Using Noun Phrase Heads to Extract Document Keyphrases

Ken Barker and Nadia Cornacchia

School of Information and Technology Engineering
University of Ottawa
Ottawa, Canada K1N 6N5
kbarker@site.uottawa.ca, nadiac@mail.intranet.ca

Abstract. Automatically extracting keyphrases from documents is a task with many applications in information retrieval and natural language processing. Document retrieval can be biased towards documents containing relevant keyphrases; documents can be classified or categorized based on their keyphrases; automatic text summarization may extract sentences with high keyphrase scores.

This paper describes a simple system for choosing noun phrases from a document as keyphrases. A noun phrase is chosen based on its length, its frequency and the frequency of its head noun. Noun phrases are extracted from a text using a base noun phrase skimmer and an off-the-shelf online dictionary.

Experiments involving human judges reveal several interesting results: the simple noun phrase-based system performs roughly as well as a state-of-the-art, corpus-trained keyphrase extractor; ratings for *individual* keyphrases do not necessarily correlate with ratings for *sets* of keyphrases for a document; agreement among unbiased judges on the keyphrase rating task is poor.

1 Introduction

Keyphrases for a document are useful for many applications. For text retrieval keyphrases can help narrow search results or rank retrieved documents. They can be used to cluster semantically related documents for the purposes of categorization. They can also be used to guide automatic text summarization.

In our Knowledge Acquisition and Machine Learning group we have been working on a system to generate summaries of documents automatically using a modular design [3, 6, 7]. The modular design divides the summarization task into several parts: keyphrase extraction, text segmentation, segment classification, sentence scoring and selection, etc. For each part, any one of several systems could be plugged in. Furthermore, each module has parameters that could be set empirically. The intent in the project is to use machine learning to configure the system (select modules and set parameter values) to produce the "best" extracted summaries.

To increase flexibility in the configurability of the system, we would ideally have a number of different modules that could be plugged in at the appropriate points in the greater system. For keyphrase extraction, we are using Peter Turney's *Extractor* [10, 11]. As an alternative we decided to build a simple keyphrase extractor in-house as well. The goal was to keep the extractor simple and to apply any linguistic insight we might have to the process.

This paper presents our simple keyphrase extractor (herein referred to as *B&C* for lack of a better name). Section 3 describes how *B&C* extracts noun phrases from a document and scores them based on their frequency and length, taking into account the frequency of noun phrase heads. We have conducted experiments to compare *B&C* to *Extractor* on the level of individual keyphrases as well as on the level of complete, coherent keyphrase sets. The experiments (using human judges) and results are described in sections 4 and 5. Unfortunately, the costs involved in experiments involving human judges limit the scope of experimental evaluation. Our experiments, therefore, are restricted to a comparison of *B&C* and *Extractor* on a small number of documents. Comparisons of *Extractor* to other keyphrase extraction systems can be found in [11]. As usual, many other experiments can be imagined and should be carried out (see section 7).

The experiments suggest that *B&C* and *Extractor* perform differently, but about equally well. Our judges preferred individual keyphrases from *Extractor* more often but complete sets from *B&C* more often. A low degree of agreement between judges prevents sweeping conclusions about the superiority of one system over the other.

2 Related Work

Krulwich & Burkey [9] extract "semantically significant phrases" from documents based on the documents' structural and superficial features. A phrase is some small number of words (one to five, for example). Phrases are chosen using several heuristics. For example, phrases occurring in section headers are candidate significant phrases, as are phrases that are formatted differently than surrounding text. The purpose of extracting such phrases is to attempt to determine a user's interests for information retrieval automatically.

