

PhraseRate: An HTML Keyphrase Extractor *

J.B. Keith Humphreys
Dept. of Computer Science
University of California, Riverside
Riverside, California 92507
keith@cs.ucr.edu

12 June 2002

Abstract

A standard feature in cataloging documents is the list of keywords. When the source documents are web pages, we can attempt to aid the cataloger by analyzing the page and presenting relevant support material. Since the keywords that occur in a document generally occur in keyphrases, and keyphrases provide contextual material for reviewing candidate keywords, they are a natural aggregate to extract and present to the cataloger.

This paper describes PhraseRate, which is an interactive aid for keyword extraction, designed to assist human classifiers in the Infomine Project (<http://infomine.ucr.edu/>). In particular, it introduces a novel keyphrase extraction heuristic for web pages which requires no training, but instead is based on the assumption that most well written webpages “suggest” keyphrases based on their internal structure. It is very fast, flexible, and its results compare favorably with the state of the art in keyphrase extraction.

1 INTRODUCTION

Various organizations provide extensive directories to web resources, which are cataloged by human adders. To facilitate this process, a suite of programs are desired to assist the adders. Keyword-document association aids typically fall into two general categories:

Assigned keywords: keywords from a controlled keyword collection are matched to the document. This is essentially a classification process.

Extracted keywords: where words from the source document are the candidates to be selected from. This essentially is a rating process.

With diverse collections of documents, keyword extraction aids have the desired flexibility to conform to the individuality among documents.

For an online keyword extraction routine, desired features are:

Speed, since it is one of many components presenting results to a waiting user.

Coverage of the top keywords is more important than precision since there is a human selector.

Friendly Interface: The information should be easy and quick to digest. If it helps to provide an orientation to the contents of the site, that is a bonus.

Flexible enough to work with a huge spectrum of pages: the Internet.

Open Sourced: Having an open source license allows a public benefit.

*This work supported in part by the National Science Foundation under grant CCR-9988360 and in part by the US Department of Education under grant P116B980450.

A natural aggregate for extraction and presentation to the adder is the keyphrase. A keyphrase consists of a short sequence of words, generally a brief sentence segment, that encapsulates a key subject of its source document. Well chosen keyphrases can succinctly capture a document's subject content, and have an expressive power over keywords in the same manner that a sentence expresses much more than its set of words. For example, consider {broad, issues, of, policy, public, range} as opposed to "broad range of public policy issues".

Some benefits of using keyphrases for interactive keyword extraction include:

- Coverage: Keywords generally appear in keyphrases.
- Context: Often keyphrases contain keywords in their most significant context within the document.
- Brevity: Multiple keyword candidates can appear in a single keyphrase,
- Readability: A list of keyphrases tends to be more readable than a list of keywords.
- Summarization: Those keyphrases that don't contain keywords still provide succinct summarizations of the document, helping the cataloger gain a quick perspective on the document's subject.

PhraseRate's goal is to extract keyphrases from an HTML document, which are then presented to the adder via a clickable menu. Suitable keyphrases can be selected and then edited for keywords.

By using the internal structure of webpages, PhraseRate attempts to determine keyphrases that reflect the document's subject. This presupposes that quality pages are, by and large, structured in such a way that their subject content is discernible via the page structure considered in isolation. Understanding that this is not always the case, the procedure also forms a confidence rating and activates various bolstering attempts if the document is deemed deficient. The two primary salvage methods are:

Spacewise: An attempt is made to find the site's "self-description" page. The local hyper-links are rated as to leading to a likely candidate, and then the best candidate is selected, provided its rating is above an acceptance threshold. Its content are then integrated.

Timewise: From a sequence of "snapshots" of a webpage, a *weighted exponential average* time analysis filters out transient subjects. This has worked out quite well on many types of dynamic pages.

This paper focuses only on the the core keyphrase extraction algorithms.

While PhraseRate was designed as a keyword extraction aid, there are other uses for keyphrases, such as document summarizers. For an extensive list of reasonable uses for keyphrases beyond being keyword fodder, see [Turney97].

2 BACKGROUND

The standard/classical method for keyword/keyphrase extraction is based on relative frequency analysis: Word frequency statistics are gathered from a corpus and stored. (Generally the stop words are first removed and then some form of stemming is imposed.) When confronted with a new document, a word frequency vector is constructed and then normalized relative to the corpus frequency. The extracted keywords are those that have the highest relative frequency.

This general approach has a number of deficiencies, including:

- Relative frequency analysis looks for "rare" words common in a document, and attempts to extract aspects of the document using essentially an eccentricity measure. While this tends to accentuate the unique aspects of the document's content, it makes the assumption that the document's focus is centered around its uniqueness. Results from this type of analysis tend to yield peculiarities rather than content. But the subject of a document is not what is distinct about it, but instead what is central to the document!
- Most corpora are rather limited compared to the diversity of the web: what is rare in a corpus might not be rare elsewhere. It is difficult to get a representative sample that covers the diversity of word usage found throughout the web. Sampling methods tend to work best with closed systems.

