# Evaluating Statistically Generated Phrases

*Raymond Wan* *Alistair Moffat*

Department of Computer Science and Software Engineering
The University of Melbourne
Victoria 3010, Australia

{*rwan,alistair*}*@cs.mu.oz.au*

**Abstract:** *An experimental framework for the evaluation of statistically generated phrases is described. Two methods are considered. The first compares the phrases with those found by a natural language processing system; while the second examines the ability to locate all instances of each phrase. The motivation for this work is to determine the usefulness of statistically generated phrases in a document retrieval system.*

**Keywords:** Information retrieval, phrase evaluation.

## 1 Introduction

Information retrieval systems generally index document collections by treating each document as a bag of unrelated words. Queries are similarly taken to be bags of words, and the similarity heuristic for scoring documents is based on word-level statistics. There are, however, situations in which phrase-based querying is desirable.

Several methods exist for obtaining phrases. If the phrases provided by the system must meet the linguistic definition of "phrase", they can be defined manually, or through natural language processing techniques [Evans and Zhai, 1996, Frank et al., 1999].

This paper focuses on the phrases produced by two different methods of automatic phrase derivation. The first involves a compression algorithm called RE-PAIR [Larsson and Moffat, 2000], which creates phrases based on the frequency of pairs of symbols. The second method uses a natural language processing system called LINK GRAMMAR [Grinberg et al., 1995]. The two sets of phrases are compared, as a way of evaluating the usefulness of RE-PAIR as an automated phrase-generation process.

The phrase evaluation framework described by this paper is similar to the one presented by Wolff [1980]. Wolff used a RE-PAIR-like system dubbed MK10 which formed phrases based on the part-of-speech categories of words (such as "noun" or "verb"), rather than the words themselves. The quality of the phrases was then compared with ones selected by a human linguist. Other methods of phrase evaluation that have been used include user studies [Jones and Paynter, 2001] and precision and recall [Croft et al., 1991].

**Proceedings of the 8th Australasian Document Computing Symposium, Canberra, Australia, December 15, 2003.**

## 2 Statistical generation of phrases

RE-PAIR is an off-line dictionary-based compression algorithm [Larsson and Moffat, 2000]. It reduces the length of a message by identifying the most frequently occurring pair of adjacent symbols, and replacing every occurrence of that pair with a new symbol. This process is applied recursively until no pair of adjacent symbols appears twice. Wan [2003] showed how a pre-processor could first separate a message into an alternating sequence of words and non-words, so that RE-PAIR would form phrases of words.

Two outputs are produced by word-based RE-PAIR. A dictionary of phrases called the *phrase hierarchy* is built, which lists all of the distinct symbols that appear in the word sequence, as well as all of the symbols introduced by RE-PAIR, and their expansions. The second output is a sequence of references to entries in the phrase hierarchy called the *reduced sequence*. Of these two streams, this paper is primarily concerned with the phrase hierarchy.

Figure 1 shows how the symbols in the RE-PAIR phrase hierarchy can be arranged into a graph structure. Each symbol in the hierarchy is represented as a node, with six pointers emanating from each. Even though only a single phrase is shown for each pointer in the figure, in practice, a pointer leads to a list of zero or more phrases that share a common attribute.

Two of the six pointers are *child pointers*, which designate the two subordinate phrases that comprise the current one. The two *sibling pointers* lead from the current symbol $\alpha$ to symbols that have the same left component and the same right component as $\alpha$, respectively. Longer symbols are obtained by using the *parent pointers*. The left parent pointer of $\alpha$ leads to a symbol which is a left extension of the current symbol. With its sibling pointers, all other phrases that are left extensions of $\alpha$ can be located. All left extensions of $\alpha$ have $\alpha$ as their right child. The RE-PAIR algorithm ensures that every symbol in the phrase hierarchy can be decomposed into a sequence of underlying words.