Turney's *Extractor* [11] extracts a small number of keyphrases from documents. Relevant keyphrases are chosen from a list of candidate phrases: all sequences of a small number of words (up to about five) with no intervening stop words or punctuation. The stop word list consists of closed category words (prepositions, pronouns, conjunctions, articles, etc.) as well as a few very general open category words (verbs, nouns, etc.). Keyphrases are selected by scoring candidate phrases on a number of features (such as frequency of the stemmed words in the phrase, length of the phrase, position of the phrase in the document, etc.). Features likely to produce keyphrases that match authors' keyphrases for a document were determined automatically using a genetic algorithm. Although *Extractor* has been evaluated primarily by comparing extracted keyphrases to authors' keyphrases, it has recently been evaluated by human judges as well. The web version of *Extractor* produces a set of keyphrases and the user is invited to mark each keyphrase as "good" or "bad". Results so far give 62% "good" phrases, 18% "bad" and 20% "no opinion".

The *Kea* system [12] uses two features to determine if a candidate phrase is a good keyphrase. Candidate phrases are sequences of consecutive words (usually no more than three) with no intervening phrase boundary indicators (such as punctuation). Proper names and phrases beginning or ending with stop words are excluded. Subphrases of a candidate phrase may appear as separate candidate phrases. The first feature used in selecting keyphrases is TF×IDF (term frequency×inverse document

frequency), which favours phrases that appear frequently in the current document and infrequently in general usage. Frequency in general usage is determined by frequency in a "global corpus" (a large, general purpose corpus). The second feature is distance from the beginning of the document. Feature values are calculated for all candidate phrases in documents in a training corpus (documents for which authors' keyphrases are available). Each candidate phrase is then marked as a positive example if it is among the author's keyphrases or as a negative example. A Naïve Bayes technique is used to assign weights to the features, based on the feature values of the positive and negative examples. Experiments on unseen documents compare extracted keyphrases to authors' keyphrases. The performance is statistically equivalent [11] to *Extractor*. The *Kea* group recognizes the limitations of evaluating keyphrases relative to author-supplied keyphrases and plans to do further evaluation using human judges to rate "how well a set of extracted keyphrases summarize a particular document" ([12]).

3 Extracting Keyphrases

Our system for extracting keyphrases from documents proceeds in three steps: it skims a document for base noun phrases; it assigns scores to noun phrases based on frequency and length; it filters some noise from the set of top scoring keyphrases.

3.1 Skimming for Base Noun Phrases

Most of our work in knowledge acquisition from texts processes parse trees generated by the DIPETT parser [8]. For the task of extracting keyphrases, full, detailed parses of complete English sentences are not needed. To avoid the overhead associated with deep parsing, we decided to implement a simple base noun phrase skimmer instead.

A base noun phrase is a non-recursive structure consisting of a head noun and zero or more premodifying adjectives and/or nouns. The base noun phrase does not include noun phrase postmodifiers such as prepositional phrases or relative clauses. A base noun phrase skimmer proceeds through a text word-by-word looking for sequences of nouns and adjectives ending with a noun and surrounded by non-noun/adjectives.[1]

Such a skimmer requires knowledge of the parts of speech of the words in the text. One possibility would be to tag the text using a tagger (such as the widely used Brill tagger [4]. A tagger assigns the most likely single part of speech tag (noun, adjective, verb, etc.) to each word in a sentence. We decided to use a simple dictionary lookup instead. The main advantage of a dictionary lookup is that our online dictionaries list the root form of each word, allowing us to treat such phrases as *good schema* and *better schemata* as instances of the same root phrase.

The skimmer uses two dictionaries: our own DIPETT dictionary, which is fairly complete for closed class words (articles, prepositions, conjunctions, etc.); and the Collins wordlist, a large list of English words with all possible parts of speech for each word (and then some). If a word appears in DIPETT's dictionary as a closed

[1] More sophistication is possible by looking specifically for noun phrase "surrounders" such as articles, prepositions, verbs, etc, or by allowing other elements in the base noun phrase such as possessives, conjoined premodifiers, etc.

category word, it is tagged <closed>. Otherwise, if the word can be a noun according to Collins, it is tagged <noun>; if it can be an adjective it is tagged <adjective>. The check in DIPETT's dictionary is required since the Collins list contains some questionable entries (such as *a* as a preposition, noun and article).