More recently, learning programs have been overlaid on these methods. Extractor [Tunney97] has a trainer that uses complex genetic algorithms. Later came Kea [Witten99], which used Naive Bayesian techniques

on a corpus of documents which had author selected keywords. This was then used to help provide keyword/keyphrase capabilities for the New Zealand Digital Library (NZDL).

At the time of PhraseRate’s development (summer ’99), we were unaware of Extractor, and as it is proprietary and uses training from a corpus, it would have been unsuitable. KEA, which is public, had not yet surfaced and again is based on a corpus. While these don’t form background material for PhraseRate’s development, they are important for comparison and hence they are lightly covered in the test section.

3 FINDING KEYPHRASES

As we were cataloging only “quality” pages of academic interest, it seemed that they would typically contain sufficient structure and organization to guide the selection of keyphrases, independent of the nature of other related pages. Target pages were expected to be normally well written, subject oriented, and in particular:

- The beginning of the page would have an overview of the contents, that is, a topic introduction.
- The verbiage in the page would focus on a topic, and so keywords and keyphrases should repeat a lot.
- Key subject ideas would more likely be emphasized with HTML emphasis markups.
- Markups would not be sporadically located inside phrases.

These properties formed the basic assumptions on the material for which PhraseRate was to be designed.

3.1 Design Constraints

The following (non-independent) constraints and considerations served as a guide in the formulation of the phrase extraction process:

1. The program should not depend on training:
There is a large variety of topics and writing styles among webpages as well as emerging topics, words, and phrases in research. It would be difficult to provide an accurate and flexible phrase extractor in an open ended environment derived from a closed set of training examples. So a design based more on the intrinsic properties of the webpages was desired. (If training was to be used, it should be for higher level parameter tuning only.)
2. Candidates for phrases should be from 2 to 5 significant words long and should not cross various blocking structures such as punctuation and HTML formatting blocks. Longer phrases *supporting* keywords are rather rare and were considered to be an excessive computational expenditure.
3. Extraction should be accomplished by rating and filtering.
4. The rating of a phrase should be in part dependent on the the number of instances in the document: After evaluating individual phrase instances, we fold in their weights to form a global evaluation. We want to emphasize repeated phrases subject to the other conditions.
5. The rating of a phrase instance should be in part dependent on the strength (weights) of its constituent words,
 - Monotonic ranking of phrase instances:
If words $\{x_1, \dots, x_n\}$ and $\{y_1, \dots, y_n\}$ satisfy $(\forall i \leq n)[W(x_i) \leq W(y_i)]$, then the phrase instance ranks should satisfy $PIR(x_1 \dots x_n) \leq PIR(y_1 \dots y_n)$.as well as their sequencing:
 - Emphasize uniformly good sequences of words:
A highly rated word accompanied by a pair of weak words does not form a distinguished sequence. For the group as a whole to be considered, all the individuals should be distinguished to a fair degree.
 - Phrase rating length coordination:
We don’t want to consider long phrases of trivial words to be somehow better than a good pair of words. On the other hand, a string of uniformly emphasized words is probably a good candidate, so we want to encourage longer strings somewhat.
6. The weight of an instance of a word was to be determined by:

- HTML mark up, in a nested fashion,
 - location from the start, up to an “introduction-limit” distance. That is, an emphasis function that decayed up to a fixed level was to magnify the importance of introductory words.
 - capitalization (modestly),
7. The global (documentwise) weight of a word was taken to be the accumulation of the weight of all its instances, and hence be additive.
8. Last, we consider the following reasonable *phrase ranking scale invariance* conditions:
- i.* Given the additive nature of rated word occurrences, if the word weights are rescaled by a constant, then the ordering of the rated phrases should remain unchanged.
 - i'.* This is satisfied by the natural *Homogeneity property*: If all the weighting values of the component words are rescaled by a constant $c > 0$, then the rating of the phrase are likewise scaled by c .
 - ii.* If a document was extended in length with the same relative phrase/word distribution as the initial document, then the ordering of the rated phrases should remain unchanged, modulo the introduction effects.
 - ii'.* This is satisfied by the natural property:
If a document was extended with the same relative phrase/word distribution as the initial document, then the ratings of all phrases are also scaled by a constant factor, modulo the introduction effects.
- Under the additive nature of rated word occurrences, these properties are related. For example if we double the length of a document, then the word weights double...

3.2 Our Approach: PhraseRate

A brief overview of PhraseRate is as follows:

WordRate: A lexer which processes the webpage, outputting important HTML structure tokens and rated word instances.

DocRate: This module consumes the WordRate output then forms the collection of phrases and provides a global (documentwise) evaluation of word strengths.

RatePhrases: It processes the DocRate data to produce rated phrases.

Selector: Selects the top entries from RatePhrases subject to redundancy (subphrase) elimination.

Note: various support modules/programs that perform tasks like fetching pages and such are routine and ignored in this paper.

