FROM N-GRAMS TO COLLOCATIONS AN EVALUATION OF XTRACT

Frank A. Smadja
Department of Computer Science
Columbia University
New York, NY 10027

Abstract

In previous papers we presented methods for retrieving collocations from large samples of texts. We described a tool, **Xtract**, that implements these methods and able to retrieve a wide range of collocations in a two stage process. These methods as well as other related methods however have some limitations. Mainly, the produced collocations do not include any kind of functional information and many of them are invalid. In this paper we introduce methods that address these issues. These methods are implemented in an added third stage to **Xtract** that examines the set of collocations retrieved during the previous two stages to both filter out a number of invalid collocations and add useful syntactic information to the retained ones. By combining parsing and statistical techniques the addition of this third stage has raised the overall precision level of **Xtract** from 40% to 80% with a precision of 94%. In the paper we describe the methods and the evaluation experiments.

1 INTRODUCTION

In the past, several approaches have been proposed to retrieve various types of collocations from the analysis of large samples of textual data. Pairwise associations (bigrams or 2-grams) (*e.g.*, [Smadja, 1988], [Church and Hanks, 1989]) as well as n-word ($n > 2$) associations (or n-grams) (*e.g.*, [Choueka *et al.*, 1983], [Smadja and McKeown, 1990]) were retrieved. These techniques automatically produced large numbers of collocations along with statistical figures intended to reflect their relevance. However, none of these techniques provides functional information along with the collocation. Also, the results produced often contained improper word associations reflecting some spurious aspect of the training corpus that did not stand for true collocations. This paper addresses these two problems.

Previous papers (*e.g.*, [Smadja and McKeown, 1990]) introduced a set of techniques and a tool, **Xtract**, that produces various types of collocations from a two-stage statistical analysis of large textual corpora briefly sketched in the next section. In Sections 3 and 4, we show how robust parsing technology can be used to both filter out a number of invalid collocations as well as add useful syntactic information to the retained ones. This filter/analyzer is implemented in a third stage of **Xtract** that automatically goes over a the output collocations to reject the invalid ones and label the valid ones with syntactic information. For example, if the first two stages of **Xtract** produce the collocation *"make-decision,"* the goal of this third stage is to identify it as a verb-object collocation. If no such syntactic relation is observed, then the collocation is rejected. In Section 5 we present an evaluation of **Xtract** as a collocation retrieval system. The addition of the third stage of **Xtract** has been evaluated to raise the precision of **Xtract** from 40% to 80% and it has a recall of 94%. In this paper we use examples related to the word *"takeover"* from a 10 million word corpus containing stock market reports originating from the Associated Press newswire.

2 FIRST 2 STAGES OF XTRACT, PRODUCING N-GRAMS

In a first stage, **Xtract** uses statistical techniques to retrieve pairs of words (or bigrams) whose common appearances within a single sentence are correlated in the corpus. A bigram is retrieved if its frequency of occurrence is above a certain threshold and if the words are used in relatively rigid ways. Some bigrams produced by the first stage of **Xtract** are given in Table 1: the bigrams all contain the word *"takeover"* and an adjective. In the table, the *distance* parameter indicates the usual distance between the two words. For example, *distance* = 1 indicates that the two words are frequently adjacent in the corpus.

In a second stage, **Xtract** uses the output bigrams to produce collocations involving more than two words (or n-grams). It examines all the sentences containing the bigram and analyzes the statistical distribution of words and parts of speech for each position around the pair. It retains words (or parts of speech) occupying a position with probability greater than a given

threshold. For example, the bigram "average-industrial" produces the n-gram *"the Dow Jones industrial average"* since the words are always used within this compound in the training corpus. Example outputs of the second stage of **Xtract** are given in Figure 1. In the figure, the numbers on the left indicate the frequency of the n-grams in the corpus, NN indicates that a noun is expected at this position, AT indicates that an article is expected, NP stands for a proper noun and VBD stands for a verb in the past tense. See [Smadja and McKeown, 1990] and [Smadja, 1991] for more details on these two stages.