3.2 Counting Noun Phrases

In this section we describe the formula for choosing noun phrases (NPs) as keyphrases. Systems that choose keyphrases on frequency alone take the most frequently occurring phrases in a document. Our decision to take the frequency of a noun phrase's head noun into consideration was based on the following observations:

1. Longer noun phrases (with more premodifiers) are more specific and may be more relevant to a particular document than more general, shorter noun phrases.
2. In the interest of economy (and ease on the reader), long noun phrases are usually not repeated frequently in a document. For example, an article about *the Canadian Space Agency* may use that phrase once, with subsequent references reduced to *the Space Agency* or even *the Agency*.

Here is our algorithm for assigning scores to noun phrases:

1. $freq_H$ = the number of times noun H appears in the document as the head of a noun phrase
2. take the top N heads $H_1..H_N$ with the highest $freq_H$; discard the rest of the heads
3. for each head $H_i \in H_1..H_N$
   i) recover all complete noun phrases $NP_1..NP_M$ having H_i as head
   ii) for each $NP_j \in NP_1..NP_M$ calculate NP_j's score as its frequency times its length (in words)
4. keep the top K highest scoring noun phrases as keyphrase candidates for the document

In steps 1 and 2, discarding relatively infrequently occurring heads allows less frequent noun phrases (with frequent heads) to compete in steps 3 and 4. For example, head H_1 may occur more frequently than any of the complete noun phrases having H_2 as head. But if H_2 occurs as head more frequently than H_1, H_1 may be discarded in favour of H_2's noun phrases.

The algorithm allows for many variations, some of which we considered. For example, in step 3 we considered taking exactly one NP (the top scoring NP) for each of the N most frequent heads, disallowing more than one keyphrase with the same head. We decided, however, to allow multiple keyphrases with the same head. One can imagine documents for which *laser printer* and *colour printer* would both be useful keyphrases. Such biases should be experimentally validated.

The thresholds N and K should be set according to heuristics (based on document length or as a percentage of distinct heads), or set by the user as a parameter, or determined empirically. For example, for all the noun phrases in step 3, if there is a gap in the scores between the higher scoring and lower scoring NPs, the threshold could be set at the gap. These thresholds could also be set according to results of

evaluations: is there a threshold beyond which keyphrases rate poorly. For the experiments we set *N* and *K* arbitrarily to the maximum number of keyphrases produced by *Extractor* for our test documents (twelve). Setting *N* and *K* higher would produce more low-scoring keyphrases (*i.e.*, shorter phrases and those with less frequent heads).

3.3 Postprocessing

Once the algorithm has produced the top *K* keyphrase candidates for a document, we apply two simple postprocessing filters: remove single letter keyphrases; remove wholly-contained subphrases.

Single letter keyphrases are an artifact of the Collins dictionary lookup, and could be filtered out prior to keyphrase selection. Normally one might consider ignoring single letter words altogether (as *Extractor* does). Previous investigations into the semantics of noun phrases [1], however, suggest that some single letter words are relevant in noun phrases (*SCSI D connector*, *Y chromosome*, *John F. Kennedy*, etc.).[2]

Removing wholly-contained subphrases is intended to prevent both a phrase and a generalization of the phrase (a subphrase) from appearing as keyphrases when both have high scores (*e.g.*, *theoretical Computer Science* and *Computer Science*). It is easy to invent examples where both a phrase and a wholly-contained subphrase would make good keyphrases for a document. But in general, given a coherent set of keyphrases, we decided that subphrases would contribute little. This decision could be added to the growing list of choices to be validated experimentally.

