A Comparative Analysis of Keyword Extraction Techniques

Michael J. Giarlo

Rutgers, The State University of New Jersey

Introduction

With widespread digitization of printed materials and steady growth of "born-digital" resources, there arise certain questions about access and discoverability. One such question is whether the full-text of this content, produced by advanced optical character recognition (OCR) techniques, is sufficient as a descriptor of the content. Will the model of mass digitization and full-text searching enable users to find the information they need? Or will we need to continue employing the classification skills of highly qualified human beings in order to ensure information is discoverable? The latter model seems to have worked well for the library community, with trained indexers and catalogers summarizing documents according to established standards and widely used thesauri or controlled vocabularies. The predictability of these techniques has some obvious benefits, such as consistency across different systems, the ability to construct browse interfaces in addition to search ones, and reduction of common errors such as differences in case, punctuation, spelling, and so forth. The process of human classification has thus proven to be quite effective in our endeavors to organize information.

The question of whether we will continue to classify digital content in a similar manner ought to be asked. Is there any hope to keep up with the dizzying pace with which documents are digitized? Classification is a costly, time-consuming process, requiring highly trained individuals to consume a large amount of information and summarize it. If the goal is to continue digitizing and making accessible information at the current rate, it is improbable that human catalogers and indexers will be able to keep up without sacrificing some of the quality that results from their considerable skills. Yet the goal of enhancing access and discoverability of digital content is one that ought to be pursued, and will likely not be realized through full-text searching alone. Indeed, why should we put so much time and effort into the process of digitization if it does not benefit our users?

Fortunately, the process of automatic extraction of keywords is one that has received much

attention. As implied by the phrase, automatic keyword extraction is a process by which representative terms are systematically extracted from a text with either minimal or no human intervention, depending on the model. The goal of automatic extraction is to apply the power and speed of computation to the problems of access and discoverability, adding value to information organization and retrieval without the significant costs and drawbacks associated with human indexers. Research is taking place in numerous fields across the globe, and there is no clear frontrunner among the technologies and algorithms. This paper explores five approaches to keyword extraction, as presented in research papers, to demonstrate the different ways keywords may be extracted, to reflect commonalities between the approaches, and to evaluate the results thereof. Each paper is presented in a different section, for ease of organization.

Applications of Keyword Extraction

Domain-Based Extraction of Technical Keyphrases

Frank *et al.* (1999) argue that the quality of automatic keyword extraction, or "keyphrase extraction" in their parlance, could be greatly improved by using machine learning techniques wherein domain-specific models are created from sets of training documents, thus tailoring the keyword judgments the system makes to the collection or set of documents from which is it extracting. There are numerous machine learning techniques that are available, though more complex algorithms require more computing power and more time to process. It is proposed that a simpler algorithm, such as the naive Bayes formula, would perform better without sacrificing quality. The system that was built by the researchers is hereafter known as Kea, or Keyphrase Extraction Algorithm.

The machine learning mechanism works as follows. First a set of training documents are provided to the system, each of which has a set of human-chosen keywords as well. Kea uses the standard TF*IDF measure -- where the frequency of a term in a document is multiplied by the inverse

of the frequency of the a term the collection -- in addition to the relative position of a term's first occurrence in a document, and must then decide based on these measures whether a term is a keyword or not. Since human-chosen keywords are provided with the training set, Kea has a "cheat sheet" of sorts to see which mappings work and which do not. The more documents are included in a training set, theoretically, the more precise the model can become.

Kea will choose up to a specified number of keywords for each document, an attribute that must be set and is a hard limit. The algorithm also limits keywords to three-token phrases (or trigrams), eliminates those that are stopword-initial or -final, filters out single-token (unigram) proper noun entries, and applies the Lovins stemming technique to arrive at its keywords.

Five experiments in total were conducted to test the performance and quality of Kea compared with more computationally expensive machine learning algorithms. The first two experiments serve to compare it with another, though more process-intensive, algorithm. The training dataset for the first experiment consisted of 55 journal articles from different domains, such as neuroscience, computer-aided design, and behavioral science. The dataset used for testing the model generated by the training consists of 20 articles from a journal for psychology and cognitive science. The second experiment has the same training dataset but had a different test set made up of *FIPS* web pages. Compared to another algorithm which can take nearly one-thousand times longer for learning and model-building, Kea performs on par both at the 5-term and 15-term extraction levels. That is, the difference between the number of correct terms chosen by the other algorithm and those chosen by Kea are not statistically significant, and Kea even outperforms the other algorithm at the 15-term level.

