurus

I. INTRODUCTION

Keywords are words used to identify a topic, theme, or subject of a document, or to classify a document. They are used by authors of academic papers to outline the topics of the paper (such as papers about "metaphor" or "leadership"), by libraries to allow people to locate books (such as all books on "Stalin" or "romance"), and other similar uses. The keywords for a document indicate the major areas of interest within it.

A keyphrase is typically a short phrase of one to five words, which fulfils a similar purpose, but with broader scope for encapsulating a concept. While it may be considered the authors' contention, it is inferred that a short phrase of a few linked words contains more meaning than a single word alone, e.g., the phrase "natural language processing" is more useful than just the word "language".

Previous work by Hussey et al. [1] showed that using a thesaurus to group similar words into keyphrases produced useful results. The experiments run used the 1911 edition of *Roget's Thesaurus* [2] as the basis of the work. This paper sets out to expand upon that work by examining the results in relation to results generated by chance and, by using a number of different thesauri, to generate the keyphrase groupings, to compare the results of the different systems, and the different thesauri.

Frank et al. [3] discuss two different ways of approaching the problem of linking keyphrases to a document. The first, keyphrase assignment, uses a fixed list of keyphrases and attempts to select keyphrases that match the themes of the document. The computational problem for this approach is then to determine a mapping between documents and keyphrases using already classified documents as learning aids. The second approach, keyphrase extraction, assumes there is no restricted list and instead attempts to use phrases from the document (or ones constructed via a reference document).

Previous research [4][5] has shown that for any given group of documents with keyphrases, there are a small number which are frequently used (examples include "shopping" or "politics" [5]) and a large number with low frequency (examples include "insomnia due to quail wailing" or "streetball china" [5]). The latter set is too idiosyncratic for widespread use; generally, even reuse by the same author is unlikely. Therefore, part of the issue of both keyphrase assignment and extraction is locating the small number of useful keyphrases to apply to the documents.

The work described here is concerned with keyphrase extraction and, as such, this paper covers the background research into keyword/keyphrase generation, outlines a proposed solution to the problem, and compares the performance of manually assigning keyphrases. The main aim is to take an arbitrary document (in isolation from a corpus) and analyse the synonyms of word-level *n*-grams to extract automatically a set of useful and valid keywords, which reflect the themes of that document. The words of the document are analysed as a series of *n*-grams, which are compared to entries in a thesaurus to find their synonyms and these are ranked by frequency to determine the candidate

keywords. The secondary aim is to look at a method of grouping the theme outputs into clusters, so that the results do not just show the most common theme swamping out any others.

The rest of the paper comprises the background and state-of-the-art (Section II), the implementation (Section III) and results gained (Section IV), a discussion (Section V), and conclusions and suggestions for future work (Section VI).

II. BACKGROUND

A review of literature in the area of automatic keyword generation has shown that existing work in these areas focuses on either cross analysing a corpus of multiple documents for conclusions or extrapolating training data from manual summaries for test documents.

While manual summaries generally require multiple documents to train upon, they do not need to compare each component of the corpus to all other components. Instead, they try to extrapolate the patterns between the pairs of documents and manual summaries in the training set.

The following two sections look at firstly the manual summaries and single document approaches, and then the multiple document methods.

A. *Single Documents*

Single document approaches make use of manual summaries or keyphrases to achieve their results. Tuning via manual summaries attempts to replicate the process by which a human can identify the themes of a document and reduce the text down to a summary/selection of keyphrases. The general approach taken involves a collection of documents (with associated human summaries) and a given method is applied to draw relationships between the document and the summary. From this, new documents (generally a test corpus that also contains human summaries) are subject to the derived relationships to see if the summaries produced by the system are useful and usable.

For creating summaries, Goldstein et al. [6] set out a system based upon assessing every sentence of the document and calculating a ranking for its inclusion in a summary. They made use of corpora of documents for which assessor-ranked summary sentences already existed, and attempted to train the system using weighted scores for linguistic and statistical features to produce similar or identical sentences.

When getting emails I get a notice that an email has been received but when I try to view the message it is blank. I have also tried to run the repair program off the install disk but that it did not take care of the problem.

(a)

Receive emails; some emails have no subject and message body

(b)

Figure 1. An example of a) a text and b) its summary [7]

A different approach is taken by the *Stochastic Keyword Generator* [7], a proposed system for classifying help desk problems with short summaries (see Figure 1). Submitted e-mails varied in their description of the problem and often contained duplicated or redundant data. Therefore, their system attempts to create a summary similar to those manually created by the help desk staff: concise, precise, consistent, and with uniform expressions. It uses a corpus of e-mails with manual summaries, and ranks source words for inclusion based on the probability that they will occur based on the probability from its training data. This allows for words that are not explicitly in the text to appear in the summary (see Figure 2).

For producing keyphrases, Barker and Cornacchia [8] propose a system that takes into account not only the frequency of a "noun phrase" but also the head noun. For example, tracking "the Canadian Space Agency" should also track counts of "the Space Agency" or "the Agency".

Wermter and Hahn [9] examine a method of ranking candidate keyphrases using the limited paradigmatic modifiability (LPM) of each phrase as a guide to locating phrases with low frequency but high interest to the document. This works on the principle that a given multi-word term is a number of slots that can be filled with others words instead. For example, "t cell response" contains three slots that are filled, respectively, by "t", "cell", and "response". Another phrase that could fit might be "white cell response" or "the emergency response". The probability there are no phrases that could fill the gaps (for any given combination of the original words and gaps) determines how

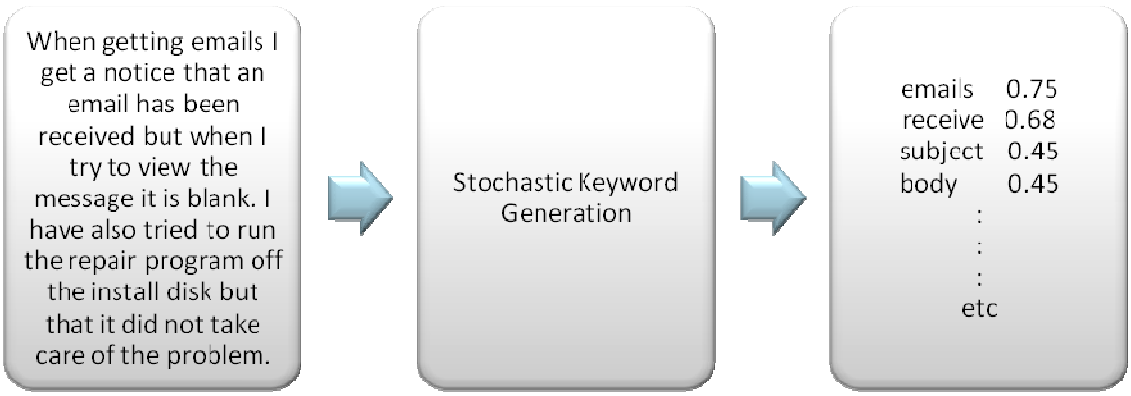

Figure 2. An example of SKG [7]

important the original phrase is, regardless of its actual frequency.

