Retrieving Collocations from Text: Xtract

Frank Smadja*
Columbia University

Natural languages are full of collocations, recurrent combinations of words that co-occur more often than expected by chance and that correspond to arbitrary word usages. Recent work in lexicography indicates that collocations are pervasive in English; apparently, they are common in all types of writing, including both technical and nontechnical genres. Several approaches have been proposed to retrieve various types of collocations from the analysis of large samples of textual data. These techniques automatically produce large numbers of collocations along with statistical figures intended to reflect the relevance of the associations. 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.

In this paper, we describe a set of techniques based on statistical methods for retrieving and identifying collocations from large textual corpora. These techniques produce a wide range of collocations and are based on some original filtering methods that allow the production of richer and higher-precision output. These techniques have been implemented and resulted in a lexicographic tool, **Xtract**. *The techniques are described and some results are presented on a 10 million–word corpus of stock market news reports. A lexicographic evaluation of* **Xtract** *as a collocation retrieval tool has been made, and the estimated precision of* **Xtract** *is 80%.*

1. Introduction

Consider the following sentences:

1. "**The Dow Jones average of 30 industrials** rose 26.28 points to 2,304.69 on Tuesday."
2. "**The Dow average** rose 26.28 points to 2,304.69 on Tuesday."
3. "**The Dow industrials** rose 26.28 points to 2,304.69 on Tuesday."
4. "**The Dow Jones industrial** rose 26.28 points to 2,304.69 on Tuesday."

⋆ 5. "**The Jones industrials** rose 26.28 points to 2,304.69 on Tuesday."

* Computer Science Department, Columbia University, New York, NY 10027. smadja@cs.columbia.edu.

Table 1
Cross linguistic comparisons of collocations.

Language	English	Translation	English correspondence
French	to see the door	voir la porte	to see the door
German	to see the door	die Tür sehen	to see the door
Italian	to see the door	vedere la porta	to see the door
Spanish	to see the door	ver la puerta	to see the door
Turkish	to see the door	kapiyi görmek	to see the door
French	to break down/force the door	enfoncer la porte	⋆ to push the door through
German	to break down/force the door	die Tür aufbrechen	⋆ to break the door
Italian	to break down/force the door	sfondare la porta	⋆ to hit/demolish the door
Spanish	to break down/force the door	tumbar la puerta	⋆ to fall the door
Turkish	to break down/force the door	kapiyi kirmak	⋆ to break the door

⋆ 6. "**The industrial Dow** rose 26.28 points to 2,304.69 on Tuesday."

⋆ 7. "**The Dow of 30 industrials** rose 26.28 points to 2,304.69 on Tuesday."

⋆ 8. "**The Dow industrial** rose 26.28 points to 2,304.69 on Tuesday."

The above sentences contain expressions that are difficult to handle for nonspecialists. For example, among the eight different expressions referring to the famous Wall Street index, only those used in sentences 1–4 are correct. The expressions used in the starred sentences 5–8 are all incorrect. The rules violated in sentences 5–8 are neither rules of syntax nor of semantics but purely lexical rules. The word combinations used in sentences 5–8 are invalid simply because they do not exist; similarly, the ones used in sentences 1–4 are correct because they exist.

Expressions such as these are called collocations. Collocations vary tremendously in the number of words involved, in the syntactic categories of the words, in the syntactic relations between the words, and in how rigidly the individual words are used together. For example, in some cases, the words of a collocation must be adjacent, as in sentences 1–5 above, while in others they can be separated by a varying number of other words. Unfortunately, with few exceptions (e.g., Benson, Benson, and Ilson 1986a) collocations are generally unavailable in compiled form. This creates a problem for persons not familiar with the sublanguage[1] as well as for several machine applications such as language generation.

In this paper we describe a set of techniques for automatically retrieving such collocations from naturally occurring textual corpora. These techniques are based on statistical methods; they have been implemented in a tool, **Xtract**, which is able to retrieve a wide range of collocations with high performance. Preliminary results obtained with parts of **Xtract** have been described in the past (e.g., Smadja and McKeown 1990); this paper gives a complete description of the system and the results obtained.

1 This is true for laymen and also for non-native speakers familiar with the domain but not familiar with the English expressions.

"Our firm **made/did** *a deal with them"*
"The swimmer **had/got** *a cramp"*
"Politicians are always **on/in** *the firing lane"*
"These decisions are to be **made/taken** *rapidly"*
"The children usually **set/lay** *the table"*
"You have to **break in/run in** *your new car"*

Figure 1
British English or American English? from Benson (1990).

sentences	candidates
"If a fire breaks out, the alarm will ?? "	*"ring, go off, sound, start"*
"The boy doesn't know how to ?? his bicycle"	*"drive, ride, conduct"*
"The American congress can ?? a presidential veto"	*"ban/cancel/delete/reject"*
	"turn down/abrogate/overrule"
"Before eating your bag of microwavable popcorn, you have to ?? it"	*"cook/nuke/broil/fry/bake"*

Figure 2
Fill-in-the-blank test, from Benson (1990).

