# LAKE system at DUC-2006

**Ernesto D’Avanzo**
Department of Communication
Sciences, University of Salerno.
Salerno, Italy
edavanzo@acm.org

**Marcello Frixione**
Department of Communication
Sciences, University of Salerno.
Salerno, Italy
frix@dist.unige.it

**Tsvi Kuflik**
MIS Department,
The University of Haifa.
Haifa, Israel
tsvikak@mis.haifa.ac.il

## ABSTRACT

**The paper discusses the third participation of the LAKE system in the DUC-2006 competition. LAKE is a keyphrase based summarizer system that makes use of linguistic analysis to extract keyphrases from documents. Since the past competition it has been also equipped with a module able to extract sentences from documents. As in the past campaign the system showed a very interesting performance, specially with respect the Linguistic Quality of the summaries created.**

### Keywords

Linguistic Analysis, Patterns, Machine Learning, Naïve bayes Classifier, Keyphrase Extraction.

## INTRODUCTION

LAKE focuses on linguistically motivated keyphrases extraction as the underlying technology for documents summarization. LAKE already participated in the DUC-2004 (D’Avanzo et al., 2004) and DUC-2005 (D’Avanzo et al., 2005). Both past competitions showed that the use of Keyphrase Extraction (hereafter KE) approach for document summarization proved to be not less effective than other approches and in several aspects even among the best. In particular LAKE scored in the middle of the final rank at its first participation to the competition (DUC-2004) and obtained very encouraging results at DUC-2005, especially for the *Linguistic Quality* where LAKE scored among the first positions. Previously, LAKE has also been tested to be as a useful device in text mining application suitable for small devices as well (D’Avanzo and Kuflik, 2005).

This year the DUC competition is essentially the same as last year. Given a topic (question) and a set of 25 relevant documents, the task is to synthesize a fluent, well-organized 250-word summary of the documents that answers the question in the topic statement. It is expected again that LAKE will do well with respect to the linguistic quality which is among the most relevant aspect for an “information consumer”.

The rest of the paper is organized as follows. Section 2 provides a brief background on keyphrases. Section 3 provides a brief introduction to LAKE, its implementation and adaptation for the DUC-2006 scenario. Section 4 presents experimentations results and evaluation. Section 5 concludes with summary suggestions for future work.

## KEYPHRASE EXTRACTION

Keywords, or keyphrases[1], provide semantic metadata that characterize documents, producing an overview of the subject matter and contents of a document.

Keyphrase extraction is a relevant for number of information retrieval related tasks, including document indexing and retrieval, Web page retrieval, text categorization and clustering and summarization, Human and Machine Readable Indexing and Interactive Query Refinement (see (Turney, 2000) and (Gutwin et al., 1998)).

There are two major tasks exploiting keyphrases, keyphrase assignment (KA) and keyphrase extraction (KE) (Turney, 1999).

In a KA task, keyphrases are treated as classes, and techniques from text categorization are used to learn models for assigning a document to a given class. Usually a document may belong to several different classes, based on keyphrases it contains.

In KE task, keyphrases are selected from the body of the input document, without a predefined list. When authors assign keyphrases without a controlled vocabulary (free text keywords or free index terms), typically about 70% to 80% of their keyphrases appear somewhere in the body of their documents (Turney, 1997). This suggests the possibility of using author-assigned free-text keyphrases to train a KE system. In this approach, a document is treated as a set of candidate phrases and the task is to classify each candidate phrases as either a keyphrase or nonkeyphrase (Turney, 1997; Frank et al., 1999).

## LAKE

LAKE (Linguistic Analysis based Keyphrase Extractor) is a keyphrase extraction system based on a supervised learning approach which makes use of linguistic processing of documents. The system uses Naïve Bayes (Mitchell, 1997) as the learning algorithm and TF × IDF term weighting with the *position* of a phrase as features. Unlike other keyphrase exctraction systems, like Kea (Frank et al., 1999) and Extractor (Turney, 1999), LAKE chooses the candidate

[1] Throughout this document we use the latter term to subsume the former

phrases using linguistic knowledge. The candidate phrases generated by LAKE are sequences of Part of Speech containing Multiword expressions and Named Entities. Extraction is driven by a set of ”patterns” which are stored in a pattern database; once there, the main work is done by the learner device. The linguistic database makes LAKE unique in its category.