Table 1: Output of Stage 1

w_i	w_j	*distance*
hostile	takeovers	1
hostile	takeover	1
corporate	takeovers	1
hostile	takeovers	2
unwanted	takeover	1
potential	takeover	1
unsolicited	takeover	1
unsuccessful	takeover	1
friendly	takeover	1
takeover	expensive	2
takeover	big	4
big	takeover	1

3 STAGE THREE: SYNTACTICALLY LABELING COLLOCATIONS

In the past, Debili [Debili, 1982] parsed corpora of French texts to identify non-ambiguous predicate argument relations. He then used these relations for disambiguation in parsing. Since then, the advent of robust parsers such as **Cass** [Abney, 1990], **Fidditch** [Hindle, 1983] has made it possible to process large amounts of text with good performance. This enabled Hindle and Rooth [Hindle and Rooth, 1990], to improve Debili's work by using bigram statistics to enhance the task of prepositional phrase attachment. Combining statistical and parsing methods has also been done by Church and his colleagues. In [Church *et al.*, 1989] and [Church *et al.*, 1991] they consider predicate argument relations in the form of questions such as *What does a boat typically do?* They are preprocessing a corpus with the **Fidditch** parser in order to statistically analyze the distribution of the predicates used with a given argument such as *"boat."*

Our goal is different, since we analyze a set of collocations automatically produced by **Xtract** to either enrich them with syntactic information or reject them. For example, if a bigram collocation produced by **Xtract** involves a noun and a verb, the role of Stage 3 of **Xtract** is to determine whether it is a *subject-verb* or a *verb-object* collocation. If no such relation can be identified, then the collocation is rejected. This section presents the algorithm for **Xtract** Stage 3 in some detail. For illustrative purposes we use the example words *takeover* and *thwart* with a distance of 2.

3.1 DESCRIPTION OF THE ALGORITHM

Input: A bigram with some distance information indicating the most probable distance between the two words. For example, *takeover* and *thwart* with a distance of 2.

Output/Goal: Either a syntactic label for the bigram or a rejection. In the case of *takeover* and *thwart* the collocation is accepted and its produced label is VO for *verb-object.*

The algorithm works in the following 3 steps:

3.1.1 Step 1: PRODUCE TAGGED CONCORDANCES

All the sentences in the corpus that contain the two words in this given position are produced. This is done with a concordancing program which is part of **Xtract** (see [Smadja, 1991]). The sentences are labeled with part of speech information by preprocessing the corpus with an automatic stochastic tagger.[1]

3.1.2 Step 2: PARSE THE SENTENCES

Each sentence is then processed by **Cass**, a bottom-up incremental parser [Abney, 1990].[2] **Cass** takes input sentences labeled with part of speech and attempts to identify syntactic structure. One of **Cass** modules identifies predicate argument relations. We use this module to produce binary syntactic relations (or labels) such as *"verb-object"* (VO), *"verb-subject"* (VS), *"noun-adjective"* (NJ), and *"noun-noun"* (NN). Consider Sentence (1) below and all the labels as produced by **Cass** on it.

(1) *"Under the recapitalization plan it proposed to* **thwart** *the* **takeover.***"*

label	bigram
SV	it proposed
NN	recapitalization plan
VO	thwart takeover

For each sentence in the concordance set, from the output of **Cass**, **Xtract** determines the syntactic relation of the two words among VO, SV, NJ, NN and assigns this label to the sentence. If no such relation is observed, **Xtract** associates the label U (for undefined) to the sentence. We note *label*[*id*] the label associated

[1] For this, we use the part of speech tagger described in [Church, 1988]. This program was developed at Bell Laboratories by Ken Church.

[2] The parser has been developed at Bell Communication Research by Steve Abney, **Cass** stands for Cascaded Analysis of Syntactic Structure. I am much grateful to Steve Abney to help us use and customize **Cass** for this work.

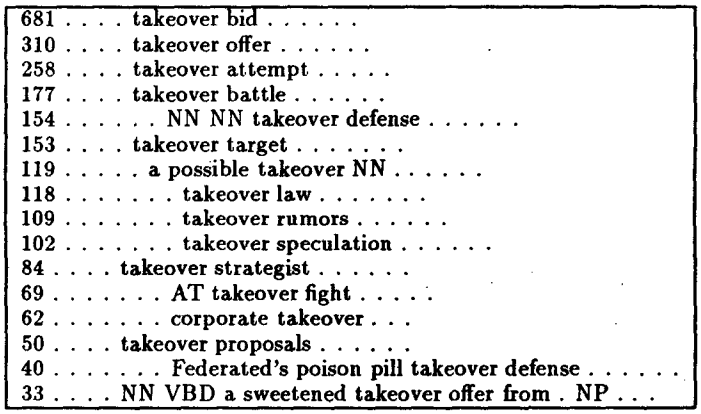

Figure 1: Some n-grams containing *"takeover"*

with Sentence *id*. For example, the label for Sentence (1) is: $label[1] = VO$.