B. Multiple Documents

Multiple document approaches take a corpus and attempt to analyse relationships between the component elements to create methods for dealing with unseen elements. Most of these approaches are based on examining parts of an individual document in the corpus and then examining how that differs across the other documents.

"*TagAssist*" [4] makes use of a continually updated corpus of blog posts (supplied by [5]) and author-supplied tags to suggest tags for new blog posts. The system compares the author's tags and content of blog posts to work out the relationships that prompt the former to be chosen to represent the latter. Their baseline system works on a simple frequency count for determining output. Evaluated by ten human judges (unaware of which system produced each tags), the results showed that the original tags were the most appropriate (48.85%) with *TagAssist* coming in second (42.10%), and the baseline system last (30.05%).

The *C-Value* [10] is presented as a method for ranking "term words", taking into account phrase length and frequency of its occurrence as a sub-string of another phrase. It makes use of a linguistic filter, expressed as a regular expression, to ensure that only particular strings can be considered as candidate terms. Three filters were tested:

- Filter 1 – Noun $^{+}$ Noun
- Filter 2 – (Adjective | Noun) $^{+}$ Noun
- Filter 3 – ((Adjective | Noun) $^{+}$ | ((Adjective | Noun) * (Noun Preposition)$^{?}$) (Adjective | Noun)*) Noun

The more permissive filters, which accepted more grammatical structures, were found to perform more poorly, though all filters performed better than the baseline.

The *C-Value* is extended by the *NC-Value* [10], which adds a context weight to the calculation to determine which words surrounding the term are important.

The *SNC-Value* [11] (or *TRUCKS*) extends the *NC-Value* work, combining it with [12], to use contextual information surrounding the text to improve further the weightings used in the *NC-Value*.

Extra data may be used to gain more information on the relationships between the components, often gained from reference documents. Joshi and Motwani [13] make use of a thesaurus to obtain extra meaning from keywords. Their program, "*TermsNet*", can observe keywords in their original context in attempt to link keywords though a framework of linked terms, with directional relevance. This allows them to discover the "non-obvious" but related terms. For example, the term 'eurail' strongly suggests 'Europe' and 'railways', but neither suggest 'eurail' with the same strength. This means that 'eurail' is a non-obvious but highly relevant search keyword for both 'Europe' and 'railway'.

Scott and Matwin [14] use the *WordNet* lexical database [15] to find the hyponyms and feed this information to the Ripper machine learning system. The authors tested it against the DigiTrad folk song database [16], the Reuters-21578 news corpus [17], and a selection of USENET articles. They concluded that the system works better on documents written with "extended or unusual vocabulary" or which were authored collaboratively between several people.

Wei et al. [18] demonstrate such a system that uses *WordNet* to generate keywords for song lyrics. Their approach clusters the words of a song using *WordNet's* data to link words across the song. Keywords are then found at the centres of these links.

C. Background Conclusions

In conclusion, the literature review determined that work such as [13] or [14] used similar methods to the ones outlined in this paper. However, there are some key differences.

Joshi and Motwani [13] used a system of weighted links, which can differ in value from one side to another (in some cases being uni-directional as the weight 'removes' the link by setting it to a value of zero). This would differ from the proposed system, as the thesaurus does not contain the lexical knowledge to weight the links and a link from one synonym group to another is reciprocated in kind.

In [14], hyponyms were used, rather than synonyms. Hyponyms are words or phrases that share a *type-of* relationship, e.g. scarlet and vermilion are hyponyms of red, which is in turn a hyponym of colour. The proposed system would instead use synonyms: different words with almost identical or similar meetings.

III. IMPLEMENTATION

The basis of the work presented here is the examination of a document with reference to its synonyms and therefore the main bulk of the coding of the system related to this and the associated thesaurus file. Three input thesauri were used for analysis of the corpora, and these were Roget's "*Thesaurus of English Words and Phrases*" [16], Miller's "*WordNet*" [14], and Grady Ward's "*Moby Thesaurus*" [19].

The system was tested on a number of papers taken from a collection of online e-journals, Academics Conferences International (ACI) [20]. There were five e-journals in this collection, each on a different topic, and they were analysed separately. The topics were *Business Research Methods* (EJBRM), *E-Government* (EJEG), *E-Learning* (EJEL), *Information Systems Evaluation* (EJISE), and *Knowledge Management* (EJKM).

For each of the methods described below the thesaurus was loaded into the program and stored as a list of linked pairs of data, consisting of a unique Key (base word in the thesaurus) and an associated Value (its synonyms). The keys and values ranged from unigram word entries up to 7-gram phrases.

The project was split into a number of studies, and all the results were compared to a set of results generated by chance. The studies undertaken were the chance study, the unigram system, the *n*-gram study, and the clustering study. The following sections outline these approaches. The results are presented in Section IV.

A. Chance Study

For the chance study, the words from the source document were split into a list of individual words. From this list, a start point was chosen at random and a number of contiguous words were strung together to form a keyphrase. After each word was added, there was a chance that no further words would be added and this chance increased after each word so that it was more likely to produce shorter keyphrases than longer. The maximum length of the keyphrase was set at $n = 7$. The algorithm used was:

- Randomly select a word in the source document to act as a starting point.
- After each word is added, generate a random number less than or equal to n. If this number is greater than the number of words already in the phrase, add another word.
- Repeat until r keyphrases have been produced (in this study, r was chosen to be 5).