## 3 Natural phrase derivation

In contrast to RE-PAIR, the LINK GRAMMAR system employs natural language processing (NLP) techniques for phrase selection [Grinberg et al., 1995]. The software is available from `http:`

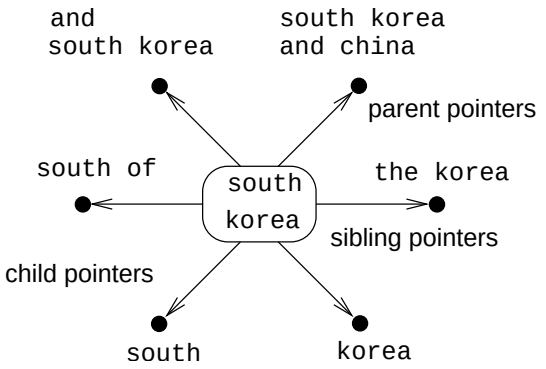


Figure 1: *Directed graph showing the relationship between symbols, with the center one (“*south korea*”) as the focus.*

//www.link.cs.cmu.edu/link/, and the current version is 4.1, dated August 2000.

LINK GRAMMAR processes a sentence at a time by first assigning each word to a part-of-speech (POS) category. The categories, together with a set of rules, are used to create edges which link words with each other. For example, the two words “the account” can merge because of a right-facing “D+” link on the word “the”, and a corresponding left-facing “D−” link on the noun “account”. In both cases, “D” represents a determiner, such as “an”.

If all words in a sentence can be linked, then the sentence is grammatical, and a valid *linkage* exists. Multiple linkages may be possible for a sentence, and each linkage is scored based on factors such as the number of words involved, and variations to the POS categories. In the tests described here, only the highest scoring linkage is used, which is then post-processed so that constituents (phrases) can be established.

## 4 Preparing for the experiments

Experiments were conducted on a collection of 7,520 news articles from the Wall Street Journal in 1987 (WSJ). These articles form part of Disk 1 of the TIPSTER collection of the Text REtrieval Conference (TREC) [Harman, 1995]. The articles include SGML markup and total just over 20 MB in size.

The WSJ collection and the phrases from it are transformed before and after the two phrase derivation systems, as detailed in Figure 2. First, the throughput of LINK GRAMMAR is improved through the removal of document sections with only a small chance of containing grammatical sentences. A sample article from WSJ is shown in Figure 3, with the removed sections shaded. The purpose of the unshaded boxes is mentioned below.

A second process identifies only simple noun phrases from the output of LINK GRAMMAR. Generally, phrase queries take the form of noun phrases, a fact which has been exploited by other retrieval systems (for example, Evans and Zhai [1996]). Simple noun phrases do not contain conjunctions (“and”), or prepositional phrases.

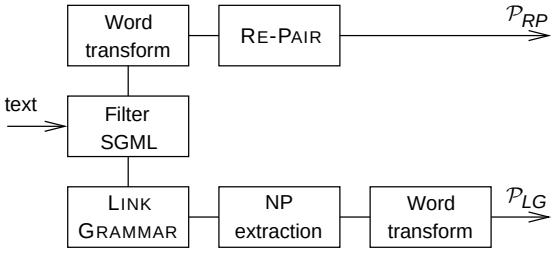


Figure 2: *Transformations applied before and after* RE-PAIR *or* LINK GRAMMAR.

<DOC> <DOCNO> WSJ870323-0181 </DOCNO>
<HL> South Korea 's Current Account </HL>
<DD> 03/23/87</DD>
<SO> WALL STREET JOURNAL (J)</SO>
<IN> FREST MONETARY NEWS, FOREIGN EXCHANGE,
TRADE (MON) </IN>
<DATELINE> SEOUL, South Korea </DATELINE>
<TEXT>
South Korea posted a surplus
on its current account of $419 million
in February, in contrast to a deficit
of $112 million a year earlier,
the government said. The current account
comprises trade in goods and services
and some unilateral transfers.
</TEXT> </DOC>

Figure 3: *An article from* WSJ*. Sections of the article that were removed prior to the experiments are shown shaded. Unshaded boxes indicate the noun phrases identified by* LINK GRAMMAR.