The third experiment was designed to gauge how changing the size of the training set affected the number of correct terms selected in the test set. For this experiment, training sets of 50 and 500 documents from the New Zealand Digital Library were used, specifically computer science technical reports. It was determined that no performance increases were evident for document sets with greater than fifty documents and, indeed, there was minimal effect on performance with document sets

sized greater than twenty. The researchers attribute this to the fact that the training set drew documents from multiple disciplines within a certain field, computer science, and propose that constructing models from domain-specific training collections will yield different results.

The fourth experiment uses a number of different domains, running nine separate tests with variant training sets and test sets, the goal being to see how Kea responded to being trained on one domain and tested on a separate domain. Though this experiment did show minimal gains in the number of correct keywords chosen from training sets that were related to test sets, the gains were not statistically significant. The researchers thus concluded that a major performance increase would result if the notion that keywords vary in terms of relevance in different domains were added to the machine learning algorithm.

Given the results of the fourth experiment, the naive Bayes-based machine learning algorithm was modified to include a measure of keyword frequency, that is, the number of a times one of the human-selected keywords appears in the training set. The theory is that adding this feature to the machine learning would generate stronger keywords from models that learn from specific domains. To test this hypothesis, a fifth experiment was conducted consisting of 130 training documents from a single domain, computer science, and 500 test documents. As hypothesized, the number of correct keywords chosen remains impressive -- though exact precision and recall measures are not included -- and, perhaps more importantly, the size of the training set is found to be statistically significant to the number of correct keywords chosen. It is shown that increasing the training set from 50 to 100, and from 100 to 1000 results in dramatic improvements.

Spoken Language Processing with Term Weighting

Suzuki *et al.* (1998) seeks to use spoken language processing techniques to extract keywords from radio news, using an encyclopedia and newspaper articles as a guide for relevance. The methodology proposed is separated into two phases: term-weighting and keyword extraction. First, an

encyclopedia containing 141 subject domains is used to generate an initial set of feature vectors, after nouns are extracted by the JUMAN morpheme-analysis system, and frequencies are computed. A similar process of common noun extraction, frequency counting, and feature vector calculation is then performed on a corpus of approximately 110,000 newspaper articles. The encyclopedia vectors are compared with the article vectors using a similarity calculation so as to separate the latter into different domains, after which they are sorted, producing the final set of feature vectors.

In the second phrase, keyword extraction, a segment is analyzed such that the most relevant domain is selected for it using the pre-existing feature vectors. Phoneme recognition software is employed to do the analysis, looking for the best fit between a segment's vectors and that of one of the encyclopedia domains. When the best fitting domain is chosen, its keywords are then assigned to the radio news segment.

Two experiments were conducted using the two-phase methodology. In the initial experiment, 643 radio news segments were run through the term-weighting and keyword extraction phase, and then tested for the standard measures of precision and recall. Recall was found to be 58.7%, and precision was calculated to be 74.0%. It should be noted that these terms are not used in the typical sense, measuring percentage of collection. Recall is defined as the number of keywords in what the system choose as the "most suitable keyword path" (MSKP) divided by the number of selected words in MSKP. Precision is similarly defined as the number of keywords in MSKP divided by the number of keywords in the segment. The second experiment differed in that phoneme recognition was used on 50 segments. Both recall and precision were lower than the first experiment, measured respectively at 34.1% and 42.5%, demonstrating the uneven quality of the phoneme recognition employed by the researchers.

Spoken Text Keyword Extraction with Lexical Resources

Not all keyword extraction is based on statistical methods and encyclopedias. Some

keyword extraction is performed using linguistically-informed tools and resources. Plas *et al.* (2004) set out to evaluate two lexical resources: the EDR electronic dictionary, and Princeton University's freely available WordNet. Both provide well-populated lexicons including semantic relationships and linking, such as IS-A and PART-OF relations and concept polysemy. The resources are compared by using them for the same task of automatic keyword extraction from multiple-party dialogue episodes. It is argued by the authors that using lexical resources will result in better quality keywords than purely statistical methods, such as measures involving term-weighting and TF*IDF. To that end, the lexical resources are compared to a statistical method, relative frequency ratio (RFR), in addition to each other.