This algorithm is shown in Figure 3.

B. Unigram Systemc

The Unigram system was designed to act as a baseline for the experiments. The source text was split into a list of unigrams, and a count of the number of times each appeared in the source document occurred. The unigrams were then stemmed (to remove plurals, derivations, etc.) using the Porter Stemming Algorithm [21], and added to the list with combined frequencies from each of the unigrams that reduced to that stem. The resultant corpus of unigrams and stems was then compared to the entries in the thesaurus. Only the highest frequency keyword was output from the unigram system.

- For each n-gram in the thesaurus, compare the n-gram to the associated synonyms.
- For each synonym that matches, add the word to a list, and increase its frequency value by the value of the n-gram.
- Sort the list by frequency and output the top r ranked items (in this study, r was chosen to be 1).

C. The n-gram study

Following the results of the unigram study, the experiment was extended to examine the effects of multi-gram words on the output of the system. This allowed the system to output keyphrases as opposed to just the singular keywords of the unigram study.

For the n-gram study, the words from the source document were split into a number of n-gram lists, from unigrams up to 7-grams. For all of the lists the entries overlapped so that all combinations of words from the text were included. E.g., if the source text were "The quick fox jumped" then the bigrams would be "The quick", "quick fox", and "fox jumped" and the trigrams would be "The quick fox", and "quick fox jumped". For each document, the results of each of the n-grams were combined and considered together to determine the overall output.

- For each n-gram in the thesaurus, compare the n-gram to the associated synonyms.
- For each synonym that matches, add the word to a list, and increase its frequency value by the value of the n-gram.
- Sort the list by frequency and output the top r ranked items (in this study, r was chosen to be 5).

This algorithm is shown in Figure 4.

D. The clustering study

Examining the results of the n-gram study (as discussed in Section V below) revealed that only the highest frequency "group" or cluster of synonyms was being matched, and as such the clustering algorithm attempts to extend the n-gram algorithm to group the keyphrases into "clusters". It achieves this by finding the keyphrases that are of a similar theme and returning a single keyphrase for that group.

For example, the word "recovery" can mean either "acquisition" or "taking" [2]. The base system therefore could return multiple versions of the same concept as keyphrases. By clustering the results, the attempt was to prevent a single, "popular", concept dominating and allow the other themes to be represented. The method for this was:

- For each n-gram in the thesaurus, compare the n-gram to the associated synonyms
- For each synonym that matches, add the word to a list, and increase its frequency value by the value of the n-gram divided by the number of associated synonyms
- Then, for each Key entry in the thesaurus check to see if the frequency is equal to the highest frequency value in the found in the preceding step.
- For each synonym entry associated with the Key, add the synonym to a second list of words and increase its value by one.
- Sort the second list by frequency and output the top r ranked items (in this study, r was chosen to be 5).

This algorithm is shown in Figure 5.