4 Experiments

4.1 Using Human Judges

We conducted two experiments using human judges to compare our keyphrases to those produced by *Extractor*. Our previous experiences using human judges have taught us that using human judges should be avoided. Making the necessary judgments is usually a difficult, time-consuming and energy-consuming process. Drawing statistically significant conclusions from such experiments can be difficult because there is a limit on the amount of data that can be collected. Nonetheless, *automatic* evaluation of keyphrases would require some gold standard set of keyphrases for a document, and these simply do not exist. Other researchers [11, 12] have used author's keyphrases as a gold standard. There are several problems with using author's keyphrases:

- author's keyphrases are not always taken from the text (in experiments reported in [11], 75% of them are)
- author's keyphrases are often restricted to a very small number of phrases (two or three, for example)

---

[2] Some writers may have a preference for *D-connector* or *Y-chromosome*, but hyphenating is far from universal and cannot be assumed by systems dealing with unrestricted text.

- author's keyphrases are often chosen for a specific purpose (for classification according to an existing set of keyphrases, to steer review of a document, to distinguish a document from others in one specific collection of documents, etc.)
- author's keyphrases are usually only available for the very few kinds of documents for which authors supply keyphrases; these are exactly the kinds of documents for which we have no need for automatically generated keyphrases.

The judges in our experiments were university faculty, postdoctoral fellows, graduate students and AI researchers. It is possible that this community has preconceptions about what makes a good keyphrase.

4.2 Choosing Documents

Having decided to enlist human judges in our evaluation, we had to choose a small number of documents. To ensure a fair comparison to *Extractor*, we chose nine documents from corpora used in training and testing *Extractor*. The nine consisted of three documents chosen at random from each of three of the five *Extractor* corpora. One of these corpora was used in training *Extractor*, all three were used in testing it. To the nine *Extractor* documents we added four documents from different domains with no particular consideration given to subject matter or style.

For each of the thirteen documents, we extracted keyphrases using *Extractor* and *B&C*. The documents were then given to twelve judges who were asked to read them. No further instructions about what to look for in the documents were given, though the judges knew that they would have to rate keyphrases for them.

After reading the documents, the judges were asked to rate keyphrases in two separate experiments: one to rate individual keyphrases for each document, and one to compare *Extractor*'s complete set of keyphrases for each document to *B&C*'s set of keyphrases for each document.

4.3 Rating Individual Keyphrases

For each document the judges were given a single list of keyphrases in no particular order. The list was the union of keyphrases from *Extractor* and *B&C* keyphrases (duplicates removed). Judges rated each keyphrase as "good", "so-so" or "bad", with minimal instructions about the definitions of those terms (to avoid biasing them toward a particular kind of keyphrase).

4.4 Comparing Sets of Keyphrases

The second experiment had the judges compare *Extractor*'s keyphrases to *B&C* keyphrases for each document. The two keyphrase sets were normalized (converting all characters to lower case, for example) and presented to the judges in random order.

The judges were instructed to consider each of the two keyphrase sets as a coherent whole and to compare them to each other. They were told to mark as preferred the set that they felt better represented the content of the document for any

reason. They were also given the option to mark neither as preferred if they felt that there was no significant difference between the two.

5 Evaluation

Here we present the results of the experiments from three points of view: the straight numbers for the two experiments; the degree to which the human judges agreed amongst each other for each experiment; the correlation between the two experiments. We leave discussion of the results to section 6.

5.1 Individual Keyphrase Ratings

For each keyphrase, judges' ratings were converted to numeric scores by assigning 2 points for a "good" rating, 1 point for a "so-so" rating and 0 points for a "bad" rating. The score for each keyphrase was calculated simply as the sum of the scores from all twelve judges. We then assigned a score to *Extractor* and *B&C* for each document by taking the sum of the keyphrase scores for keyphrases produced by the system divided by the total number of keyphrases produced by the system for the document. The normalized results appear in Table 1.

Table 1. Document scores based on individual keyphrase ratings

	Average document score	Standard deviation
Extractor	0.56	0.11
B&C	0.47	0.10

On average, *Extractor* produced 6.2 keyphrases per document, *B&C* produced 9.1. Forcing *B&C* to produce more keyphrases may explain the lower average document score, assuming the extra keyphrases were the ones rated lower by the judges. The average length of keyphrases was 1.7 words for *Extractor* and 1.9 words for *B&C*. Fig. 1 shows the average proportion of keyphrases of varying length per document. The figure clearly illustrates *B&C*'s bias toward longer phrases.