Keyword extraction is limited to nouns, due to wider coverage in the lexical resources, and also because it is argued that they are the most commonly found part of speech in keywords. Each segment of spoken text from the multiple-party dialogues is first tagged with TreeTagger, a probabilistic part of speech tagger. The nouns are then selected as potential keywords, and relative frequency ratio (RFR) is calculated, which is a basic TF*IDF measure. The nouns are then related with the most common sense of the concept from each lexical resource, and are checked for similarity using the Leacock-Chorodow measure, based on the length of the paths between the concepts, such as in the IS-A relationship hierarchy, e.g., "tiger" IS-A "cat" IS-A "animal", etc. After the semantic similarity measure is calculated, keyword candidates are chosen using single-link clustering. These clusters are assigned a cluster-level score and a concept-level score, and then ranked. The ranking was informed by manually selected keywords, used as a very basic type of correction or learning. The number of keywords was limited to 10, and the system chose 6.2 keywords for each dialogue segment on average.

The dataset used for the experiment is a collection of transcriptions of International Computer Science Institute's meeting dialogues. Each transcription contains an average of six parties conversing about a small set of narrow topics, such as speech recognition and audio equipment. Twenty-five of these transcriptions were already separated into coherent segments, and the clearest six were selected. The six transcriptions contained an average of nine segments each.

The results of the experiment were evaluated based on two measures: average k-accuracy, or correct keywords chosen, and a combined measure of precision, recall, and F-score. These measures were computed at three levels, when the number of keywords was set to 2, 5, and 10, and each set of measures was computed for each lexical resource and for the standard RFR technique. Overall, the WordNet resource was found to perform the best, especially when the semantic similarity between the concepts was judged at the highest level. At lower semantic similarity levels, the EDR resource performed better than WordNet. Across the various semantic similarities and numbers of keywords, WordNet typically showed a higher precision measure, whereas the recall of EDR was higher. The authors suggest that the EDR was less susceptible to some of this variability due to the difference in depth between the two resources. Finally, it is shown that both lexical resources clearly outperform the basic statistical method, RFR, or TF*IDF.

Keyword Extraction with Thesauri and Content Analysis

Deegan *et al.* (2004) explore keyword extraction from a large number of documents on forced migration, through the building and usage of two separate resources: a thesaurus, and a number of newspapers and web-based news pages. During the course of the research, a thesaurus of refugee terminology (ITRT) was actually built specifically for the purposes of the extraction project. Two experiments were conducted by independent teams, one using the thesaurus and the other using web news and newspapers for terms, in order to contrast the two. The first team used the UCREL Semantic Annotation System (USAS), a dictionary-based content analysis tool, to tag its collection of texts. The second team, who generated data from web news and newspapers, analyzed keywords primarily using the Wordsmith Tools software package.

The dataset for both experiments was an unspecified subset of the digital library, Forced Migration Online, containing approximately 80,000 documents. The corpus of the second experiment is not mention, but the first experiment had a document set with 432,317 words.

The first team first had to examine the semantic domains analyzed by USAS and compare them to the ITRT thesaurus, and figured out mappings between them. The USAS system works by part of speech-tagging every word and phrase using probabilistic Markov models. Then it uses SEMTAG to apply semantic tags based on matching between the text and the provided dictionaries/thesauri, and finally runs a disambiguation process to judge the best sense given the context the word or phrase is in. The second team conducted its tests using the WordSmith Tools package, which is not presented in great detail.