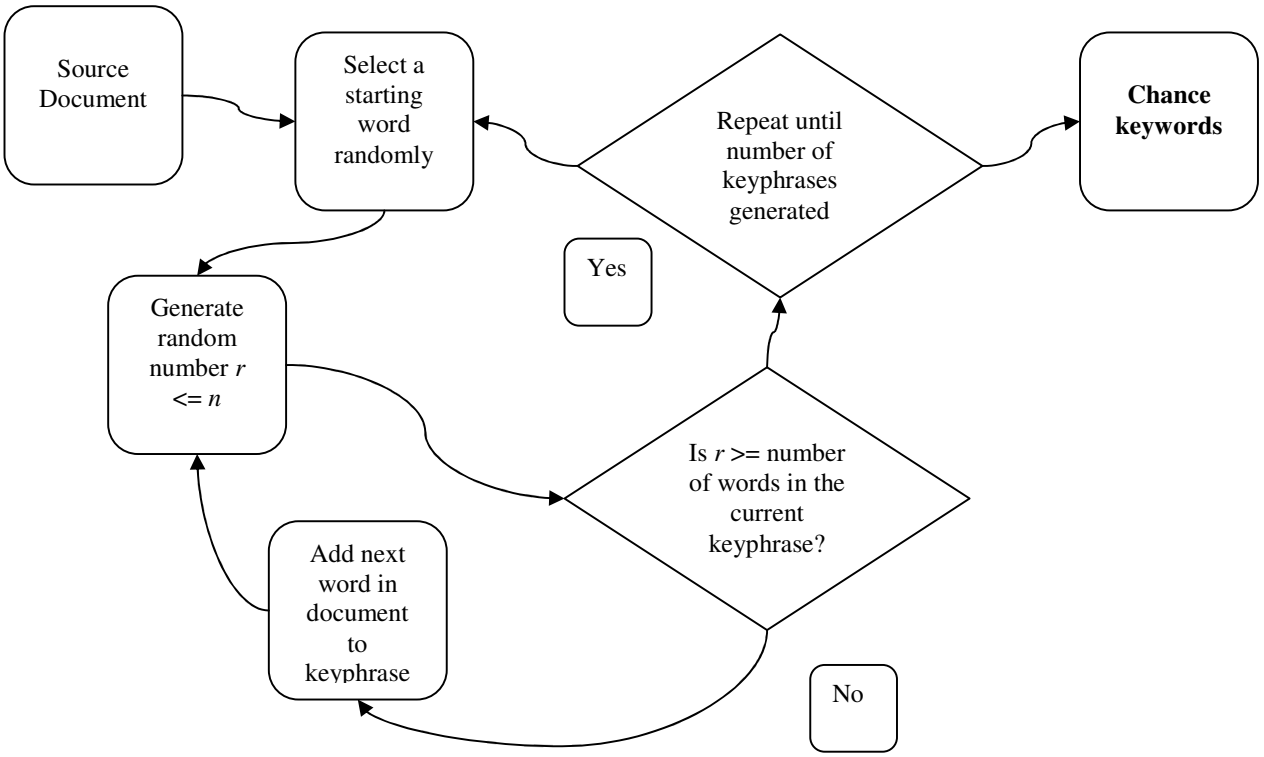

Figure 3. Chance algorithm

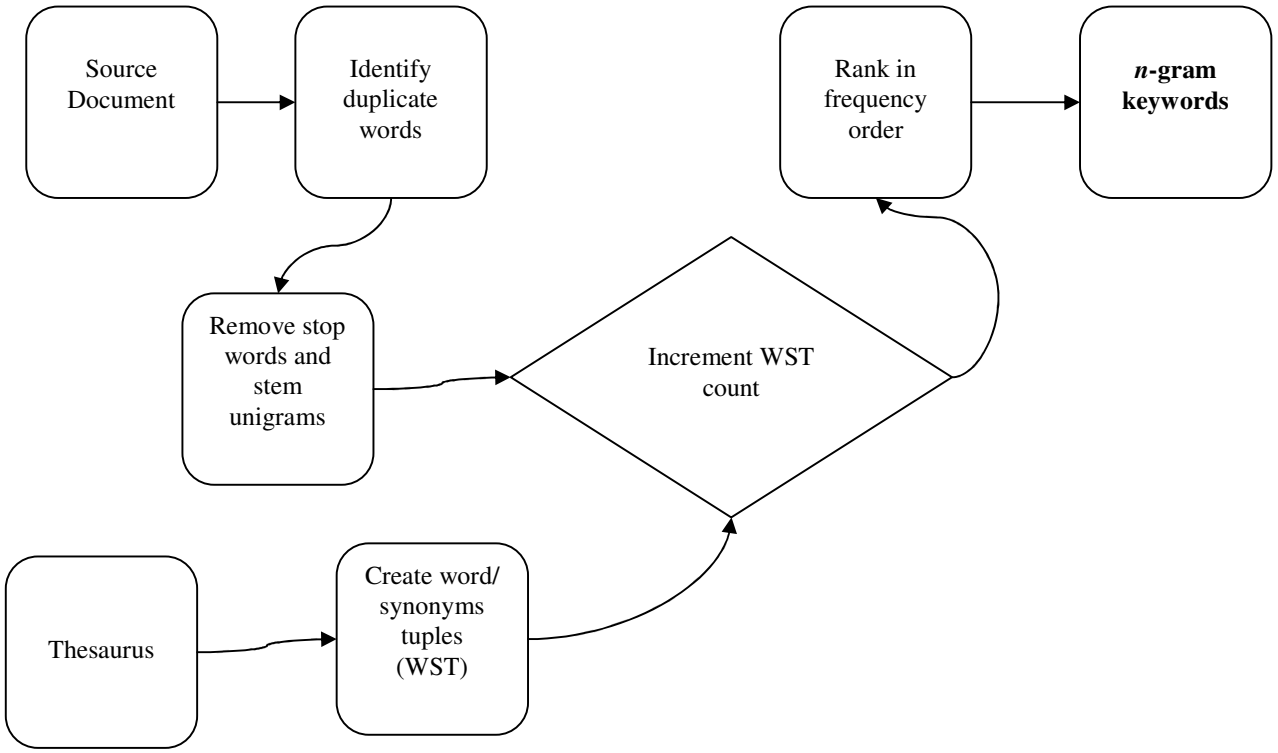

Figure 4. *n*-gram algorithm

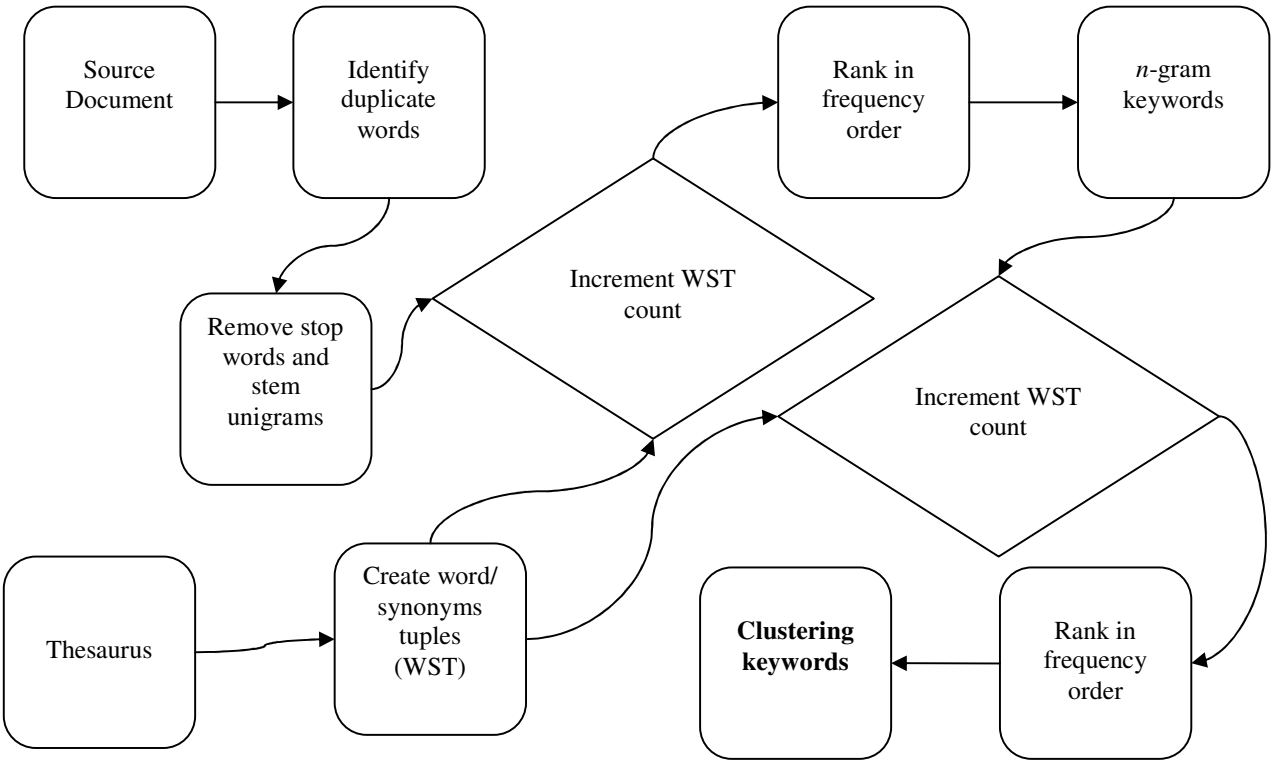

Figure 5. Clustering algorithm

IV. RESULTS

The results of these four studies are shown below. For each of the e-journals used, the authors of each paper in the journal had supplied an accompanying list of keyphrases summarising the content of that paper. These were therefore leveraged to provide a method of automatically evaluating the results of the work presented here.