3.2.1 WordRate

We begin with the HTML page being submitted to WordRate, which is a lexer that processes word instances. Wordrate’s goal is to:

- Ignore irrelevant HTML sections, such as comments, scripts, applets, ...
- Keep a running sense of the emphasis of the HTML mark up.
For example, if the program encounters a `<big>` tag, it then considers the succeeding text more emphasized. If it then comes across a `` tag, it further increases the running word emphasis. After reaching the matching `` tag, the initial `` weighting effect is removed, and then when the `</big>` tag is encountered its weight is further removed.
- Mark all block breaking structures and output a block break token.
- Identify stop words and emit a stop word token.
- Identify “gluons”, which are words that in themselves “carry no weight”, but are often used in phrases as “glue”, such as prepositions. The word is emitted with its gluon token. These words are usually stop words, but are treated differently for the purpose of forming phrases. As an example, consider “birds of prey” or “green eggs and ham”.¹

¹This useful observation was noticed independently by Marek Chrobak, who contributed the terminology which stuck.

```

<big>   There   was   a   farmer   had   a   <em>   dog   </em>
        $     $   $     2     $     $           3

 and
  I           2           i   P

<a href="bingo.htm" > Bingo </a> was his name-o </big>
  U           3     u   $     $     2

#: Regular word      -: Block Break      [U,u]: Url Delimiters
P: Gluon             $: Stop word          [I,i]: Image Delimiters

```

Figure 1: Sing Along With WordRate

```

(resources,10330)   (resource,6514)   (sciences,5227)
(search,9813)     (internet,6475)  (science,4958)
(scholarly,8118) (collections,5960) (what's,4892)
(library,7260)   (electronic,5699) (information,4767)

```

Figure 2: Top Weighted Words from Infomine's Homepage.

- Identify regular words. These are emitted with the current running weight attached.
- Identify special HTML sections such as title, various useful meta sections, and img alt text. Text tokens in these sections are bracketed by section identifier tokens, to allow special handling.
- Warnings and verbose comments from the lexer are passed with warning and verbose tokens, for debugging and development.

See Figure 1 for a brief display of WordRate in action.

3.2.2 DocRate

DocRate consumes the content of the lexer and

- adjusts the weights by a decaying boost factor, which emphasizes the introductory text
- does special processing on word instances from the special HTML sections. (For example meta keywords are given more weight if meta keywords are sparse.)
- determines the accumulated weight of each word,
- forms phrases of length 2 to 5 words, not counting gluons. No phrase is allowed to start or end with a gluon.

Figure 2 shows the top rated words and Figure 3 some phrases from Infomine's homepage.

Figure 3: Phrases with # Occurrences.

```

directories of researchers;1   educational technology;1
distance learning;1          electronic books;2
educational resources;1      electronic journals;3

```

3.2.3 RatePhrase

The rating is described in steps, starting with global word rankings that are weighted in an additive fashion in DocRate. Afterwards we show that the fairly extensive list of design constraints is satisfied.

To rate an individual phrase *instance* $w_1 \dots w_k$ with word weights $x_i := W(w_i)$, we use the harmonic mean:

$$H(x_1, \dots, x_k) = \frac{k}{\frac{1}{x_1} + \dots + \frac{1}{x_k}}.$$

This function naturally captures many good conditions consistent with our constraints. In common with all homogeneous means, H is

- homogeneous: $H(cx_1, \dots, cx_k) = cH(x_1, \dots, x_k)$,
- and strictly monotonic:

$$(x_1, \dots, x_k) < (x'_1, \dots, x'_k) \Rightarrow H(x_1, \dots, x_k) < H(x'_1, \dots, x'_k),$$

Beyond the homogeneous means, H is also

- symmetric: $H(x_1, \dots, x_k) = H(\sigma(x_1, \dots, x_k))$,
- and has the very desirable property of favoring uniform distributions of weights.

Note: This function was initially nosed out by seeing that the behavior of capacitors in series had the desired nature of breaking weak links as well as the proper mathematical structure. Adjusting by an equalizing weight of k , since $C(1, \dots, 1) = 1/k$, led to $kC(\cdot)$, which was the harmonic mean.

We can consider H as a family of functions $H_k : \mathbb{R}_+^k \rightarrow \mathbb{R}_+$. To adjust the relative weighting of phrases of different length, we may select coefficients $\{a_k\}_{k \in \mathbb{N}}$ to produce the family $a_k H_k(x_1, \dots, x_k)$, yet still preserve our essential properties. As we want to favor longer strings a bit, initial tests showed that the increasing $a_k := \sqrt{k}$ worked well.

In passing from ratings of individual instances of phrases to global documentwise rating of phrases, we simply consider repetition. An adjusting repetition factor

$$R(\# \text{ of occurrences}) := (\# \text{ of occurrences})^2$$

was used to favor repetition, and was derived through testing.

We now show that this rating satisfies the strong phrase ranking scale invariance design constraints:

- 8i' If we adjust the weights of the words from x_i to cx_i , then since $H(\cdot)$ is homogeneous and the rest of the factors are constant per document, we see that the total adjustment of the ratings is by the same c .
- 8ii' To examine the document length extension property, for simplicity we look at the case of appending the document to itself. Ignoring the leading weight effects, we have that the word weights are doubled. Hence the rating of a phrase instance is also doubled by the homogeneity of $H(\cdot)$. But we also have twice as many instances of phrases, and as we multiply $H(\cdot)$ by $a_k(2 \cdot \# \text{ of original occurrences})^2$, we see that this factor scales by a constant of 4. Hence the total increase is by the constant 8 on all instances of phrases. For the general situation, if we increase the length of a document by a factor of L while preserving the nature of the word/HTML structure, we see the the rated phrases are uniformly scaled by L^3 .