The USAS tagger was found to work with 97% accuracy, an impressive result, and the SEMTAG worked with 92% accuracy. The categories suggested by USAS were found to map quite easily to the ITRT high-level categories, so the authors conclude that the results were quite promising. Cross-domain extraction was handled effectively as well, with the system seeming not to show preferences for keywords in any particular domain. While the thesaurus was concluded to aid in the extraction of keywords quite notably, it was also found that the keyword extraction generated a number of keywords that were missed in the original resource. So we see something of a symbiotic relationship obtaining between keyword extraction and thesauri in this case. In the second experiment, which was judged to be successful like the first, the difference in the nature of the resources was reflected, i.e., the difference between the thesaurus used by the first team and the newspaper articles used by the second team. Namely, the keywords chosen in the latter scheme seemed to reflect "highly emotive, persuasive and manipulative terminology" of the media, unlike the more neutral and objective terms used by the intergovernmental agency that produced the thesaurus. The authors conclude that the automatic keyword extraction techniques worked almost as well as human indexers could have.

Linguistic Features as Error Correction in Keyword Extraction

Hulth (2003) proposes that linguistic properties of texts will yield higher quality keywords

and better retrieval, and examines a few different methods of incorporating linguistics into keyword extraction. Three methods of extraction are evaluated: *n*-grams, NP chunks, and part-of-speech pattern matches. Terms are vetted as keywords based on three features: document frequency (TF), collection frequency (IDF), and relative position of its first occurrence in a document. An additional fourth feature is evaluated independently of the other three features, namely the term's (or phrase's) part of speech tag. A supervised machine learning algorithm is used, very similar to Kea, whereby a classifier is trained by using a set of training documents with known keywords. Contrary to Kea, the author argues against setting arbitrary limits on the number of terms that should be allowed in a keyword (which is often limited to three-word phrases), and also against imposing a limit on the number of keywords chosen for a document. Some documents are denser than others, and some are simply "about" less than others. A system that forces each document to have a certain number of keywords will likely strip away the usefulness of keywords for documents with many possible keywords, and will possibly add "junk" keywords to those documents that may be summarized in a word or two. The author believes that these determinations are better left to the system, assuming of course that it is "smart" enough to handle such decisions. Analyzing NP chunks and POS tags also gets around the problem of arbitrary term length since it permits the system to let actual linguistic properties of the text to determine the results of indexing. The usage of machine learning algorithms can skirt the problem of choosing an arbitrary number of keywords per document, since it will more or less intelligently choose keywords based on the training set and properties of the document, rather than an externally imposed limit.

The dataset for the experiment conducted consists of 2,000 English abstracts (hereafter, "documents"), with titles and manual keywords included, from the *Inspec* databases, within fields related to computer science and information technology. Each document was provided with two sets of terms, one set of terms controlled by the database and one set of free, uncontrolled terms chosen by the human indexer. The experiment was concerned only with the uncontrolled terms since they more

frequently appeared in the document -- 76.2% of the uncontrolled terms were present in the documents, compared with 18.1% of the controlled terms. The set of 2,000 documents was divided randomly into three sets: a training set of 1,000 documents, a validation set of 500 documents, and a test set of 500 documents.

Though the name of the machine learning algorithm used is not provided, other details of the experiment were given. The *n*-gram extraction method was performed using a list of stopwords, after which keywords were stemmed using the Porter algorithm. The NP-chunking was done with LT CHUNK. And the POS-tagging was done with LT POS. Both LT CHUNK and LT POS are freely available. The POS-tagger used 56 patterns for extraction, the most common of which were Adjective Noun (singular/mass), Noun Noun (both singular/mass), Adjective Noun (plural), Noun (singular/mass) Noun (plural), and Noun (singular/mass). As mentioned previously, the features calculated for each keyword are document frequency (TF), collection frequency (IDF), relative position of first occurrence, and part-of-speech tag. It should be noted that the TF and IDF measures were used independently rather than as the composite measure TF*IDF.

The results of the validation set were evaluated using the F-score, combining the measures of precision and recall, both of which were judged to be pertinent to this experiment. The purpose of the validation set was to determine the best-performing models or classifiers generated by the machine learning algorithm with the training set. The highest-performing models are then selected and used on the test set of 500 documents which again are computer science and information technology abstracts.