Finally, each individual word was transformed for both RE-PAIR and LINK GRAMMAR in order to reduce the size of the word lexicon, and to improve overall phrase quality. Words were restricted to being no more than 16 alphanumeric characters in length, case folded to lowercase, and contracted to their root form using the Porter stemming algorithm [Porter, 1980]. These operations are performed as a post-processing step for LINK GRAMMAR, since it requires case and stemming information for proper POS classification.

At the conclusion of these steps, two sets of phrases are established, $\mathcal{P}_{RP}$ with RE-PAIR and $\mathcal{P}_{LG}$ with LINK GRAMMAR. Figures 3 and 4 compare a WSJ article parsed with these two systems. The boxes in Figure 3 indicate noun phrases which were isolated by LINK GRAMMAR. The unshaded parts of the article have been duplicated in Figure 4 for RE-PAIR. Each symbol in the RE-PAIR sequence has been expanded into words to produce the top-level boxes of the figure. Within each top-level box are smaller boxes and underlines, which indicate how each symbol was constructed. Underlines and boxes are omitted where no ambiguity exists.

Even though there are phrases such as “South Korea”, RE-PAIR’s frequency-based heuristic breaks up ones that LINK GRAMMAR identifies – the phrase “to a”

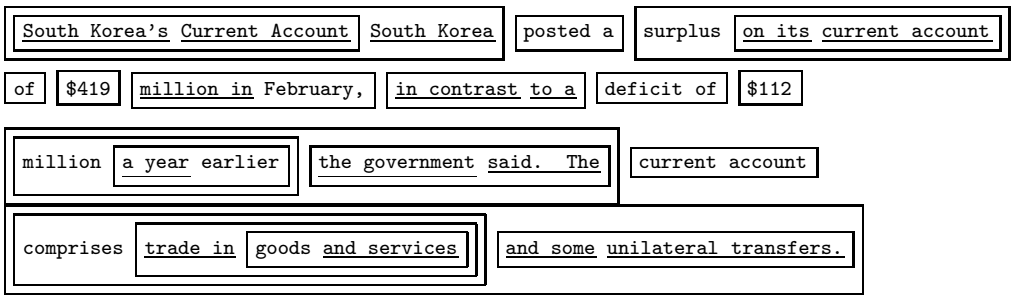


Figure 4: *The phrases identified by* RE-PAIR *for the article shown in Figure 3.*

| | Before | After |
|---|---|---|
| Number of words | 3,126,789 | 3,126,789 |
| Vocabulary size | 71,647 | 37,623 |
| Phrase hierarchy size | 269,042 | 229,102 |
| Reduced sequence length | 1,621,331 | 1,544,705 |

Table 1: *The effect of case folding and stemming on* `WSJ` *with* RE-PAIR.

was selected by RE-PAIR on the second line, instead of "`a deficit`". Also, since RE-PAIR does not process a sentence at a time, phrases like "`said. The`" on the third line appear. The restriction on LINK GRAMMAR to only identify simple noun phrases prevents "`goods and services`" on the last line from being added to $\mathcal{P}_{LG}$.

## 5 Experiments

Three aspects of the experiments are reported in this section: the effect of case folding and stemming on RE-PAIR, a comparison between the two sets of phrases, and the recall effectiveness of the symbols in $\mathcal{P}_{RP}$.

### Case folding and stemming

Table 1 shows the impact that case folding and stemming has on the `WSJ` test file. The number of distinct words is halved, while both the size of $\mathcal{P}_{RP}$ and the length of the resultant sequence are reduced.

### Comparison with LINK GRAMMAR

Filtering `WSJ` lowered its size from 20 MB to 18.1 MB. There are 177,672 sentences in the document set, yielding 745,361 noun phrases, of which 237,085 were distinct. Statistics about this set of phrases and the ones from RE-PAIR are displayed in Table 2. In both cases, the phrases have been divided into groups based on their length (in words), with singletons not considered.