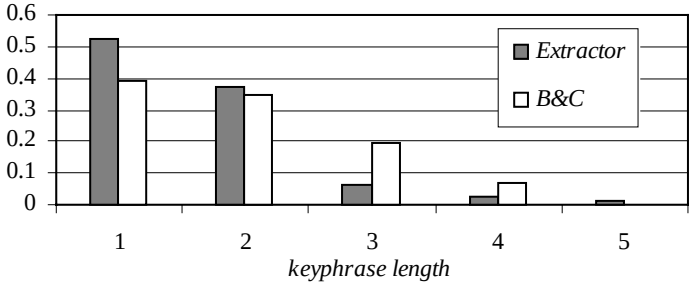

Fig. 1. Average proportion of keyphrases of various lengths (per document)

5.2 Side-by-Side Comparison of Keyphrase Sets

For the side-by-side comparison experiment we counted the number of times judges preferred *Extractor*'s set of keyphrases, the number of times judges preferred *B&C* sets and the number of times neither set was preferred. The results appear in Table 2.

Table 2. Number of times complete sets of keyphrases were preferred

	Times preferred
Extractor	61/156 (0.39)
B&C	74/156 (0.47)
neither	21/156 (0.13)

Judges preferred the longer of the two sets of keyphrases 39% of the time and the shorter set 40% of the time. For one document, both *Extractor* and *B&C* produced the same number of keyphrases.

5.3 Inter-Judge Agreement

In any experiment involving human judgments there must be some analysis of the degree to which the judges agree.

For the side-by-side comparison judges agreed on their preferences 43% of the time. Of course, we would expect them to agree some of the time by chance alone. To correct for chance, we measured the inter-judge agreement using the Kappa Statistic [5], which is widely used in the field of content analysis and growing in popularity in the field of natural language processing. Briefly, the Kappa Statistic is a measure of agreement between two judges that takes into account chance. Kappa is defined as:

$$\kappa = \frac{P(A) - P(E)}{1 - P(E)}$$

P(A) is the number of times the two judges agree relative to the total number of judgments; P(E) is the proportion of times the judges are expected to agree by chance. $\kappa = 0$ indicates chance agreement. Notice that the definition allows for negative κ values when judges agree less often than would be expected by chance. Normally, P(E) would be set to the number of combinations of identical judgments (agreements) divided by the total number of combinations of judgments (in our case, 1/3):

judge1	Ex'tor	Ex'tor	Ex'tor	B&C	B&C	B&C	neither	neither	neither
judge2	Ex'tor	B&C	neither	Ex'tor	B&C	neither	Ex'tor	B&C	neither
	1/9	1/9	1/9	1/9	1/9	1/9	1/9	1/9	1/9
	1/9		+		1/9		+		1/9

This calculation is based on the assumption that a judge is equally likely to choose *Extractor*, *B&C* or *neither*. In fact, there was definite reticence among the judges to choose neither. The observed likelihood that a judge would choose neither was about

0.13, making the probability that a judge would choose *Extractor* or *B&C* 0.43 each. So our estimate for P(E) is the probability that both judges choose neither (0.13×0.13) plus the probability that both choose *Extractor* (0.43×0.43) plus the probability that both choose *B&C* (0.43×0.43): P(E) = 0.39. That is, given the judges' avoidance of *neither*, we expect them to agree by chance somewhat more often than 1/3 of the time. The κ values reported here are lower than if we had used P(E) = 1/3, though not by much. Table 3 shows the κ values for each pair of judges on the side-by-side comparison experiment.