Concerning the desired preference for uniformly strong sequences of words, this follows directly from $H(\cdot)$. Last, the phrase rating length coordination property was handled by the attached a_k .

So we see that this rating scheme nicely satisfies all our initial design constraints.

3.2.4 Selector

Currently, the top 9 phrases are selected subject to the condition that if a string is a substring of a previously emitted string, then it is suppressed. In Figure 4 we have the results from Infomine's homepage.

```

=====
Leader Of The Pack
=====
59996 electronic journals
24576 online library card catalogs
20226 electronic books
16411 government information
14693 bulletin boards
13357 scholarly internet resource collections
12857 scholarly resources
11654 social sciences
10362 instructional resources
#####

```

Figure 4: Selector's Output from the Demo Site.

Note: The general omission of substrings makes the output more concise and impressive, but as the results were to be used in a click-to-include situation, the inclusion of substrings that were higher weighted than following superstrings was considered a desirable feature.

3.3 Generalities

We have nailed the algorithm down to specifics for program design. But it is important to notice that we have at hand a natural parameterized family of solutions that is consistent with the constraints.

1. If we temporarily ignore the uniform word strength preference condition, then any homogeneous mean will satisfy the constraints. The symmetric homogeneous means form the one parameter family of elementary means $\{\mathcal{M}_r | r \in \mathbb{R}\}$, where \mathcal{M}_0 is the sporadic geometric mean [Hardy51]. We can then vary this parameter r to adjust the uniform strength condition. If we feel that the symmetric condition on the arguments is not quite accurate, we can attach convex weight factors (positive and sum to 1) and move to the generalized elementary means. By adjusting these factors, we can encourage sequences of words with a desired "shape".

Note that the conditions allow for different means to apply to different \mathbb{R}_+^k and hence different phrase lengths. Also, more general functions, such as convex combinations of homogeneous means, will also work. But there should be good cause before foraging off into the wilderness.

2. The coefficients a_k , used to provide rating synchronization between phrases of different length, can be any non-negative real number. Coefficients increasing with k seems reasonable, though our condition of selecting phrases between 2 and 5 amounts to setting $a_k := 0$ for $k > 5$.
3. With the repetition factor $R(\cdot)$, we want

$$R(c \cdot \# \text{ of occurrences}) = c' \cdot R(\# \text{ of occurrences}),$$

independent of the $\#$ of occurrences. Sanity would dictate that $R(\cdot)$ should be positive, increasing, and continuous. If $R(1) = 1$, then

$$(\forall c > 0)[R(c) = R(c1) = c' R(1) = c'],$$

and so

$$(\forall c_1, c_2 > 0)[R(c_1 c_2) = c'_1 R(c_2) = R(c_1) R(c_2)],$$

which shows R is a endomorphism on (\mathbb{R}_+, \times) . But with increasing continuous functions, we recognize this as the necessary and sufficient condition for R to be a power function of positive degree.

In general, we see that $R(1)^{-1} R(x)$ satisfies the above restricted condition, and so $R(x) = R(1)x^r, r > 0$. Thus the functions we have to select from are $\{b_k x^{r_k} | b_k, r_k > 0\}$.

Note that training at this level of abstraction would probably not only be safe, but desirable.

4 PROGRAM AND WORK IN PROGRESS

The heart of PhraseRate is essentially as specified above. In addition, it contains the standard collection of “web procedures” for handling URLs, fetching pages, dealing with framesets, and providing a web access point with diagnostic outputs. For the adders there is an appropriate GUI wrapper.

4.1 Initial Difficulties

Further developments in PhraseRate were motivated by problems encountered with challenging web sites. These fell into four general categories:

Discussion, news, and meta-sites:

Pages such as `www.cnn.com/` and `s1ashdot.org/` have their content centered around a collection of “subject capsules”. While the subject is *world news* or *computer news*, these phrases occurred relatively infrequently in the source documents. Some directories list a panorama of resources, but don’t express the general focus or objective of the site. Other sites have a “topic of the day”, and have essentially the same problem, though it is time sequential.

Heavy graphic sites:

Some pages express the majority of their content though graphics. While the probing of meta tag content and image alternate text comments provides sufficient information for those sites concerned about crawlers and text base browsers, this information is often missing within this category of sites.

Multi-subject sites:

These differ from the first case in that there really isn’t a primary focus. With these sites even professional catalogers have problems. In some sense, a failure here is not to be considered a failure since

- i. keywords may not be inappropriate,
- ii. lacking a focus, it is an unlikely candidate to catalog for academic purposes.

The “helter-skelter” site:

Some sites, while focused, are just so badly composed that it’s just best to gracefully concede defeat. Fortunately, these are rarely significant sites, at least in the realm of academic interests.

4.2 Improvements

PhraseRate has two enhancements to improve its response with the important first two cases.

Find the site’s self-description page: Some sites provide a page that describes the focus of the site. By scanning though the internal URLs and using a variety of heuristics, such as the surrounding text, directory names, and directory depth, one can often locate this page. Unfortunately, as is consistent with sites that have a weaker presentation, they either are lacking such a page or they designate it in unique ways. This tactic has met with moderate success.