LAKE is based on three main components: the Linguistic Pre-Processor, the candidate Phrase Extractor and the Candidate Phrase Scorer. In the following section there is a brief description of the system, for more detailed description the reader is referred to previous publications (D’Avanzo et al., 2004, D’Avanzo et al., 2005).

**Linguistic Pre-Processor**

Every document is analyzed by the Linguistic Pre- Processor in the following three consecutive steps: Part of speech analysis, Multiword recognition and Named Entity Recognition

**Candidate Phrase Extractor**

Syntactic patterns that described either a precise and well defined entity or concise events/situations were selected as candidate phrases (e.g. phrases that may be selected as document reorientations). In the former case, the focus was on uni-grams and bi-grams (for instance Named Entity, noun, and sequences of adjective+noun, etc.), while in the latter have been considered longer sequences of parts of speech, often containing verbal forms (for instance noun+verb+adjective+noun). Once all the uni-grams, bi-grams, tri-grams, and four-grams were extracted from the linguistic pre-processor, they were filtered with the patterns defined above. The result of this process is a set of keyphrases that may represent the current document.

As an example, let consider a document belonging to the

**Candidate Phrases Scorer**

The individual candidates keyphrases identified in the previous step are now scored in order to select the most appropriate phrases as representative of the original text. The score is based on a combination of TF ×IDF and first occurrence, i.e. the distance of the candidate phrase from the beginning of the document in which it appears. (These features are commonly used keyphrase-related features.) However, since the frequency of a candidate phrase in the whole collection is not significant, candidate phrases do not appear frequently enough in the collection. It has been decided to estimate the values of the TF ×IDF using the head of the candidate phrase, instead of the phrase itself. According to the principle of headedness (Arampatzis et al., 2000), any phrase has a single word as head. The head is the main verb in the case of verb phrases, and a noun (last noun before any post-modifiers) in noun phrases. As learning algorithm, it has been used the Naïve Bayes Classifier provided by the WEKA package (Witten and Frank, 1999)[2].

The classifier was trained in the following way on a corpus with the available keyphrases. From the document collection we extracted all the nouns and the verbs. Each of them was marked as a positive example of a relevant keyphrase for a certain document if it was present in the assessor’s judgment of that document; otherwise it was marked as a negative example. Then the two features (i.e. TF × IDF and first occurrence) were calculated for each word. The classifier was trained upon this material and a ranked word list was returned. The system automatically looks in the candidate phrases for those phrases containing these words. The top candidate phrases matching the word output of the classifier are kept. The model obtained is reused in the subsequent steps. When a new document or corpus is ready we use the pre-processor module to prepare the candidate phrases. The model we got in the training is then used to score the phrases obtained. In this case the pre-processing part is the same. So, using the model we got in the training, we extract nouns and verbs from documents, and then we keep the candidate phrases containing them.

For DUC-2005, two new parameters were added to the system, one is the maximum number of words allowed in a keyphrase and the second is the maximum number of keyphrases to be extracted from a document.

**Adaptation of LAKE to DUC-2005**

The DUC-2005 task was: given a *user profile*, a DUC *topic*, and a cluster of documents relevant to the DUC topic, were asked to create from the documents a brief, well-organized, fluent summary addressing the need for information expressed in the topic, at the level of granularity specified in the user profile.

The requirement from LAKE was then to select the most representative keyphrases that have the highest *relevance* and *coverage* scores of a set of document, given the topic and profile.

The *relevance* of a keyphrase list $kl_j$ with respect to a cluster $C_j$ is computed considering the frequency of the keyphrases composing the list. The intuition is that keyphrases with higher frequency bring the more relevant information in the cluster:

$$relevance(kl_j) = \frac{\sum_{w=1}^{n} freq(w,kl_j)}{freq(w,C_j)}$$

where $freq(w,kl_j)$ is the count of a word $w$ in a certain document and $freq(w,C_j)$ is the count of $w$ in all the documents in the cluster $C_{J.}$

The *Coverage* of a keyphrase list $kl_j$ is an indication of the amount of information that the keyphrase list contain with respect to the total amount of information included in a cluster of documents:

$$\mathrm{cov}\,erage(kl_j,C) = \frac{length(kl_j)}{\max length(kl_j,C)}$$

where $length(kl_j)$ is the number of keyphrases extracted from document $j$ and $maxlength(kl_j,C)$ is the length of the longest keyphrase list extracted from a document belong-

[2] http://www.cs.waikato.ac.nz/~ml/weka/

ing to cluster $C_j$. The intuition underlying being that the longer the keyphrase list, the more is its coverage for a certain cluster.