The last two columns in Table 2 show that only 30% of the noun phrases of length two also exist in $\mathcal{P}_{RP}$. This value quickly diminishes as phrases increase in length – a problem caused by the RE-PAIR requirement that a phrase cannot be recognized unless it appears at least twice. Overall, just 15% of the noun phrases in $\mathcal{P}_{LG}$ occur in the RE-PAIR phrase hierarchy.

| Length | $\mathcal{P}_{RP}$ | $\mathcal{P}_{LG}$ | $\mathcal{P}_{RP} \cap \mathcal{P}_{LG}$ | Ratio |
|---|---|---|---|---|
| 2 | 79,793 | 78,570 | 22,044 | 0.281 |
| 3 | 60,722 | 72,549 | 6,026 | 0.083 |
| 4 | 25,669 | 32,436 | 1,475 | 0.045 |
| 5 | 10,797 | 12,312 | 383 | 0.031 |
| 6 | 5,014 | 4,956 | 129 | 0.026 |
| 7 | 2,761 | 1,695 | 25 | 0.015 |
| 8 | 1,728 | 664 | 4 | 0.006 |
| 9 | 1,141 | 248 | 3 | 0.012 |
| 10+ | 3,854 | 203 | 3 | 0.015 |
| Overall | 191,478 | 203,633 | 30,092 | 0.148 |

Table 2: *Relative size of* $\mathcal{P}_{\mathrm{RP}}$ *and* $\mathcal{P}_{\mathrm{LG}}$*, distributed by phrase length in words. The size of the intersection of the two sets is shown in the fourth column, and the ratio* $|\mathcal{P}_{\mathrm{RP}} \cap \mathcal{P}_{\mathrm{LG}}|/|\mathcal{P}_{\mathrm{LG}}|$ *is calculated last.*

### Measuring recall

One important way of assessing the effectiveness of a retrieval mechanism is by calculating its *recall* – the fraction of the set of valid answers that are fetched in response to a query. In general, if there are a total of $f$ valid answers, and $s$ of them are identified, then the recall of the system is given by the ratio $s/f$. An ideal system attains recall of 1.0 for all requests to it.

The statistical heuristic employed by RE-PAIR makes it difficult to achieve very high recall on phrase-based searches. Suppose that a search for the $i$th phrase in some set of phrases locates $s_i$ occurrences, but that there are actually $f_i$ locations in the original message at which that phrase occurs. Then, two ways of calculating the average recall is as follows:

$$\frac{1}{n}\sum_{i=1}^{n}\frac{s_i}{f_i} \tag{1}$$

$$\frac{\sum_{i=1}^{n} s_i}{\sum_{i=1}^{n} f_i} \tag{2}$$

Equation 1 assumes that each retrieval task involves a search for a single symbol $\alpha$, and that every symbol is equally likely to be the subject of a search. The alternative formulation in Equation 2 ensures that the contribution of a symbol to the total recall is proportional to its frequency of appearance in the original text.

| Length | Unweighted | | | Weighted |
|---|---|---|---|---|
| | Median | Mean | StdDev | mean |
| 2 | 0.736 | 0.700 | 0.292 | 0.611 |
| 3 | 1.000 | 0.786 | 0.262 | 0.635 |
| 4 | 1.000 | 0.853 | 0.228 | 0.706 |
| 5 | 1.000 | 0.902 | 0.196 | 0.788 |
| 6 | 1.000 | 0.928 | 0.174 | 0.752 |
| 7 | 1.000 | 0.942 | 0.157 | 0.875 |
| 8 | 1.000 | 0.959 | 0.132 | 0.875 |
| 9 | 1.000 | 0.965 | 0.129 | 0.897 |
| 10+ | 1.000 | 0.966 | 0.115 | 0.923 |
| Overall | 1.000 | 0.778 | 0.273 | 0.630 |

Table 3: *Recall for all symbols in* $\mathcal{P}_{\text{RP}}$.