Table 3. The κ values for each pair of judges (κ > 0 in boldface)

	A	B	C	D	E	F	G	H	I	J	K	L
A		-0.01	-0.01	-0.14	**0.49**	-0.01	-0.01	-0.01	**0.49**	**0.37**	-0.01	-0.14
B			**0.11**	-0.27	**0.24**	**0.24**	**0.24**	-0.01	**0.24**	**0.11**	-0.14	-0.14
C				-0.27	**0.24**	-0.01	-0.39	-0.27	-0.14	**0.37**	-0.01	**0.11**
D					-0.39	**0.24**	-0.01	**0.24**	**0.24**	**0.24**	**0.11**	-0.27
E						**0.11**	-0.14	-0.14	**0.24**	**0.24**	**0.11**	**0.24**
F							**0.24**	**0.24**	**0.24**	**0.11**	**0.37**	-0.39
G								**0.75**	**0.24**	-0.14	**0.11**	-0.27
H									**0.24**	**0.11**	**0.37**	-0.39
I										**0.11**	**0.11**	-0.27
J											**0.11**	-0.14
K												-0.39
L												

The Kappa values are spectacularly low. The average κ is 0.06, meaning that, on average, the judges agree only about as much as can be expected by chance (κ = 0).[3]

If we isolate the situations where judges *did* agree, we can compare the number of agreements on *Extractor* keyphrase sets to *B&C* keyphrase sets. Table 4 shows what the judges were agreeing on when they agreed.

Table 4. Distribution of agreements among the three categories

	Number of agreements
Extractor	142 (0.39)
B&C	210 (0.57)
neither	14 (0.04)
total	366

For the individual keyphrase rating experiment it is somewhat more difficult to determine the degree to which judges agreed. Judges agreed on the rating of

[3] Setting P(E) to 1/3 gives an average κ of 0.14, which still indicates very little agreement beyond what would be expected by chance.

individual keyphrases (as "good", "so-so" or "bad") 52% of the time. But we would expect them to agree by chance 1/3 of the time (assuming a single keyphrase is equally likely "good", "so-so" or "bad"). For individual keyphrases, the average Kappa is 0.27, which is quite low.

But Kappa is not necessarily a good indicator of agreement for individual keyphrase ratings. Given the three-point scale, it is possible that judges rated keyphrases for a document similarly, but not identically. For example, one judge might rate half the phrases "so-so" and the other half "bad". A second judge might rate all of judge A's "so-so" keyphrases "good" and all of A's "bad" keyphrases "so-so". These two judges would have the minimum Kappa of –0.5, even though their relative ratings were similar. Kappa requires absolute agreement of judges.

To account for relative similarities between judges' individual ratings of keyphrases, we calculated the correlative coefficient[4] between the ratings of each pair of judges. The average coefficient was 0.47, indicating moderate agreement among judges on the quality of a keyphrase relative to other keyphrases. Individual coefficients appear in Table 5.

Table 5. Correlative coefficients for each pair of judges

	A	B	C	D	E	F	G	H	I	J	K	L
A		0.53	0.49	0.44	0.37	0.60	0.47	0.42	0.40	0.49	0.32	0.35
B			0.46	0.47	0.51	0.59	0.44	0.38	0.43	0.47	0.45	0.55
C				0.52	0.57	0.63	0.47	0.44	0.52	0.51	0.54	0.52
D					0.39	0.56	0.47	0.46	0.47	0.52	0.39	0.36
E						0.53	0.35	0.32	0.47	0.39	0.56	0.55
F							0.49	0.43	0.45	0.53	0.50	0.48
G								0.47	0.49	0.50	0.36	0.51
H									0.55	0.47	0.39	0.40
I										0.37	0.49	0.51
J											0.31	0.44
K												0.47
L												

5.4 Correlation Between the Two Evaluations

The purpose of conducting both of the experiments we have described was in part to investigate the connection between the quality of individual keyphrases and sets of keyphrases. For example, the phrase *alien abduction experience* may be considered a "good" keyphrase for a particular document; the keyphrase *experience* may be

[4] The correlative coefficient is a measure to which the differences among data points in one list are similar to the differences among data points in a second list, even if the data points are scaled differently in each list. A correlative coefficient of 0 indicates no relationship between the data in the two lists. Correlative coefficients of 1 and –1 indicate direct and inverse relationships between the data points in the two lists.

considered too general and rated "bad". But if the set of keyphrases containing *experience* also contains *alien* and *abduction*, the set as a whole might be considered just as good as the set containing *alien abduction experience*.