3.1.3 Step 3: REJECT OR LABEL COLLOCATION

This last step consists of deciding on a label for the bigram from the set of $label[id]$'s. For this, we count the frequency of each label for the bigram and perform a statistical analysis of this distribution. A collocation is accepted if the two seed words are consistently used with the same syntactic relation. More precisely, the collocation is accepted if and only if there is a label $\mathcal{L} \neq U$ satisfying the following inequation:

$$\boxed{probability(label[id] = \mathcal{L}) > T}$$

in which T is a given threshold to be determined by the experimenter. A collocation is thus rejected if no valid label satisfies the inequation or if U satisfies it.

Figure 2 lists some accepted collocations in the format produced by **Xtract** with their syntactic labels. For these examples, the threshold T was set to 80%. For each collocation, the first line is the output of the first stage of **Xtract**. It is the seed bigram with the distance between the two words. The second line is the output of the second stage of **Xtract**, it is a multiple word collocation (or n-gram). The numbers on the left indicate the frequency of occurrence of the n-gram in the corpus. The third line indicates the syntactic label as determined by the third stage of **Xtract**. Finally, the last lines simply list an example sentence and the position of the collocation in the sentence.

Such collocations can then be used for various purposes including lexicography, spelling correction, speech recognition and language generation. In [Smadja and McKeown, 1990] and [Smadja, 1991] we describe how they are used to build a lexicon for language generation in the domain of stock market reports.

4 A LEXICOGRAPHIC EVALUATION

The third stage of **Xtract** can thus be considered as a retrieval system which retrieves valid collocations from a set of candidates. This section describes an evaluation experiment of the third stage of **Xtract** as a retrieval system. Evaluation of retrieval systems is usually done with the help of two parameters: *precision* and *recall* [Salton, 1989]. Precision of a retrieval system is defined as the ratio of retrieved valid elements divided by the total number of retrieved elements [Salton, 1989]. It measures the quality of the retrieved material. Recall is defined as the ratio of retrieved valid elements divided by the total number of valid elements. It measures the effectiveness of the system. This section presents an evaluation of the retrieval performance of the third stage of **Xtract**.

4.1 THE EVALUATION EXPERIMENT

Deciding whether a given word combination is a valid or invalid collocation is actually a difficult task that is best done by a lexicographer. Jeffery Triggs is a lexicographer working for Oxford English Dictionary (OED) coordinating the North American Readers program of OED at Bell Communication Research. Jeffery Triggs agreed to manually go over several thousands collocations.[3]

We randomly selected a subset of about 4,000 collocations that contained the information compiled by **Xtract** after the first 2 stages. This data set was then the subject of the following experiment.

We gave the 4,000 collocations to evaluate to the lexicographer, asking him to select the ones that he

[3] I am grateful to Jeffery whose professionalism and kindness helped me understand some of the difficulty of lexicography. Without him this evaluation would not have been possible.

takeover bid -1
681 . . . . takeover bid IN . . . . .
Syntactic Label: NN
10 11
An investment partnership on Friday offered to sweeten its
takeover bid for Gencorp Inc.

takeover fight -1
69 . . . . . AT takeover fight IN 69
Syntactic Label: NN
10 11
Later last year Hanson won a hostile 3.9 billion takeover fight for Imperial Group
the giant British food tobacco and brewing conglomerate and raised more than 1.4
billion pounds from the sale of Imperial s Courage brewing operation and
its leisure products businesses.

takeover thwart 2
44 . . . . . to thwart AT takeover NN 44
Syntactic Label: VO
13 11
The 48.50 a share offer announced Sunday is designed to thwart a takeover bid
by GAF Corp.

takeover make 2
68 . . . . . MD make a takeover NN . JJ 68
Syntactic Label: VO
14 12
Meanwhile the North Carolina Senate approved a bill Tuesday that would make a
takeover of North Carolina based companies more difficult and the House was
expected to approve the measure before the end of the week.

takeover related -1
59 . . . . takeover related 59
Syntactic Label: SV
2 3
Among takeover related issues Kidde jumped 2 to 66.