Note that the related idea of internal crawling is usually not very useful. The problem here is that if the root page is too weak to determine the keyphrases, then it is unlikely to provide sufficient guidance to amalgamate the various sub-pages in a coherent manner. Further, topics tend to diverge and/or become specialized on subpages. The problem of information distributed in overlapping ways within frames also needs to be dealt with.

Time-series analysis: For sites that have dynamic content such as news or topic of the day sites, the current contents are transitory. But references to the general focus of the site are durable. As dynamic content sites are easy to identify, they can be selected for time-analysis. This method essentially amounts to building a low pass filter, using exponential averaging to filter for durable words and phrases. Currently, we have pages sampled daily over a year from a number of dynamic sites on which to run tests. Initial

tests established success in a number of challenging categories. This intriguing method is useful in off-line keyphrase extraction and promises to be an interesting area of investigation, which will also peer into periodic behavior on different time scales.

4.3 Future Improvements

Several future enhancements for PhraseRate relate to the core procedure.

- Currently, negations are dropped as stop words. But the subject description can change profoundly with leading negations. For example, a site whose focus is “No More Taxes” will currently have the suggested keyphrase “More Taxes”.
- Phrases with trailing verbs do not form keyphrase type expressions. Common verbs can be gathered and the initial phrase generation can be run through a filter to remove these cases. There is a difficulty with words that have multiple characteristics, but those words that are “strictly verbs” can be used to improve performance. The drawback is that this makes the process significantly more language centric. Currently the main language dependencies are the stop words and gluons. These are easy to modify to adapt PhraseRate to many other languages. But altering the verbs on the other hand is a significantly more difficult problem.
- Last, PhraseRate does not use any stemming. A light stemmer that handles plurals and past tense word variants would clearly be beneficial and easy to implement.

For a more ambitious approach to handling difficult pages, we are examining exterior crawling. Currently at Infomine, we are building a focused crawler. In an early stage of crawling, the first level of gathered pages are clustered by similarity and the distinguished words from each cluster are presented as a guide as to the nature and content of the individual clusters. The initial tests proved impressive, and a similar method will eventually be ported over to PhraseRate to be activated when it detects a failure to determine keyphrases by its principal methods.

5 EXPERIMENTAL RESULTS

5.1 The Test Candidates

PhraseRate (www.infoborg.org/PhraseRate/Current/) was tested against the following three programs:

Kea: <http://www.nzdl.org/Kea/>

Kea is a component of the New Zealand Digital Library, designed for automatically extracting keywords/keyphrases from text documents. Briefly, it is a naive Bayesian classifier trained on a *document to author keyword* corpus.

Turney’s Extractor: http://extractor.iit.nrc.ca/on_line_demo.html

Extractor is a commercial product which attempts to directly mimic an author’s selection of keywords by comparing its results to the author’s results. It uses a complex genetic algorithm to learn the parameters of keyphrases.

DC-dot: <http://www.ukoln.ac.uk/metadata/dcdot/>

The Dublin Core is a specification of a small set of metadata elements for describing information resources: RFC 2731. The DC-dot Dublin Core Metadata Editor is a service to provide this data from processed pages for the benefit of catalogers. The “subject or keywords” results were extracted for test comparisons.

All these programs have concerns about keywords/keyphrases, but their intended utilization and integration induced distinct specializations that were apparent in their responses, as will be noted.

5.2 Test Method

Test web pages were selected by submitting queries on *subjects* and *properties* to Google [Google] and then using the returned URLs from the response page. The reason for using properties in some of the queries was to gather a diverse collection of URLs, and in particular to include some pages that were not strongly subject oriented. This tactic turned out to be quite successful. Due to the broad spectrum of the sample, a number of the test subjects were more pathological than those typically found in the academic environment. Yet stressing the programs provided an insight into their behavior.

Figure 5: Google Search Categories

Bees	Bee Ears	Bee Witchcraft
Car Manufacturers	Linux	Mind-blowing Excitement
Pyromorphite	Quantum Computing	Red Meat
Squirm	Speakers	Turquoise
Twitch		

The test URLs were then submitted to the web demos in the case of DC-dot, Turney’s Keyphrase Extractor Demo, and PhraseRate.

In the case of Kea, as it currently does not have a web presence, the web pages were processed to text (which is the input format for Kea), and submitted as a tar file to Gordon Paynter, who applied Kea 2.0 on them. To prepare the HTML, a custom HTML to text lexer was written that would include the title, meta description, meta subject, and meta keywords from the test pages. This was necessary to provide a fair comparison for Kea, as a number of the web pages were very weak in content. Gordon Paynter [Paynter01] remarked that Kea was trained on a corpus of web pages (Aliweb, from Frank et al., 1999) and that it was restricted to phrases of 1 to 3 words.

The results returned from Kea, Extractor, and PhraseRate were listed in order of perceived weight. Gordon Paynter [Paynter01] suggested that he would take the first 7 of each set from Kea, as this was a good trade off between noise and accuracy in their experiments. So the top 7 (if available) were extracted for the Kea evaluation. Like PhraseRate, these included redundant sub-keyphrases. For Extractor’s results, it was assumed that the entries returned in the demo reflect the appropriate selection. For PhraseRate, the top 9 (if available) were selected. It was not clear that DC-dots’s results were ordered by perceived relevancy, and were included en masse.