For every paper, a match was recorded if at least one author-supplied keyphrase was a substring of, a superstring of, or exactly equal to a system-supplied keyword. This naïve text-matching approach would match the word "know" with both the words "know" and "knowledge".

For all of the tables the following explanations of each column apply. The 'Journal' column lists the five e-journals from ACI [20], and the 'Papers' column lists the number of papers in that corpus. The number 'Matched' is the number of papers in that journal that recorded a match, and 'Percentage' is the percentage number of papers in that journal that were considered a match. Where it appears, 'Increase' is the numerical value by which the percentage match has increased over the results of the chance study – i.e. if the match percentage was 5% in the chance study and 11% in *n*-gram study that would be an increase of 6.

A. *Chance Study*

The chance results showed almost no keyphrases being produced that matched the authors. The results can be seen in Table I.

TABLE I. CHANCE RESULTS

Journal	Papers	Matched	Percentage
EJBRM	72	0	0.00%
EJEG	101	2	1.98%
EJEL	112	0	0.00%
EJISE	91	1	1.11%
EJKM	110	5	4.81%
Average			1.58%

B. *Baseline System*

Table II, Table III, and Table IV show the baseline results for the study. The increase measures the performance compared to the results from Table I. The average percentage correct was 5.80%, an increase of 4.22 over the chance results from Table I.

TABLE II. BASE LINE ROGET RESULTS

Journal	Papers	Matched	Percentage	Increase
EJBRM	72	4	5.56%	5.56
EJEG	101	3	2.97%	0.99
EJEL	112	18	16.07%	16.07
EJISE	91	7	7.69%	6.58
EJKM	110	19	17.27%	12.46
Average			9.91%	8.33

TABLE III. BASE LINE WORDNET RESULTS

Journal	Papers	Matched	Percentage	Increase
EJBRM	72	0	0.00%	0.00
EJEG	101	3	2.97%	0.99
EJEL	112	0	0.00%	0.00
EJISE	91	1	1.11%	0.00
EJKM	110	6	5.77%	0.64
Average			1.90%	0.32

TABLE IV. BASE LINE MOBY RESULTS

Journal	Papers	Matched	Percentage	Increase
EJBRM	72	5	6.94%	6.94
EJEG	101	4	3.96%	1.98
EJEL	112	3	2.68%	2.68
EJISE	91	9	9.89%	8.78
EJKM	110	5	4.55%	-0.26
Average			5.60%	4.02

C. The n-gram study

The *n*-gram results showed a small improvement over the baseline, as can be seen in Table V, Table VI, and Table VII. The increase measures the performance compared to the results from Table I. The average percentage correct was 23.59%, an increase of 22.01 over the chance results from Table I.

TABLE V. RESULTS OF ROGET *N*-GRAM STUDY

Journal	Papers	Matched	Percentage	Increase
EJBRM	72	16	24.62%	24.62
EJEG	101	21	20.79%	18.81
EJEL	112	54	49.54%	19.54
EJISE	91	27	30.00%	28.89
EJKM	110	70	67.31%	62.50
Average			38.45%	30.87

TABLE VI. RESULTS OF WORDNET *N*-GRAM STUDY

Journal	Papers	Matched	Percentage	Increase
EJBRM	72	9	13.85%	13.85
EJEG	101	17	16.83%	14.85
EJEL	112	12	11.01%	11.01
EJISE	91	8	8.89%	7.78
EJKM	110	15	14.42%	9.61
Average			13.00%	11.42

TABLE VII. RESULTS OF MOBY *N*-GRAM STUDY

Journal	Papers	Matched	Percentage	Increase
EJBRM	72	17	23.61%	23.61
EJEG	101	18	17.82%	15.84
EJEL	112	18	16.07%	16.07
EJISE	91	19	20.88%	19.77
EJKM	110	20	18.18%	13.37
Average			19.31%	17.73

D. The clustering study

The clustering results show a reasonable improvement over the *n*-gram results and a significant increase over the chance results, as can be seen in Table VIII, Table IX, and Table X. The increase measures the performance compared to the results from Table I. The average percentage correct was 45.75%, an increase of 44.17 over the chance results from Table I.

TABLE VIII. RESULTS OF ROGET CLUSTERING STUDY

Journal	Papers	Matched	Percentage	Increase
EJBRM	72	31	43.06%	43.06
EJEG	101	73	72.28%	70.30
EJEL	112	77	68.75%	68.75
EJISE	91	46	50.55%	49.44
EJKM	110	94	85.45%	80.64
Average			64.02%	62.44

TABLE IX. RESULTS OF WORDNET CLUSTERING STUDY

Journal	Papers	Matched	Percentage	Increase
EJBRM	72	41	63.08%	63.08
EJEG	101	69	68.32%	66.34
EJEL	112	37	33.94%	33.94
EJISE	91	38	42.22%	41.11
EJKM	110	57	54.81%	50.00
Average			52.47%	50.89

TABLE X. RESULTS OF MOBY CLUSTERING STUDY

Journal	Papers	Matched	Percentage	Increase
EJBRM	72	16	22.22%	22.22
EJEG	101	21	20.79%	18.81
EJEL	112	20	17.86%	17.86
EJISE	91	20	21.98%	20.87
EJKM	110	23	20.91%	16.10
Average			20.75%	19.17