Statistics for $\mathcal{P}_{RP}$ are shown in Table 3, grouped by phrase length. Equation 1 produces the column labelled "unweighted mean", which is no less than $0.700$. Recall effectiveness is worst for phrases of length two words. Thereafter, the median remains at $1.000$, while the mean rises and the standard deviation decreases. The weighted mean recall (last column) is lower than the unweighted one. The reason is that recall is generally high for many symbols, but in some cases $f_i$ is high, yet $s_i$ is quite low. This occurs when $i$ was split by another replacement operation which was more frequent.

## 6 Conclusion

An experimental framework encompassing phrase evaluation and recall measurement has been proposed for statistically generated phrases. Phrases were evaluated through comparison with ones extracted by a natural language processing system. While RE-PAIR was evaluated using LINK GRAMMAR as a baseline, these metrics are also applicable to other statistical and NLP systems.

The results show that $15\%$ of the noun phrases identified by LINK GRAMMAR also appear in the phrase hierarchy generated by RE-PAIR. While this result is somewhat disappointing, it is compensated for by speed of computation. Phrase generation using RE-PAIR for `WSJ` requires about $18$ seconds of CPU time (933 MHz Pentium III with 1 GB RAM and 256 kB on-die cache). In stark contrast, LINK GRAMMAR required between five and six hours for each 1 MB block of `WSJ`. While the time requirements would vary for other systems, it is expected that such a difference between these two phrase selection techniques will remain.

Hence, an interesting direction for future work is to integrate the two methods so that RE-PAIR's heuristics are augmented with basic linguistic knowledge. A useful compromise between phrase effectiveness and computational efficiency might then be possible.

Recall with the RE-PAIR phrases was also lower than expected. One improvement might be to restrict the type of phrases formed. Another possibility is to modify RE-PAIR so that only frequent pairs are replaced. Alternatively, RE-PAIR could be used purely for phrase identification, with an explicit index being deployed for phrase searching.

The two recall metrics proposed are directed at statistically generated phrases, with overlap in the phrase hierarchy. If no overlap is permitted, then recall can be expected to be perfect. Such a system would be analogous to a Boolean query system with an explicit phrase-based index. Our work here explores options that might be pursued if the time or space overheads of constructing and storing an index cannot be tolerated.

**Acknowledgements** Vo Ngoc Anh (University of Melbourne) provided helpful input. This work was supported by the Australian Research Council.

## References

W. B. Croft, H. R. Turtle, and D. D. Lewis. The use of phrases and structured queries in information retrieval. In *Proc. 14th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 32–45, 1991.

D. A. Evans and C. Zhai. Noun-phrase analysis in unrestricted text for information retrieval. In *Proc. ACL-96, 34th Annual Meeting of the Association for Computational Linguistics*, pages 17–24, 1996.

E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-Manning. Domain-Specific keyphrase extraction. In *Proc. International Joint Conference on Artificial Intelligence*, pages 668–673, 1999.

D. Grinberg, J. Lafferty, and D. Sleator. A robust parsing algorithm for link grammars. Technical Report CMU-CS-95-125, Carnegie Mellon University, School of Computer Science, 1995.

D. K. Harman. Overview of the second Text REtrieval Conference (TREC-2). *Information Processing and Management*, 31(3):271–289, May 1995.

S. Jones and G. W. Paynter. Human evaluation of Kea, an automatic keyphrasing system. In E. A. Fox and C. L. Borgman, editors, *Proc. First ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 148–156. ACM Press, 2001.

N. J. Larsson and A. Moffat. Offline dictionary-based compression. *Proc. IEEE*, 88(11):1722–1732, November 2000.

M. F. Porter. An algorithm for suffix stripping. *Program*, 14: 130–137, 1980. Reprinted in Readings in Information Retrieval, pages 313–316, 1997. Web site created for algorithm: `http://www.tartarus.org/~martin/PorterStemmer/`.

R. Wan. *Browsing and Searching Compressed Documents*. PhD thesis, University of Melbourne, Australia, 2003. Submitted.

J. G. Wolff. Language acquisition and the discovery of phrase structure. *Language and Speech*, 23(3):255–269, 1980.