KEA: Practical Automatic Keyphrase Extraction

Ian H. Witten,[*] Gordon W. Paynter,[*] Eibe Frank,[*]
Carl Gutwin[†] and Craig G. Nevill-Manning[‡]

[*] Dept of Computer Science,
University of Waikato,
Hamilton, New Zealand.
{ihw,gwp,eibe}@cs.waikato.ac.nz

[†] Dept of Computer Science,
University of Saskatchewan,
Saskatoon, Canada
gutwin@cs.usask.ca

[‡] Dept of Computer Science,
Rutgers University,
Piscataway, New Jersey
nevill@cs.rutgers.edu

Keyphrases provide semantic metadata that summarize and characterize documents. Kea is an algorithm for automatically extracting keyphrases from text. We use a large test corpus to evaluate its effectiveness in terms of how many author-assigned keyphrases are correctly identified. The system is simple, robust, and publicly available. Kea identifies candidate keyphrases using lexical methods, calculates feature values for each candidate, and uses a machine-learning algorithm to predict which candidates are good keyphrases. The machine learning scheme first builds a prediction model using training documents with known keyphrases, and then uses the model to find keyphrases in new documents.

Keyphrases are useful because they briefly summarize a document's content. As large document collections such as digital libraries become widespread, the value of such summary information increases. Keywords and keyphrases are particularly useful because they can be interpreted individually and independently of each other. They can be used in information retrieval systems as descriptions of the documents returned by a query, as the basis for search indexes, as a way of browsing a collection, and as a document clustering technique (e.g. [2], [3], [4]).

Keyphrases are usually chosen manually. In many academic contexts, authors assign keyphrases to documents they have written. Professional indexers often choose phrases from a "controlled vocabulary" that is predefined for the domain at hand. However, the great majority of documents come without keyphrases, and assigning them manually is a tedious process that requires knowledge of the subject matter. Automatic extraction techniques are potentially of great benefit.

THE KEA ALGORITHM

Kea is an algorithm for automatically extracting keyphrases from text. The algorithm has two stages:

1. Training: create a model for identifying keyphrases, using training documents where the author's keyphrases are known.
2. Extraction: choose keyphrases from a new document, using the above model.

Both stages choose a set of candidate phrases from their input documents, and then calculate the values of certain attributes, or features, for each candidate.

Candidate phrases. Kea chooses candidate phrases in three steps. It first cleans the input text, then identifies candidates, and finally stems and case-folds the phrases. After splitting the text into words and sentences, Kea considers all the subsequences in each sentence and determines which of these are suitable candidate phrases. All words are then case-folded and stemmed.

Feature Calculation. Two features are calculated for each candidate phrase and used in training and extraction. They are *TF×IDF*, a measure of a phrase's frequency in a document compared to its rarity in general use; and *first occurrence*, which is the distance into the document of the phrase's first appearance.

Training. The training stage uses a set of training documents for which the author's keyphrases are known. For each training document, candidate phrases are identified and their feature values are calculated as described above. The scheme then generates a model that predicts the class using the values of the other two features.

We have experimented with a number of different machine learning schemes; Kea uses the Naïve Bayes technique because it is simple and yields good results [1]. This scheme learns two sets of numeric weights from the discretized feature values, one set applying to positive ("is a keyphrase") examples and the other to negative ("is not a keyphrase") instances.

Extracting keyphrases from new documents. To select keyphrases from a new document, Kea extracts candidate phrases, determines feature values, and then applies the model built during training. The model determines the overall probability that each candidate is a keyphrase, and then a post-processing operation selects the best set of keyphrases.

DL 99, Berkeley, CA USA
Copyright ACM 1999 1-58113-145-3/99/08 . . . $5.00

Protocols for secure, atomic transaction execution in electronic commerce		Neural multigrid for gauge theories and other disordered systems		Proof nets, garbage, and computations	
anonymity	*atomicity*	disordered systems	disordered	*cut-elimination*	cut
atomicity	*auction*	*gauge fields*	gauge	linear logic	*cut elimination*
auction	customer	*multigrid*	*gauge fields*	*proof nets*	garbage
electronic commerce	*electronic commerce*	neural multigrid	interpolation kernels	sharing graphs	*proof net*
privacy	intruder	neural networks	length scale	typed lambda-calculus	weakening
real-time	merchant		*multigrid*		
security	protocol		smooth		
transaction	*security*				
	third party				
	transaction				

Figure 1 Examples of author- and Kea-assigned keyphrases

EVALUATION

We carried out an empirical evaluation of Kea using documents from the New Zealand Digital Library [5]. Our goals were to assess Kea's overall effectiveness, and also to investigate the effects of varying several parameters in the extraction process. We measured keyphrase quality by counting the number of matches between Kea's output and the keyphrases that were originally chosen by the document's author. Figure 1 lists the Kea- and author-assigned keyphrases for three computer science technical reports. Phrases that appear in both lists are italicized.

Our results show that Kea can on average match between one and two of the five keyphrases chosen by the author in this collection [1]. We consider this to be good performance. Although Kea find less than half the author's phrases, it must choose from many thousands of candidates; also, it is highly unlikely that even another human would select the same set of phrases as the original author.

Furthermore, we have determined that the following are reasonable minimums on source data for using Kea effectively:

- Kea works well with a training set of as few as 20 documents, meaning that human indexers need only assign manual keyphrases to a small number of documents in order to extract good keyphrases from the rest of the collection.
- Kea works best on the full text of documents, rather than just titles and abstracts
- The global document corpus (used to calculate TFxIDF scores) can contain as few as 10 documents, and does not need to contain documents that are similar to the collection being processed.

CONCLUSION

Kea is an algorithm for automatically extracting key phrases from text. Our goal is to provide useful metadata where none existed before. By extracting reasonable summaries from text documents, we give a valuable tool to designers and users of digital libraries.

In future, we plan to expand the evaluation of the algorithm. In particular, we have been working with the assumption that using author-specified keyphrases to evaluate the scheme is a reasonable indicator of finding 'good' keyphrases. However, in the near future we will test that assumption by evaluating Kea's output using human expert judges, and by comparing Kea to other document summarization methods.

Kea is available from the New Zealand Digital Library project (http://www.nzdl.org/).

REFERENCES

[1] Frank E., Paynter G.W., Witten I.H., Gutwin C. and. Nevill-Manning C.G. (1999). Domain-Specific Keyphrase Extraction. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, Morgan Kaufmann Publishers, San Francisco, CA.

[2] Gutwin, C., Paynter, G., Witten, I.H., Nevill-Manning, C.G., and Frank, E. (1999) Improving Browsing in Digital Libraries With Keyphrase Indexes. *J. Decision Support Systems*. To Appear.

[3] Jones, S. and Paynter G.W. (1999) Topic Based Browsing Within a Digital Library Using Keyphrases. In *Proc. DL'99.*

[4] Witten I.H. (1999) Browsing around a digital library. In *Proc. Australasian Computer Science Conference*, Auckland, New Zealand, 1–14.

[5] Witten, I.H., McNab, R., Jones, S., Apperley, M., Bainbridge, D., and Cunningham, S.J. (1999) Managing Complexity in a Distributed Digital Library. *IEEE Computer*, 32, 2 (1999), 74-79.