Similarly, a set with many weak keyphrases and many good keyphrases may rate poorly in the document score that normalizes for the number of keyphrases in the set. But in an application where recall of keyphrases is more important than precision, the set might be useful.

To measure the correspondence between each judge's individual keyphrase ratings and side-by-side preferences, we used individual keyphrase ratings to predict which set a judge would prefer. Here is the simple formula to predict if judge *J* will prefer *Extractor* or *B&C* (or neither) for document *D*. As previously, keyphrases rated "good" are given 2 points, etc.

1. For each keyphrase *K* in *D*
   if *K* is in *Extractor*'s set, add *K*'s points from *J*'s rating to *Extractor*'s score for *D*
   if *K* is in *B&C*'s set, add *K*'s points from *J*'s rating to *B&C*'s score for *D*
2. Predict that *J* will prefer whichever set has the greater score for *D* (or neither, if the two scores are equal

We then counted the number of times the prediction for each judge on each document matched the judge's actual preference. The proportions of matches are shown in Table 6. The average was 0.51. Again we would expect that by chance a judge's individual keyphrase scores would match that judge's preferred set some of the time. The average Kappa measuring the agreement between a judge's keyphrase-based document scores and the judge's stated preference is 0.21.

Table 6. Number of times individual keyphrase ratings predict keyphrase set preference

Judge	A	B	C	D	E	F	G	H	I	J	K	L
keyphrase ratings predict set preference	0.46	0.69	0.31	0.54	0.31	0.62	0.46	0.54	0.69	0.69	0.62	0.15

6 Discussion

The judges seemed on average to assign higher scores to individual keyphrases produced by *Extractor*, though not significantly higher scores. Normalizing the document scores by dividing by the number of keyphrases produced should correct for any advantage *B&C* gained by producing more keyphrases. The fact that judges preferred short sets as often as long sets (40% vs. 39%) suggests that having more keyphrases was not necessarily an advantage to *B&C*.

For the side-by-side comparison of keyphrase sets, judges more often preferred *B&C* keyphrases, despite the fact that *Extractor*'s individual keyphrase ratings were higher and the fact that *B&C*'s keyphrase sets were longer. This can be accounted for by looking at inter-judge agreement on the side-by-side comparisons: when more

judges chose *B&C* keyphrase sets, they chose them as a larger majority; when more judges chose *Extractor* keyphrase sets, there was more disagreement.

The discrepancy between the results of the two experiments is supported by the weak correlation between individual keyphrase-based document scores and keyphrase set preferences. Judges did not prefer keyphrase sets based simply on the individual keyphrases they contained. A set of keyphrases is somehow more than the sum of its individual keyphrases.

7 Future Considerations

The *B&C* system is overly simplistic. The noun phrase skimmer could be improved (or we could go back to DIPETT for better noun phrases). System parameters (such as the number of heads considered (*N*) and the number of keyphrases generated (*K*)) should be set according to empirical observations, perhaps as the result of a machine learning experiment. New parameters, such as phrase position in the document, could be added. One side-trip experiment showed little difference between keyphrases extracted from the whole document and those extracted from the first half only.

More evaluation is also needed. The low inter-judge agreement (due in part to the unconstrained nature of our experiments) suggests that a more directed experiment is required: one with a particular application of keyphrases in mind. Other experiments are required to evaluate design decisions in isolation (such as the decision to allow multiple keyphrases with the same noun phrase head).

A more ambitious project would be to plug the different keyphrase extractors into a larger system. How would different keyphrases affect sentence extraction in a text summarization system, for example? It would also be interesting to adjust the keyphrase selection algorithm to allow for compound heads: *theoretical natural language processing* and *empirical natural language processing* are kinds of *natural language processing*, not just kinds of *processing*.