Xtract now works in three stages. In the first stage, pairwise lexical relations are retrieved using only statistical information. This stage is comparable to Church and Hanks (1989) in that it evaluates a certain word association between pairs of words. As in Church and Hanks (1989), the words can appear in any order and they can be separated by an arbitrary number of other words. However, the statistics we use provide more information and allow us to have more precision in our output. The output of this first stage is then passed in parallel to the next two stages. In the second stage, multiple-word combinations and complex expressions are identified. This stage produces output comparable to that of Choueka, Klein, and Neuwitz (1983); however the techniques we use are simpler and only produce relevant data. Finally, by combining parsing and statistical techniques the third stage labels and filters collocations retrieved at stage one. The third stage has been evaluated to raise the precision of **Xtract** from 40% to 80% with a recall of 94%.

Section 2 is an introductory section on collocational knowledge, Section 3 describes the type of collocations that are retrieved by **Xtract**, and Section 4 briefly surveys related efforts and contrasts our work to them. The three stages of **Xtract** are then introduced in Section 5 and described respectively in Sections 6, 7, and 8. Some results obtained by running **Xtract** on several corpora are listed and discussed in Section 9. Qualitative and quantitative evaluations of our methods and of our results are discussed in Sections 10 and 11. Finally, several possible applications and tasks for **Xtract** are discussed in Section 12.

2. What Are Collocations?

There has been a great deal of theoretical and applied work related to collocations that has resulted in different characterizations (e.g., Allerton 1984; Cruse 1986; Mel'čuk 1981). Depending on their interests and points of view, researchers have focused on different aspects of collocations. One of the most comprehensive definition that has

been used can be found in the lexicographic work of Benson and his colleagues (Benson 1990). The definition is the following:

Definition
A collocation is an arbitrary and recurrent word combination (Benson 1990).

This definition, however, does not cover some aspects and properties of collocations that have consequences for a number of machine applications. For example, it has been shown that collocations are difficult to translate across languages—this fact obviously has a direct application for machine translation. Many properties of collocations have been identified in the past; however, the tendency was to focus on a restricted type of collocation. In this section, we present four properties of collocations that we have identified and discuss their relevance to computational linguistics.

2.1 Collocations Are Arbitrary

Collocations are difficult to produce for second language learners (Nakhimovsky and Leed 1979). In most cases, the learner cannot simply translate word-for-word what s/he would say in her/his native language. As we can see in Table 1, the word-for-word translation of *"to open the door"* works well in both directions in all five languages. In contrast, translating word-for-word the expression: *"to break down/force the door"* is a poor strategy in both directions in all five languages. The co-occurrence of *"door"* and *"open"* is an open or free combination, whereas the combination *"door"* and *"break down"* is a collocation. Learners of English would not produce *"to break down a door"* whether their first language is French, German, Italian, Spanish, or Turkish, if they were not aware of the construct.

Figure 1 illustrates disagreements between British English and American English. Here the problem is even finer than in Table 1 since the disagreement is not across two different languages, but across dialects of English. In each of the sentences given in this figure, there is a different word choice for the American (left side) and the British English (right side). The word choices do not correspond to any syntactic or semantic variation of English but rather to different word usages in both dialects of English.

Translating from one language to another requires more than a good knowledge of the syntactic structure and the semantic representation. Because collocations are arbitrary, they must be readily available in both languages for effective machine translation.

2.2 Collocations Are Domain-Dependent

In addition to nontechnical collocations such as the ones presented before, domain-specific collocations are numerous. Technical jargons are often totally unintelligible for the layman. They contain a large number of technical terms. In addition, familiar words seem to be used differently. In the domain of sailing (Dellenbaugh and Dellenbaugh 1990), for example, some words are unknown to the nonfamiliar reader: *rigg*, *jib*, and *leeward* are totally meaningless to the layman. Some other combinations apparently do not contain any technical words, but these words take on a totally different meaning in the domain. For example, *a dry suit* is not a suit that is dry but a special type of suit used by sailors to stay dry in difficult weather conditions. Similarly *a wet suit* is a special kind of suit used for several marine activities. Native speakers are often unaware of the arbitrariness of collocations in nontechnical core English; however, this arbitrariness becomes obvious to the native speaker in specific sublanguages.

type	example
N-Adj	**"heavy/light [] trading/smoker/traffic"**
N-Adj	**"high/low [] fertility/pressure/bounce"**
N-Adj	**"large/small [] crowd/retailer/client"**
SV	**"index [] rose**
SV	**"stock [] [rose, fell, jumped, continued, declined, crashed, ...]"**
SV	**"advancers [] [outnumbered, outpaced, overwhelmed, outstripped]"**
V-Adv	**"trade ⇔ actively," "mix ⇔ narrowly,"**
V-Adv	**"use ⇔ widely," "watch ⇔ closely"**
VO	**"posted [] gain**
VO	**"momentum [] [pick** *up,* **build, carry** *over,* **gather, loose, gain]"**
V-Part	**"take [] from," "raise [] by," "mix [] with"**
VV	**"offer** *to* **[acquire, buy"]**
VV	**"agree** *to* **[acquire, buy"]**

Figure 3
Some examples of predicative collocations.