*Relevance* and *Coverage* are combined according to the following formula:

$$rep(kl_j) = relevance(kl_j, C) \times \mathrm{cov}erage(kl_j C)$$

which gives an overall measure of the representativeness of a keyphrase list for a certain document with respect to a cluster.

Finally, the keyphrase list which maximize the two parameters is selected as the most representative of the cluster and each keyphrase is substituted with the whole sentence in which it appears, until a 250 word summary is built.

## LAKE at DUC-2006

Like in the past campaign, NIST launched several evaluation task to judge the effectiveness of the participant systems. In particular, NIST:

1. Manually evaluated the linguistic well-formedness of each submitted summary;
2. Manually evaluate the relative responsiveness of each submitted summary to the topic;
3. Run the latest version of ROUGE to compute ROUGE-2 and ROUGE-SU4
4. Calculate overlap in Basic Elements (BE) between automatic and manual summaries.

In addition to the above evaluations, Columbia University organized. An optional Pyramid evaluation for a subset of the topics.

Unfortunately, this year we couldn't participate in the Pyramid evaluation. Therefore, in the following only on the first four evaluation metrics are reported.

*Linguistic Quality* assess how readable and fluent the summaries are. Five *Quality Questions* were used:

1. Grammaticality
2. Non-redundancy
3. Referential clarity
4. Focus
5. Structure and Coherence

All linguistic quality questions were assessed on a five-point scale from "1" (very poor) to "5" (very good).

For this metric LAKE scored 3.7 which is more then one standard deviation above the mean of all system (mean 3.35, standard deviation 0.32) and shared the 3rd place together with two additional systems (the best one scored 4.1) out of 34 systems.

As for responsiveness the evaluation assesses how well each summary responds to the topic. The score was an integer between 1 and 5, with 1 being least responsive and 5 being most responsive. The overall responsiveness score obtained by LAKE was 2.2, which is below the mean (2.56 with standard deviation of 0.28) and ranked, in overall, 13th out of 34 systems (in fact, shard places 13 to 19 with other 6 systems).

While the previous two evaluation tasks were manually performed by NIST's assessors, two other automatic metrics have been used. First, ROUGE-2 and ROUGE-SU4 scores were computed by running ROUGE-1.5.5 with stemming but without removal of stopwords. Second, Basic Elements (BE) scores were computed by first using the tools in BE-1.1 to extract BE from each sentence-segmented summary. The BE-F were then matched by running ROUGE-1.5.5 with stemming, using the Head-Modifier (HM) matching criterion.

For ROUGE-2, LAKE scored 0.07 where the men of all systems was 0.07 with standard deviation of 0.01, hence even though LAKE ranked 28th out of 34 systems, in practice the differences among the participating systems are relatively small (the best result was 0.12). For ROUGE-SU4 LAKE scored 0.12, which is a little below the mean of all systems that was 0.13 with standard deviation of 0.02 (the best result was 0.16) and was ranked 30th out of 34 systems. Again, it must be noticed that in both cases the value of the scores range in a small interval.

Finally, for the BE-score LAKE scored 0.03, which is a little below the mean of all systems that was 0.04 with standard deviation of 0.01, and ranked 30th out of 34 systems.

Figures 1 and 2 ilustrates the overall results of DUC competitors, where lake is numbered 34 .