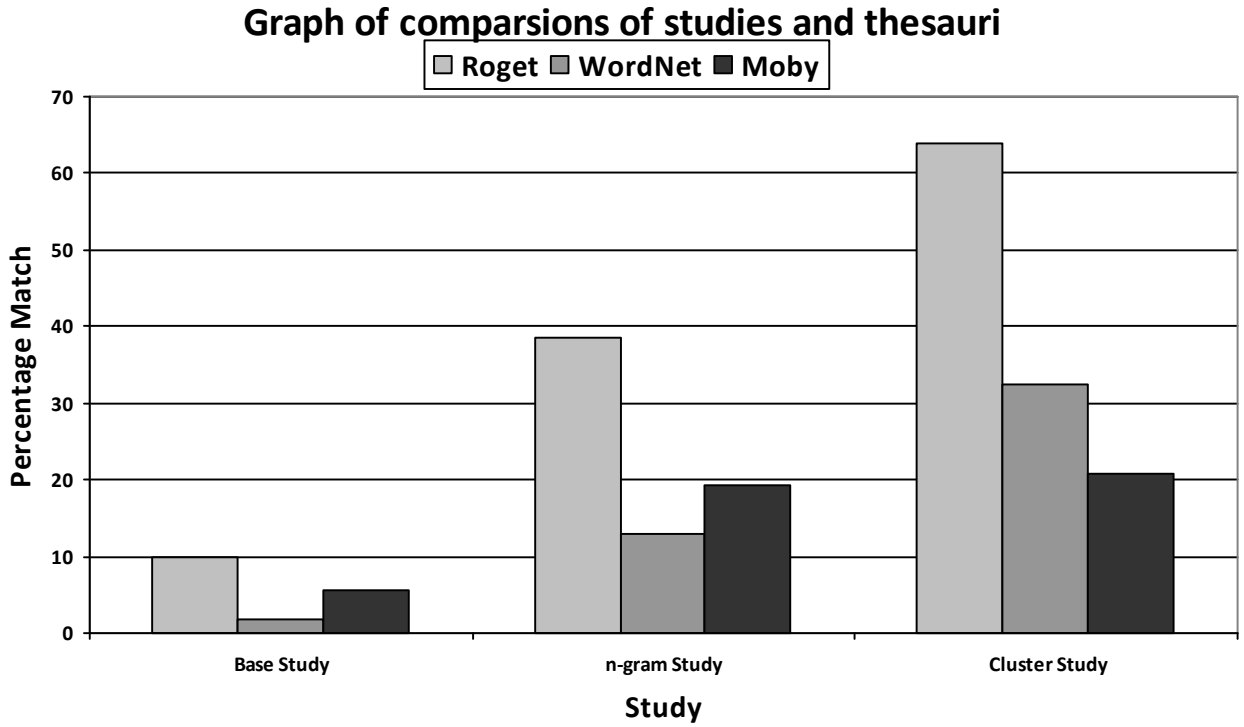

Figure 6. Graph of Studies/Percentage

V. DISCUSSION

The results show that using *n*-grams on their own produces a significant improvement over both chance and the baseline study (an average over the three thesauri of 23.59%). This shows that this method of using a thesaurus to group words into their conceptual clusters has potential to produce useful outputs.

However, the results did not vary when the number of *n*-grams was changed (ranging between 1 and 7) but the number of outputs *r* was maintained (this section was only tested on for the *WordNet* thesaurus). A possible explanation for this would be only the highest frequency group of synonyms is being matched by the author keywords.

Therefore, the algorithm was extended to include the clustering algorithm, which in turn produced a further, and significant, improvement (an average of 45.75% across the three thesauri). The results are shown in Figure 6 grouped by study, and clearly show that each addition to the study improved on the average result, and that in all studies the *Roget* thesaurus outperformed the rest. This is confirmed by Figure 7, which shows the same results grouped instead by thesaurus.

In addition to the issues found in the *n*-gram study further improvement on the results seems to be unlikely due to issues with the mechanism for confirming a match – author keywords. Some of the keywords submitted by the authors of the papers in the corpus may be tags instead of keywords. These can display meta-data that can often be irrelevant to the understanding of the document. An example seen in the corpus was the keyword "University of Birmingham" because the author of that paper worked there. This is valid as a tag but as a keyword, as it does not indicate a topic or a theme to which the document holds (other than in a rare case where the paper is about the University of Birmingham). This therefore lowers the chances of keyphrases being matched as the comparison data is filled with `noise'.

The synonyms are currently analysed context-free, and thus for a word with multiple meanings (e.g., "recovery" can mean "acquisition", "improvement", or "restoration" [2]) every occurrence of that word is treated the same. This means that a document equally about "improvement" and "restoration" could end up with the theme of "recovery" which (while a correct assumption) may not give the right meaning.

A. *Thesauri outcomes*

The results from the various studies all show that on average the *Roget's Thesaurus* outperforms *WordNet,* which in turn outperforms *Moby's Thesaurus*.

Appendix A contains a sample entry from each thesaurus for the word "question" (as an example). As can be seen, the *Roget* entry is the shortest and the *Moby* entry the longest and most comprehensive. As a thesaurus, *Roget* has 55,000 entries, *Moby* has 30,000, and *WordNet* has 5,000.