8 Conclusions

In this paper we have presented a simple system for extracting keyphrases automatically from documents. It requires no training and makes use of publicly available lexical resources only. Despite its lack of sophistication, it appears to perform no worse than the state-of-the-art, trained *Extractor* system in experiments involving human judges.

More importantly, however, experiments show that judges do not necessarily consider the quality of *sets* of keyphrases as a simple function of the quality of *individual* keyphrases. This suggests that neither experiments involving the rating of individual keyphrases only (as reported in [11]) nor experiments rating the quality of sets of keyphrases only (as proposed in [12]) are sufficient for evaluating the performance of a keyphrase extraction system.

Acknowledgments

This research is supported by the Natural Sciences and Engineering Research Council of Canada. The authors would like to thank the Knowledge Acquisition and Machine Learning Group at the University of Ottawa for helpful comments and suggestions. We are grateful to our twelve human judges for their patience and input. We would also like to thank Peter Turney for providing us with his corpora and for generously offering us copies of his most recent soon-to-be-published papers. Finally, the insightful suggestions of anonymous reviewers helped improve this paper.

References

1. Barker, Ken & Stan Szpakowicz (1998). "Semi-Automatic Recognition of Noun Modifier Relationships." *Proceedings of COLING-ACL '98*. Montréal, 96-102.
2. Barker, Ken, Sylvain Delisle & Stan Szpakowicz (1998). "Test-driving TANKA: Evaluating a Semi-Automatic System of Text Analysis for Knowledge Acquisition." *Proceedings of the Twelfth Canadian Conference on Artificial Intelligence* (LNAI 1418), Vancouver. 60-71.
3. Barker, Ken, Yllias Chali, Terry Copeck, Stan Matwin & Stan Szpakowicz (1998). "The Design of a Configurable Text Summarization System". TR-98-04, School of Information Technology and Engineering, University of Ottawa.
4. Brill, Eric (1995). "Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging." *Computational Linguistics* 21(4), December, 1995. 543-566.
5. Carletta, Jean (1996). "Assessing Agreement on Classification Tasks: The Kappa Statistic." *Computational Linguistics* 22(2), June, 1996. 249-254.
6. Chali, Yllias, Stan Matwin & Stan Szpakowicz (1999) "Query-Biased Text Summarization as a Question-Answering Technique". *Proceedings of the AAAI Fall Symposium Workshop on Question-Answering Systems*. Cape Cod, Massachusetts, November 1999.
7. Delannoy, Jean-François, Ken Barker, Terry Copeck, Martin Laplante, Stan Matwin & Stan Szpakowicz (1998) "Flexible Summarization". *AAAI Spring Symposium Workshop on Intelligent Text Summarization*. Stanford, March, 1998.
8. Delisle, Sylvain (1994). "Text processing without A-Priori Domain Knowledge: Semi-Automatic Linguistic analysis for Incremental Knowledge Acquisition." Ph.D. thesis, TR-94-02, Department of Computer Science, University of Ottawa.
9. Krulwich, Bruce & Chad Burkey (1996). "Learning user information interests through the extraction of semantically significant phrases." In M. Hearst and H. Hirsh, editors, *AAAI 1996 Spring Symposium on Machine Learning in Information Access*. California: AAAI Press.
10. Turney, Peter D. (1999). "Learning to Extract Keyphrases from Text." National Research Council, Institute for Information Technology, Technical Report ERB-1057.
11. Turney, Peter D. (2000). "Learning Algorithms for Keyphrase Extraction." *Information Retrieval*. To appear.
12. Witten, Ian H., Gordon W. Paynter, Eibe Frank, Carl Gutwin & Craig G. Nevill-Manning (1999). "KEA: Practical Automatic Keyphrase Extraction." *Proceedings of the Fourth ACM Conference on Digital Libraries*.