Figure 2: Some examples of collocations with *"takeover"*

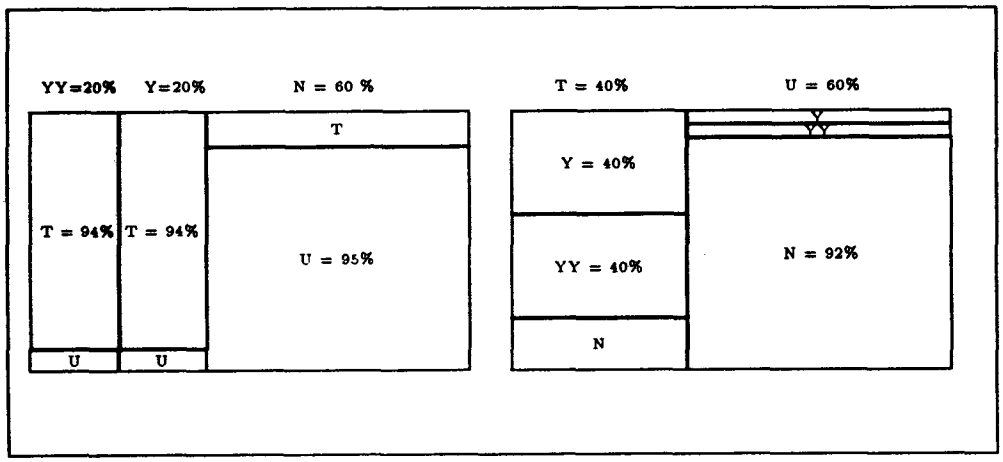

Figure 3: Overlap of the manual and automatic evaluations

would consider for a domain specific dictionary and to cross out the others. The lexicographer came up with three simple tags, **YY**, **Y** and **N**. Both **Y** and **YY** are good collocations, and **N** are bad collocations. The difference between **YY** and **Y** is that **Y** collocations are of better quality than **YY** collocations. **YY** collocations are often too specific to be included in a dictionary, or some words are missing, etc. After Stage 2, about 20% of the collocations are **Y**, about 20% are **YY**, and about 60% are **N**. This told us that the precision of **Xtract** at Stage 2 was only about 40 %.

Although this would seem like a poor precision, one should compare it with the much lower rates currently in practice in lexicography. For the OED, for example, the first stage roughly consists of reading numerous documents to identify new or *interesting* expressions. This task is performed by professional *readers*. For the OED, the readers for the American program alone produce some 10,000 expressions a month. These lists are then sent off to the dictionary and go through several rounds of careful analysis before actually being submitted to the dictionary. The ratio of proposed candidates to good candidates is usually low. For example, out of the 10,000 expressions proposed each month, less than 400 are serious candidate for the OED, which represents a current rate of 4%. Automatically producing lists of candidate expressions could actually be of great help to lexicographers and even a precision of 40% would be helpful. Such lexicographic tools could, for example, help readers retrieve sublanguage specific expressions by providing them with lists of candidate collocations. The lexicographer then manually examines the list to remove the irrelevant data. Even low precision is useful for lexicographers as manual filtering is much faster than manual scanning of the documents [Marcus, 1990]. Such techniques are not able to replace readers though, as they are not designed to identify low frequency expressions, whereas a human reader immediately identifies interesting expressions with as few as one occurrence.

The second stage of this experiment was to use **Xtract** Stage 3 to filter out and label the sample set of collocations. As described in Section 3, there are several valid labels (VO, VS, NN, etc.). In this experiment, we grouped them under a single label: T. There is only one non-valid label: U (for unlabeled). A T collocation is thus accepted by **Xtract** Stage 3, and a U collocation is rejected. The results of the use of Stage 3 on the sample set of collocations are similar to the manual evaluation in terms of numbers: about 40% of the collocations were labeled (T) by **Xtract** Stage 3, and about 60% were rejected (U).

Figure 3 shows the overlap of the classifications made by **Xtract** and the lexicographer. In the figure, the first diagram on the left represents the breakdown in T and U of each of the manual categories ($Y - YY$ and N). The diagram on the right represents the breakdown in $Y - YY$ and N of the the T and U categories. For example, the first column of the diagram on the left represents the application of **Xtract** Stage 3 on the **YY** collocations. It shows that 94% of the collocations accepted by the lexicographer were also accepted by **Xtract**. In other words, this means that the recall of the third stage of **Xtract** is 94%. The first column of the diagram on the right represents the lexicographic evaluation of the collocations automatically accepted by **Xtract**. It shows that about 80% of the T collocations were accepted by the lexicographer and that about 20% were rejected. This shows that precision was raised from 40% to 80% with the addition of **Xtract** Stage 3. In summary, these experiments allowed us to evaluate Stage 3 as a retrieval system. The results are:

Precision = 80%	**Recall** = 94%

5 SUMMARY AND CONTRIBUTIONS

In this paper, we described a new set of techniques for syntactically filtering and labeling collocations. Using such techniques for post processing the set of collocations produced by **Xtract** has two major results. First, it adds syntax to the collocations which is necessary for computational use. Second, it provides considerable improvement to the quality of the retrieved collocations as the precision of **Xtract** is raised from 40% to 80% with a recall of 94%.