Any web page that was not accepted by all the keyphrase extractors, which amounted to about 5 or 6 pages, was dropped from the test. The causes for these failures were due to a candidate not being able to process frames, transfer to forward references, or handle non-standard characters in the URL. Otherwise, all results were presented, including pages that generated no keywords or meaningless results from the tested programs. Weaker pages form “boundary cases” which stress the extractors. These all too common pathological web sites need to be handled, and while all the programs could use improvement, the phrase extractors were often admirably sparse in their responses.

NOTE: Both Extractor and PhraseRate can handle framesets, DC-dot evidently does not, and Kea was not given the opportunity. There were 10 framesets included in the tests and they are noted as such.

5.3 Test Results

Rating programs such as keyphrase extractors is at best difficult. One can use a panel of referees, or compare results to author selected keywords on a fixed corpus, but such methods usually feel artificial and always lack meaningful rigor. Rather than consuming time in forming such hazy judgments, it was felt sufficient to allow interested parties to review sample test results and try the demos. Among all parties we discussed the relative merits of the programs with (our adhoc panel), PhraseRate was universally viewed quite favorably, especially for its intended usage as an online keyword aid.

The full test results, which included 101 cases, are presented at:

<http://www.infoborg.org/PhraseRate/doc/phraserate.compared.html>.

Nine representative or noteworthy samples are located in the appendix, with specific case comments and contrasts between these programs. You are encouraged to peruse them at this point.

Scanning over these results shows that PhraseRate compares favorably with the state of the art Kea and Extractor, especially for interactive usage. In reviews of the results, it was generally felt that Kea performed a bit better than Extractor. Kea's results appeared to be appropriate for off-line keyword/keyphrase extraction, where there is no reviewer involved.

PhraseRate, whos intended usage was to be online, was agreed to be superior in this usage, as the keyword coverage was more comprehensive and the presentation of the phrases were more appropriate for human consumption and guidance:

- the keyphrases were short descriptions and easy to digest,
- they presented the keywords in their appropriate context,
- the collection of keyphrases generally provided a nice summery of the webpage.

Further, its speed was extremely fast, as is suitable for on-line usage: On a standard PC (1.2 Ghz) that was accessing on-campus sites, from the time it received a URL request at the command line until it finished with a diagnostic data dump, took about 0.03 seconds real time!

There are two general cases where Kea or Extractor performed better in some respect than PhraseRate:

1. The cases where Kea or Extractor contained a good keyword which PhraseRate missed were generally limited to situations in which the keyword was in the title or in the very initial segment. This occurred more often as the document length increase, suggesting that PhraseRate should adjust the initial section weighting modifiers in proportion to the total text weight.
2. PhraseRate's lack of simple stemming can lead to essentially redundant results or under valuations of some keyphrases. Adding a simple stemmer is an easy adjustment as was noted in the future improvements section.

Overall, it is evident from the test results that PhraseRate performs well as a keyphrase extractor, and that it is especially suitable for an online keyword aid. Given that its source code will be released under an open source license, it would be a very reasonable choice for any group requiring such a tool.

References

- [EN98] Endres-Niggemeyer, B., *Summarizing Information*, Springer Verlag, 1999.
- [Hardy51] Hardy, G. H., Littlewood, J. E., & Polya, G., *Inequalities, Second Edition*, Cambridge University Press, 1988.
- [Jones99] Jones, S. and Staveley, M. (1999). "Phrasier: a system for interactive document retrieval using keyphrases", SIGIR '99 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval. University of California, Berkeley August 15-19, 1999, pp 160-167.
- [Mitchell97] Mitchell, T. M. *Machine Learning*, McGraw Hill, 1997.
- [Musciano00] Musciano, C. and Kennedy, B., *HTML & XHTML: The Definitive Guide, 4th ed.*, O'Reilly & Associates, 2000.
- [Paynter00] Paynter, G. W., Cunningham, S. J. and Witten, I. H. (2000). "Evaluating extracted phrases and extending thesauri", Proc Asian Digital Libraries Conference, Seoul, Korea, pp. 131-138.
- [Paynter01] Paynter, Gordon W., (2001). *Personal Communication*,
- [Salton96] Gerard Salton, Amit Singhal, Chris Buckley, and Mandar Mitra. (1996). "Automatic Text Decomposition Using Text Segments and Text Themes", In Proceedings of the Hypertext '96 Conference, Washington D.C., USA,

[Turney97] P. Turney. (1997). *“Extraction of keyphrases from text: evaluation of four algorithms”*, Technical Report NRC 41550, National Research Council of Canada, 1997.

[Witten99] 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.