Across the board, experiments with stemming enabled yielded better precision and recall results than those without stemming enabled. The *n*-gram extraction method chose 4.37 correct keywords per document, where "correct" means that they align with manually selected keywords, of which there were 7.63 per document. Along with the 4.37 correct keywords, a staggering 38 incorrect keywords were also generated. With the fourth feature, POS-tag, factored into the *n*-gram experiment, the number of correct keywords drops a little and the number of incorrects drops by a third. *n*-gram

extraction with the POS-tag feature generated the highest F-score of all measures. NP-chunking chose 16.38 keywords per document with the POS-tag feature turned off, and 9.58 keywords with POS-tagging enabled. Though the number of these that were correct is not included in the report, it is stated that more than half of the keywords were lost, and the number of incorrect keywords "decreased considerably." With the POS-tag feature enabled, the number of incorrect terms was halved, with little impact on the correct terms. It should be noted that the highest precision in the experiment was seen with the NP-chunking method with POS-tag feature enabled. The POS-tag pattern matching test resulted in 5.04 keywords per document without the POS-tag feature, and 3.05 with the feature. Without the POS-tag feature, the pattern matching approach achieved the highest level of recall.

The results indicate that each method of keyword extraction -- *n*-gram, NP-chunks, and POS patterns -- has benefits and drawbacks, as demonstrated by the highest values for F-score, precision, and recall each belonging to different methods. Perhaps the most interesting data to come out of this experiment is that the POS-tag feature, across the board, serves to weed out the number of incorrect terms without having a statistically significant effect on the number of correct terms. The linguistic properties of texts are shown to be quite powerful in the automatic extraction of keywords.

Conclusion

Though the selection of articles analyzed in this paper is limited, one can already see the different ways in which keyword extraction is being used, and also some of the commonalities among applications thereof. It is first worth noting that the fundamentals of keyword extraction are being applied both to spoken language processing (SLP) and natural language processing (NLP) or text processing, which are very different technologies. Though the technologies for processing audio signals and parsing textual data are so dissimilar, keyword extraction is found to be an effective

alternative to human-produced index terms.

There are also differences in the type of keyword extraction that is chosen, which may be broken into three categories: statistical methods, linguistic methods, and mixed methods. Statistical methods, such as those employed in Kea, tend to focus on non-linguistic features of the text such as term frequency, inverse document frequency, and position of a keyword. The benefits of purely statistical methods are their ease of use, limited computation requirements, and the fact that they do generally produce good results. However, as is shown in a number of the articles herein, methods which pay attention to linguistic features such as part-of-speech, syntactic structure (e.g., NP chunks), and semantic qualities tend to add value, functioning sometimes as filters for bad keywords. Some of the linguistic methods are in fact mixed methods, incorporating some linguistic methods with common statistical measures such as term frequency and inverse document frequency. In fact, one of the common features of all the articles reviewed is the use, to some extent, of both TF and IDF as document features.

Keyword extraction techniques seem to be maturing rapidly, with new techniques arising concurrently. This is well-timed given a number of high-profile, mass-digitization projects. If they are any indication that this is the Digitization Age, the benefits offered by automatic keyword extraction would best be investigated by all who are now engaged in digitization and wish to provide value-added search and discovery to their content.

References

Deegan, M., Short, H., Archer, D., Baker, P., McEnery, T., & Rayson, P. (2004). Computational linguistics meets metadata, or the automatic extraction of key words from full text content. *RLG DigiNews, 8*(2). Retrieved from http://www.rlg.org/en/page.php?Page_ID=17068.

Frank, E., Paynter, G.W., Witten, I.H., Gutwin, C., & Nevill-Manning, C.G. (1999). Domain-specific keyphrase extraction. *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, *1999,* 668-673.

Hulth, A. (2003). Improved automatic keyword extraction given more linguistic knowledge. *Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2003*, 216-223.

Plas, L. van der, Pallotta, V., Rajman, M., & Ghorbel, H. (2004). Automatic keyword extraction from spoken text. A comparison of two lexical resources: the EDR and WordNet. In Lino, M.T., Xavier, M.F., Ferreira, F., Costa, R., Silva, R. (eds.), *Proceedings of the 4th International Conference on Language Resources and Evaluation, European Language Resource Association*, *2004*, 2205-2208.

Suzuki, Y., Fukumoto, F., & Sekiguchi, Y. (1998). Keyword extraction of radio news using term weighting with an encyclopedia and newspaper articles. *SIGIR, 1998,* 373-374.