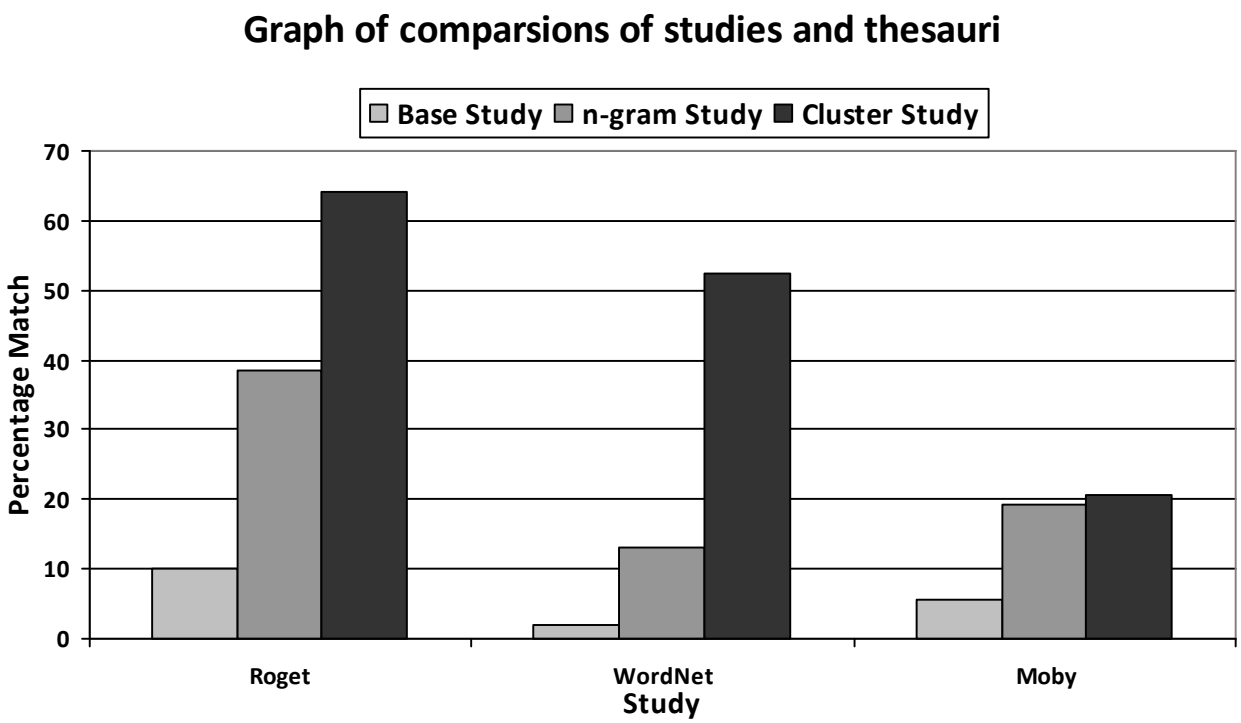

Figure 7. Graph of Thesauri/Percentage

The *Moby* and *WordNet* thesaurus entries are both newer (less than fifteen years old) than their counterpart *Roget* is, and consequently contain modern phrases such as "sixty-four dollar question" (see Appendix A). Yet, in spite of this, they perform worse than the one hundred year old thesaurus.

McHale [22] compares *WordNet* and *Roget* for measuring semantic similarity, and concludes that due to the combined relative uniformity of the hierarchy in *Roget* and the broader allowed set of semantic relationships, that it seems better at capturing "the popular similarity of isolated word pairs". This potentially allows it to find more words around a single concept, compared to the other thesauri studied, which work in smaller concept-circles.

VI. Conclusion and Further Work

The approach to synonym analysis developed in this paper shows good results for the test corpora used and potential for future study. Further study is required to compare the system to ones developed in similar areas, but this should provide a solid framework for taking the project forward.

The results, as mentioned in Section 0, show that the number of *n*-grams used does not affect the outcome of the system – all that matters is using the synonyms. This does not, however, mean that the keywords produced may not be more useful to the user, as they could be different enough not to match the success criteria but still relevant.

The results themselves were evaluated against the keywords submitted by the authors of the papers. *TagAssist* [4] showed that in 54.15% of cases, author keywords were judged as being inappropriate for the work with which they were associated. Therefore, when interpreting the results (which averaged around 60% matches) it should be remembered that they are produced by matching the output against the author keywords, which may be less than perfect for the task. A new method of evaluating the results is therefore required.

Another area of further work is to conduct more experiments to determine what differences there are between the thesauri, and what impacts the differences have on the results. When compared, results from Roget's thesaurus produced better results than WordNet and Moby, but it is not clear at this stage why that is it the case. It is possible, for example, that each of the thesauri is suited to a certain subject corpora (e.g., a medical corpus vs. a computer science corpus). Therefore, more experiments will need to be run with different corpora to ascertain if this is the case, or if the Roget's thesaurus is simply better suited to this application than the other two.

In addition, given the difference in size of each thesaurus a further area of study would be to attempt to make a single thesaurus that only contains the words found in all three and to see how well that thesaurus compares to the existing results. In a similar vein to this, another study would be to combine all three thesauri into a single but larger thesaurus and compare that to the existing results as well as to the version with reduced entries.

APPENDIX A

This appendix includes the entries from the three thesauri for the word "question".

A. *Roget entry for "question"*

Question

- inquiry, irreligion, unbelief doubt

Taken from [2]

B. *WordNet entry for "question"*

Question

- inquiry, query, interrogation, interrogate

Taken from [15]

C. *Moby entry for "question"*

Question

- Chinese puzzle, Parthian shot, Pyrrhonism, absurd, address, affirmation, agonize over, allegation, answer, apostrophe, apprehension, approach, ask, ask a question, ask about, ask questions, assertion, assuredly, at issue, averment, awake a doubt, baffling problem, basis, be at sea, be curious, be diffident, be doubtful, be dubious, be sceptical, be uncertain, beat about, bill, blind bargain, bone of contention, borderline case, brain twister, bring into question, burden, burn with curiosity, calendar, call in question, case, catechism, catechize, certainly, challenge, chance, chapter, clause, comment, communicate with, companion bills amendment, concern, confusion, contact, contest, contingency, correspond, crack, cross-interrogatory, cross-question, crossword puzzle, crux, debatable, debating point, declaration, definitely, demand, demurral, demurrer, dictum, difficulty, diffidence, dig around for, dig up, dispute, distrust, distrustfulness, double contingency, doubt, doubtful, doubtfulness, doubtlessly, dragnet clause, dubiety, dubiousness, enacting clause, enigma, enigmatic question, enquiry, escalator clause, essence, establish connection, examine, exclamation, expression, feel unsure, feeler, focus of attention, focus of interest, gamble, gape, gawk, get to, gist, greet with scepticism, greeting, grill, grope, guess, half believe, half-belief, harbour suspicions, have reservations, head, heading, hold-up bill, impossible, in doubt, in question, inconceivable, indubitably, inquire, inquire of, inquiry, insupportable, interjection, interpolate, interrogate, interrogation, interrogative, interrogatory, interview, issue, jigsaw puzzle, joker, knot, knotty point, leader, leading question, leeriness, living issue, main point, maintain connection, make advances, make contact with, make inquiry, make overtures, make up to, matter, matter in hand, meat, mention, mind-boggler, misdoubt, misgive, misgiving, mistrust, mistrustfulness, moot point, motif, motion, motive, mystery, nose around for, nose out, note, nut, nut to crack, objection, observation, omnibus bill, open question, peer, perplexed question, perplexity, phrase, piece of guesswork, point, point at issue, point in question, poser, position, preposterous, privileged question, problem, pronouncement, propose a question, proposition, propound a question, protest, proviso, pump, put queries, puzzle, puzzle over, puzzlement, puzzler, query, question, question at issue, question mark, questionable, questioning, quiz, quodlibet, raise, raise a question, reach, reflection, relate to, remark, remonstrance, remonstration, reply to, require an answer, respond to, rider, ridiculous, rubber, rubberneck, rubric, saving clause, say, saying, scruple, scrupulousness, seek, self-doubt, sentence, shadow of doubt, sight-unseen transaction, sixty-four dollar question, scepticalness, scepticism, smell a rat, sound out, stare, statement, sticker, stumper, subject, subject matter, subject of thought, subjoinder, substance, suspect, suspicion, suspiciousness, test, text, theme, thought, thrash about, throw doubt upon, topic, toss-up, total scepticism, touch and go, tough proposition, treat with reserve, trial balloon, uncertainty, undecided issue, under consideration, undoubtedly, unthinkable, utterance, vexed question, wager, want to know, wariness, why, wonder, wonder about, wonder whether, word, worm out of