DC-dot: <http://www.ukoln.ac.uk/metadata/dcdot/>

Infomine: <http://Infomine.ucr.edu/>

Google: <http://www.google.com/>

Kea: <http://www.nzdl.org/Kea/>

PhraseRate Demo: <http://www.infoborg.org/PhraseRate/Current/>

PhraseRate Test Results:

<http://www.infoborg.org/PhraseRate/doc/phraserate.compared.html>

PhraseRate Test Keyword Covering Results:

<http://www.infoborg.org/PhraseRate/doc/phraserate.compared.keyword.covering.html>

Turney’s Keyphrase Extractor: <http://extractor.iit.nrc.ca/>

A Appendix

The examples contain the following information:

- URL and page title,
- the number of words and characters in each page, both with and without HTML (and excess space),
- whether the page was a frameset or not (#2 in this collection),
- an explanation as to why the example is noteworthy,
- results from the different sites, with Dublin Core results often truncated for brevity. The token leading a keyphrase in the Kea and Extractor section is assigned by:
 - + : This phrase occurs as a subphrase of PhraseRate,
 - w : else this phrase is PhraseRate’s “word of the day”,
 - t : else this phrase occurs in the title,
 - : a complete miss.

URL: http://www.pbs.org/pov/films/twitch.html #1			
Title: Twitch and Shout			
Words: 213/161 Chars: 1943/1030			
A typical example of a good page for the various programs.			
KEA	EXTRACTOR	PHRASERATE	DC-DOT
+ Twitch and Shout + Tourette Syndrome - disorder + resources	+ Shout + Twitch + Related Resources - incurable genetic disorder + Tourette Syndrome - minds - bodies	1. twitch and shout 2. tourette syndrome 3. film by laurel chiten 4. impact television 5. information transcript related resources tape 6. interactive partners 7. press information transcript related resources 8. people with tourette syndrome	○ High Impact Television ○ Laurel Chiten ○ Press Information ○ A film by ○ project ○ Twitch and Shout ○ Tape Orders ○ Transcript ○ (4 more entries)

URL: http://www.dancingbeeacres.com/beevenom.html #2			
Title: Dancing Bee Acres: Raw Honey, Pollen, Royal Jelly And ...			
Words: 3528/3017 Chars: 27140/17963 FRAMESET			
This example shows a defect that occurs in all the programs: that of listing unnecessary substrings. This can be solved simply by removing them after the selection phase. Again, the longer phrases of PhraseRate are quite helpful in determining the content of the page. For example, in Kea, after “honey” follows “Jelly”, which actually is “Royal Jelly”. “Reversing multiple sclerosis” is more informative than “MS”.			
KEA	EXTRACTOR	PHRASERATE	DC-DOT
+ Bee + therapy + Venom + Bee Venom - MS t honey + Jelly	+ bee venom + bee venom therapy + Pepe + multiple sclerosis - eye - reaction - treatment	1. bee venom 2. bee venom therapy 3. multiple sclerosis 4. dr. pepe 5. royal jelly 6. bee venom drops 7. reversing multiple sclerosis 8. bee venom therapy supplies 9. nervous system	○ honey ○ bee ○ bees ○ pollen ○ royal ○ jelly ○ propolis ○ venom ○ (83 more entries)

URL: http://www.nature.ca/notebooks/english/bees.htm #3			
Title: bees			
Words: 259/183 Chars: 2357/1098			
This example shows an advantage of phrases. Kea accepts “Bees” and “Honey” as independent keywords, but the document only references “honey bees” and the topic of honey itself does not appear. While the listing shows “insects” is missing in PhraseRate, note that “insect” does appear there. So the coverage in PhraseRate is better than the “numbers” indicate.			
KEA	EXTRACTOR	PHRASERATE	DC-DOT
+ Bees - insects + Honey + eyes + ultraviolet light - order + capable	+ bees + insect + eyes - insect order + ultraviolet light + human eyes + honey bees	1. ultraviolet light 2. honey bees 3. capable of seeing ultraviolet light 4. bees belong 5. cloud-penetrating ultraviolet light 6. earth's insect population in check 7. greatly from human eyes 8. pair of compound eyes 9. bee's eyes	o Apidae o bees

URL: http://www.redmeat.com/redmeat/ #4			
Title: Red Meat - from the secret files of Max Cannon			
Words: 195/59 Chars: 2683/372			
This shows the benefits of using the meta data. PhraseRate processes meta data and Kea was also provided with a text file including meta data. Extractor is lost on this page, which actually has plenty of information.			
KEA	EXTRACTOR	PHRASERATE	DC-DOT
+ meat + red meat + Max Cannon + files of Max + secret files + strip + comic	o ERROR: Very little text is at this URL. This URL might be mainly graphics, rather than text. Extractor can not extract keyphrases from an image.	1. max cannon 2. secret files of max cannon 3. red meat 4. twisted comic strip 5. tasteless and twisted comic strip 6. lost world 7. milkman dan 8. migraine boy	o red o meat o red meat o Max o Cannon o Max Cannon o Earl o Milkman Dan o (19 more entries)

URL: http://www.bpassion.com/queensquarters.html #7			
Title: Blessed Bee! Queen's Quarters			
Words: 1088/806 Chars: 8145/4868			
Frequently used words occurring in conjunction captured aspects of this site.			
KEA	EXTRACTOR	PHRASERATE	DC-DOT
+ Quarters + Queen's Quarters + Witch - seek - believe + Wiccan + Wiccan belief	+ life + power - religion - lives - practice - traditions + American Witches	1. american witches 2. queen's quarters 3. wiccan belief 4. blessed bee 5. creative power 6. harmony with nature 7. feminine masculine 8. philosophy of life 9. principals of wiccan belief	o Wax Facts o Goddesses o Candle Care o Company Info o Slaves' Gallery o PRINCIPALS OF WICCAN BELIEF o Floor Planz o HOME o (9 more entries)