By combining statistical techniques with a sophisticated robust parser we have been able to design and implement some original techniques for the automatic extraction of collocations. Results so far are very encouraging and they indicate that more efforts should be made at combining statistical techniques with more symbolic ones.

ACKNOWLEDGMENTS

The research reported in this paper was partially supported by DARPA grant N00039-84-C-0165, by NSF grant IRT-84-51438 and by ONR grant N00014-89-J-1782. Most of this work is also done in collaboration with Bell Communication Research, 445 South Street, Morristown, NJ 07960-1910. I wish to express my thanks to Kathy McKeown for her comments on the research presented in this paper. I also wish to thank Dorée Seligmann and Michael Elhadad for the time they spent discussing this paper and other topics with me.

References

[Abney, 1990] S. Abney. Rapid Incremental Parsing with Repair. In *Waterloo Conference on Electronic Text Research*, 1990.

[Choueka *et al.*, 1983] Y. Choueka, T. Klein, and E. Neuwitz. Automatic Retrieval of Frequent Idiomatic and Collocational Expressions in a Large Cor-

pus. *Journal for Literary and Linguistic computing*, 4:34–38, 1983.

[Church and Hanks, 1989] K. Church and K. Hanks. Word Association Norms, Mutual Information, and Lexicography. In *Proceedings of the 27th meeting of the ACL*, pages 76–83. Association for Computational Linguistics, 1989. Also in *Computational Linguistics*, vol. 16.1, March 1990.

[Church *et al.*, 1989] K.W. Church, W. Gale, P. Hanks, and D. Hindle. Parsing, Word Associations and Typical Predicate-Argument Relations. In *Proceedings of the International Workshop on Parsing Technologies*, pages 103–112, Carnegie Mellon University, Pittsburgh, PA, 1989. Also appears in Masaru Tomita (ed.), Current Issues in Parsing Technology, pp. 103-112, Kluwer Academic Publishers, Boston, MA, 1991.

[Church *et al.*, 1991] K.W. Church, W. Gale, P. Hanks, and D. Hindle. Using Statistics in Lexical Analysis. In Uri Žernik, editor, *Lexical Acquisition: Using on-line resources to build a lexicon*. Lawrence Erlbaum, 1991. In press.

[Church, 1988] K. Church. Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In *Proceedings of the Second Conference on Applied Natural Language Processing*, Austin, Texas, 1988.

[Debili, 1982] F. Debili. *Analyse Syntactico-Sémantique Fondée sur une Acquisition Automatique de Relations Lexicales Sémantiques*. PhD thesis, Paris XI University, Orsay, France, 1982. Thèse de Doctorat D'état.

[Hindle and Rooth, 1990] D. Hindle and M. Rooth. Structural Ambiguity and Lexical Relations. In *DARPA Speech and Natural Language Workshop*, Hidden Valley, PA, June 1990.

[Hindle, 1983] D. Hindle. User Manual for Fidditch, a Deterministic Parser. Technical Memorandum 7590-142, Naval Research laboratory, 1983.

[Marcus, 1990] M. Marcus. Tutorial on Tagging and Processing Large Textual Corpora. Presented at the 28th annual meeting of the ACL, June 1990.

[Salton, 1989] J. Salton. *Automatic Text Processing, The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley Publishing Company, NY, 1989.

[Smadja and McKeown, 1990] F. Smadja and K. McKeown. Automatically Extracting and Representing Collocations for Language Generation. In *Proceedings of the 28th annual meeting of the ACL*, Pittsburgh, PA, June 1990. Association for Computational Linguistics.

[Smadja, 1988] F. Smadja. Lexical Co-occurrence, The Missing Link in Language Acquisition. In *Program and abstracts of the* 15^{th} *International ALLC, Conference of the Association for Literary and Linguistic Computing*, Jerusalem, Israel, June 1988.

[Smadja, 1991] F. Smadja. *Retrieving Collocational Knowledge from Textual Corpora. An Application: Language Generation*. PhD thesis, Computer Science Department, Columbia University, New York, NY, April 1991.