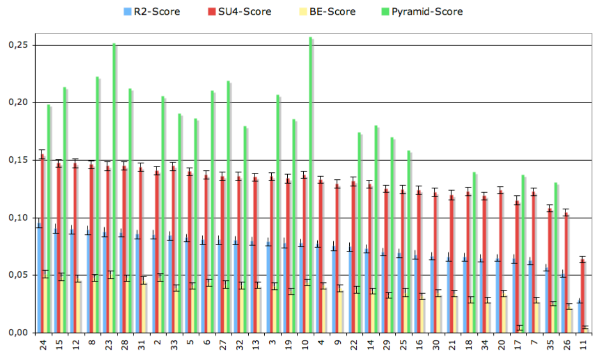
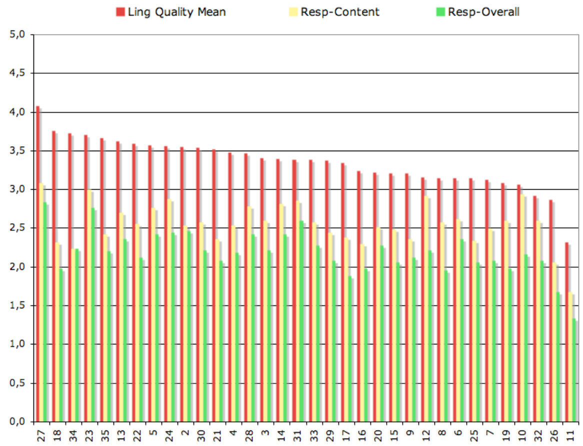


Figure 1. DUC Automatic Evaluation

Figure 1 provides an illustration of the automatic evaluation. It is easy to see that the differences between the systems are relatively small, as discussed above.

Figure 2 ilustrates the overall human evaluation. As can be expected for a linguistically motivated system, LAKE scores very high in the linguistic part and quite high on the responsiveness part.

Figure 2. DUC Human Evaluation



LALE is is dentified in the graph by the number 34

## CONCLUSIONS AND FUTURE WORK

LAKE, essentially, uses a keyphrase extraction approach to summarize documents, in order to make them readable by their human customers in addition to providing a concise summary of their content. This intuition revealed to be fruitful in several applications. For DUC-2005 and DUC-2006, the system has been extended to extract sentences from documents. The extension grounds on the representetiveness of a list of keyphrases. In other words, for each cluster of documents, the system chooses a list of keyphrases that best represent that cluster. Afterward, all sentences of the cluster that contain these keyphrases are extracted. LAKE makes also a good use of linguistic analysis. In fact, among the keyphrases (or sentences) extracted it awards those containing Named Entities, Multiwords, and other significant linguistic patterns. Results obtained are quite encouraging to this end. Especially when considering human evaluation. LAKE, in fact, ranked as one of the top systems with respect to the *Linguistic Quality* of the summaries extracted.

In the future, we plan to improve the aspects related to the automatic evaluation and improve further the use of linguistic patterns and the use of Web as for building summary closer to the information need expressed by the topics.

## REFERENCES

[1] Arampatzis, A. , van der Weide, T. , Koster, C. and van Bommel, P. 2000. An evaluation of linguistically-motivated indexing schemes. In In Proceedings of the BCSIRSG ’2000.

[2] D'Avanzo E., Magnini B. A Keyphrase-Based Approach to Summarization: the LAKE System at DUC-2005. DUC Workshop. Proceedings of Human Language Technology Conference / Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005). Vancouver, B.C., Canada, October 6-8, 2005

[3] D’Avanzo, E., Magnini, B., and Vallin, A. 2004. Keyphrase extraction for summarization purposes: The lake system at duc-2004. In LT/EMNLP. Human Language Technology Conference. Conference on Empirical Methods in NaturalLanguage Processing.

[4] D’Avanzo E. Using Keyphrases fo Text Mining: Applications and Evaluation. PhD Dissertation Series. department of Information and Communication Sciences, University of Trento. December 2005.

[5] Fellbaum, C. 1998. WordNet: An Electronic Lexical Database. MIT Press.

[6] Frank E., Paynter G.W., Witten I.H., Gutwin C. and Nevill-Manning C.G. (1999) "Domain-specific keyphrase extraction" Proc. Sixteenth International Joint Conference on Artificial Intelligence, Morgan Kaufmann Publishers, San Francisco, CA, pp. 668-673.

[7] Gutwin, C., Paynter, G., Witten, I. NevillManning, C. and Frank, E.. 1998. Improving browsing in digital libraries with keyphrase indexes. Technical report, Department of Computer Science, University of Saskatchewan, Canada.

[8] Mitchell, T. 1997. Machine Learning. McGraw-Hill.

[9] Turney, P.D. 2000. Learning algorithms for keyphrase extraction. Information Retrieval, 2 (4):303–336.

[10]Turney, P. (1999). Learning to extract keyphrases from text', Technical Report ERB-1057, National Research Council, Institute for Information Technology.

[11] Witten, H. I., and Frank, E. 1999. Data Mining: Practi cal Machine Learning Tools and Techniques with Java.