URL: http://www.autoalliance.org/ #5			
Title: The Alliance of Automobile Manufacturers			
Words: 723/69 Chars: 10368/524			
<p>Another example of the importance of examining the meta tags. All the text on this site resides there. When Kea had a web demo, they didn't examine this content. They were a bit impressed with their results when given this enriched text.</p> <p>Note the fracturing of meaningful keyphrases in Kea which produces a poor delineation of the site. It might be argued that keywords are not appropriate for programs to gather.</p>			
KEA	EXTRACTOR	PHRASERATE	DC-DOT
<ul style="list-style-type: none"> + Alliance of Automobile + Automobile Manufacturers + policy + motor + motor vehicle safety + fuel + environment 	<ul style="list-style-type: none"> o No keyphrases were extracted. 	<ol style="list-style-type: none"> 1. alliance of automobile manufacturers 2. motor vehicle safety 3. public policy 4. environment and motor vehicle safety 5. broad range of public policy issues 6. fuel economy 7. ford motor company 	<ul style="list-style-type: none"> o Alliance of Automobile Manufacturers o BMW o DaimlerChrysler o Fiat o Ford Motor Company o General Motors o Isuzu o Mazda o (14 more entries)

URL: http://theband.hiof.no/articles/rs72_rock_of_ages.html #6			
Title: Ralph J. Gleason: "Rock of Ages": A crackling, mind-blowing ...			
Words: 3025/2739 Chars: 21114/15627			
<p>This example shows how the relation of keywords to each other add significant information. This site is about The Band. The "rock" listed is related to the spiritual "rock of ages." Also, the only mention of concert is "concert album." A difficulty in this page is that the "the" in "the band" is a stop word.</p> <p>We are seeing more typical behavior out of Dublin Core here, which tends to grasp at any excuse to list a "keyphrase", though it showed more restraint here than on most longer pages.</p>			
KEA	EXTRACTOR	PHRASERATE	DC-DOT
<ul style="list-style-type: none"> - horns + concert - Band + album - recorded + music + Rock 	<ul style="list-style-type: none"> + concert - horns + album + Robbie - band + Garth + rock 	<ol style="list-style-type: none"> 1. rock of ages 2. mind-blowing moment preserved 3. concert album 4. crackling mind-blowing moment preserved 5. academy of music 6. garth hudson 7. robbie and rick 8. howard johnson 9. ray charles 	<ul style="list-style-type: none"> o Concerts o Filmography o Videography o Rock of Ages o Audio Files o Related Artists o wild Cahoots o Video Clips o (21 more entries)

URL: http://www.redmeatclub.com/			#8
Title: Red Meat Club - Online recommendations for gourmet food, ...			
Words: 1627/992 Chars: 19080/6419			
A somewhat difficult site to nail down. It's an online food store, and the page has a number of special food topics. While easy for humans to figure out, the programs find it a challenge. Note PhraseRate's summarization capabilities still show though. The excessive report from Dublin-Core is actually typical.			
KEA	EXTRACTOR	PHRASERATE	DC-DOT
<ul style="list-style-type: none"> + Gift - cocktail + Club + Meat + Red Meat Club + food + wine 	<ul style="list-style-type: none"> + gift - cocktail - York - restaurant + club - pasta - fresh 	<ol style="list-style-type: none"> 1. red meat club 2. egg cream 3. wine and luxury gifts 4. egg cream kit 5. shipped overnight 6. gift box 7. junior's of brooklyn 8. martini shop 9. gourmet food 	<ul style="list-style-type: none"> o Reserved o delicious o bran o Crunch o Sauces o authentic o York o CLICK o (198 more entries)

URL: http://www.lifesatwitch.com/			#9
Title: Life's A Twitch -- Splash Page!			
Words: 204/97 Chars: 2664/593			
A site about Tourette Syndrome, though its home page focuses on browser adjustment. The syndrome is "mentioned" once in a graphic located at the bottom. This example is offered as evidence of a page that is easily identified by a user but is impossible for a program.			
KEA	EXTRACTOR	PHRASERATE	DC-DOT
<ul style="list-style-type: none"> - Screen area + BROWSERS + font + Size 	<ul style="list-style-type: none"> + BROWSERS + BROWSER font - area adjustment - Settings tab - SCREEN area MODE - higher xxxxxx-AOL Software - FRIEND 	<ol style="list-style-type: none"> 1. splash page 2. font size 3. view/text size 4. x600 screen 5. browser font size 6. images overlap 7. text and images overlap 8. following browsers 9. aol software 	<ul style="list-style-type: none"> o go to View/Text Size and o Screen area adjustment is on the bottom right o choose a smaller font o On the Windows task bar, select "Start", "Settings", "Control Panel", and "Display" o Click on the Settings tab o (12 more entries)