Taken from [19]

ACKNOWLEDGMENT

The authors would like to thank the School of Systems Engineering for the studentship, which enabled this project, and the contributions from the reviewers to this paper.

REFERENCES

[1] R. Hussey, S. Williams, and R. Mitchell. 2011. "Keyphrase Extraction by Synonym Analysis of *n*-grams for E-Journal Classification", eKNOW, Proceedings of The Third International Conference on Information, Process, and Knowledge Management, pp. 83-86. Gosier, Guadeloupe/France. http://www.thinkmind.org/index.php?view=article&articleid=eknow_2011_4_30_60053 [Last accessed: 23 January 2012]

[2] P.M. Roget. 1911. "Roget's Thesaurus of English Words and Phrases (Index)". http://www.gutenberg.org/etext/10681 [Last accessed: 23 January 2012]

[3] E. Frank, G.W. Paynter, I.H. Witten, C. Gutwin, and C.G. Nevill-Manning. 1999. "Domain-Specific Keyphrase Extraction", Proceedings 16th International Joint Conference on Artificial Intelligence, pp. 668–673. San Francisco, CA Morgan Kaufmann Publishers.

[4] S.C. Sood, S.H. Owsley, K.J. Hammond, and L. Birnbaum. 2007. "TagAssist: Automatic Tag Suggestion for Blog Posts". Northwestern University. Evanston, IL, USA. http://www.icwsm.org/papers/2--Sood-Owsley-Hammond-Birnbaum.pdf [Last accessed: 23 January 2012]

[5] Technorati. 2006. "Technorati". http://www.technorati.com [Last accessed: 23 January 2012]

[6] J. Goldstein, M. Kantrowtiz, V. Mittal, and J. Carbonell. 1999. "Summarising Text Documents: Sentence Selection and Evaluation Metrics", ACM, pp. 121–128. Language Technologies Institute, Carnegie Mellon University, Pittsburgh, USA.

[7] C. Li, J. Wen, and H. Li. 2003. "Text Classification Using Stochastic Keyword Generation", Twentieth International Conference on Machine Learning (ICML), pp 464–471. Washington DC. https://www.aaai.org/Papers/ICML/2003/ICML03-062.pdf [Last accessed: 23 January 2012]

[8] K. Barker and N. Cornacchia. 2000. "Using Noun Phrase Heads to Extract Document Keyphrases", AI '00: Proceedings of the 13th Biennial Conference of the Canadian Society on Computational Studies of Intelligence. pp. 40–52). London.

[9] J. Wermter and U. Hahn. 2005. "Paradigmatic Modifiability Statistics for the Extraction of Complex Multi- Word Terms". Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP) pp. 843–850. Vancouver Association for Computational Linguistics.

[10] K. Frantziy, S. Ananiadou, and H. Mimaz. 2000. "Automatic Recognition of Multi-Word Terms: the C-value/NC-value Method", International Journal on Digital Libraries , 3 (2), pp. 117-132.

[11] D. Maynard and S. Ananiadou. 2000. "TRUCKS: a model for automatic multi-word term recognition". Journal of Natural Language Processing, 8 (1), pp. 101-125.

[12] D. Maynard and S. Ananiadou. 1999. "Term extraction using a similarity-based approach". Recent Advances in Computational Terminology, pp. 261–278.

[13] A. Joshi and R. Motwani. 2006. "Keyword Generation for Search Engine Advertising", IEEE International Conference on Data Mining, pp. 490–496.

[14] S. Scott and S. Matwin. 1998. "Text Classification Using WordNet Hypernyms", Proceedings of the Association for Computational Linguistics, pp. 38–44.

[15] G.A. Miller, C. Fellbaum, R. Tengi, P. Wakefield, and H. Langone. 2005. "WordNet". Princeton University. http://WordNet.princeton.edu [Last accessed: 23 January 2012]

[16] D. Greenhaus. 2002. "DigiTrad - Digital Tradition Folk Song Server". http://www.mudcat.org/download.cfm [Last accessed: 23 January 2012]

[17] Reuters. 1987. "Reuters-21578 Text Categorisation Collection". http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html [Last accessed: 23 January 2012]

[18] B. Wei, C. Zhang, and M. Ogihara. 2007. "Keyword Generation for Lyrics", Austrian Computer Society (OCG). Comp. Sci. Dept., U. Rochester, USA. http://ismir2007.ismir.net/proceedings/ISMIR2007_p121_wei.pdf [Last accessed: 23 January 2012]

[19] G. Ward. 2000. "Moby Project - Thesaurus". http://icon.shef.ac.uk/Moby/mthes.html [Last accessed: 11 July 2011]

[20] Academics Conferences International. 2009. "ACI E-Journals". http://academic-conferences.org/ejournals.htm [Last accessed: 23 January 2012]

[21] M.F. Porter. 1980. "An algorithm for suffix stripping", Program, 14(3) pp. 130–137.

[22] M.L. McHale. 1998. "A Comparison of WordNet and Roget's Taxonomy for Measuring Semantic Similarity",. http://acl.ldc.upenn.edu/W/W98/W98-0716.pdf [Last accessed: 23 January 2012]