Linguistically mastering a domain such as the domain of sailing thus requires more than a glossary, it requires knowledge of domain-dependent collocations.

2.3 Collocations Are Recurrent

The recurrent property simply means that these combinations are not exceptions, but rather that they are very often repeated in a given context. Word combinations such as *"to make a decision, to hit a record, to perform an operation"* are typical of the language, and collocations such as *"to buy short," "to ease the jib"* are characteristic of specific domains. Both types are repeatedly used in specific contexts.

2.4 Collocations Are Cohesive Lexical Clusters

By cohesive[2] clusters, we mean that the presence of one or several words of the collocations often implies or suggests the rest of the collocation. This is the property mostly used by lexicographers when compiling collocations (Cowie 1981; Benson 1989a). Lexicographers use other people's linguistic judgment for deciding what is and what is not a collocation. They give questionnaires to people such as the one given in Figure 2. This questionnaire contains sentences used by Benson for compiling collocational knowledge for the BBI (Benson 1989b). Each sentence contains an empty slot that can easily be filled in by native speakers. In contrast, second language speakers would not find the missing words automatically but would consider a long list of words having the appropriate semantic and syntactic features such as the ones given in the second column.

As a consequence, collocations have particular statistical distributions (e.g., Halliday 1966; Cruse 1986). This means that, for example, the probability that any two adjacent words in a sample will be *"red herring"* is considerably larger than the probability of *"red"* times the probability of *"herring."* The words cannot be considered as independent variables. We take advantage of this fact to develop a set of statistical techniques for retrieving and identifying collocations from large textual corpora.

3. Three Types of Collocations

Collocations come in a large variety of forms. The number of words involved as well as the way they are involved can vary a great deal. Some collocations are

2 This notion of cohesion should not be confused with the cohesion as defined by Halliday (Halliday and Hasan 1976). Here we are dealing with a more lexical type of cohesion.

"The NYSE's composite index of all its listed common stocks rose NUMBER to *NUMBER*"*
"On the American Stock Exchange the market value index was up NUMBER at *NUMBER*"*
"The Dow Jones average of 30 industrials fell NUMBER points to *NUMBER*"*
*"The closely watched index had been down about *NUMBER* points in the first hour of trading"*
*"The average finished the week with a net loss of *NUMBER*"*

Figure 4
Some examples of phrasal templates.

very rigid, whereas others are very flexible. For example, a collocation such as the one linking *"to make"* and *"decision"* can appear as *"to make a decision," "decisions to be made," "made an important decision,"* etc. In contrast, a collocation such as *"The New York Stock Exchange"* can only appear under one form; it is a very rigid collocation, a fixed expression. We have identified three types of collocations: *rigid noun phrases, predicative relations,* and *phrasal templates.* We discuss the three types in turn, and give some examples of collocations.

3.1 Predicative Relations

A predicative relation consists of two words repeatedly used together in a similar syntactic relation. These lexical relations are the most flexible type of collocation. They are hard to identify since they often correspond to interrupted word sequences in the corpus. For example, a noun and a verb will form a predicative relation if they are repeatedly used together with the noun as the object of the verb. *"Make-decision"* is a good example of a predicative relation. Similarly, an adjective repeatedly modifying a given noun such as *"hostile-takeover"* also forms a predicative relation. Examples of automatically extracted predicative relations are given in Figure 3.[3] This class of collocations is related to Mel'čuk's lexical functions (Mel'čuk 1981), and Benson's L-type relations (Benson, Benson, and Ilson 1986b).

3.2 Rigid Noun Phrases

Rigid noun phrases involve uninterrupted sequences of words such as *"stock market," "foreign exchange," "New York Stock Exchange," "The Dow Jones average of 30 industrials."* They can include nouns and adjectives as well as closed class words, and are similar to the type of collocations retrieved by Choueka (1988) and Amsler (1989). They are the most rigid type of collocation. Examples of rigid noun phrases are:[4] *"The NYSE's composite index of all its listed common stocks," "The NASDAQ composite index for the over the counter market," "leveraged buyout," "the gross national product," "White House spokesman Marlin Fitzwater."*

In general, rigid noun phrases cannot be broken into smaller fragments without losing their meaning; they are lexical units in and of themselves. Moreover, they often refer to important concepts in a domain, and several rigid noun phrases can be used to express the same concept. In the New York Stock Exchange domain, for example, *"The*

3 In the examples, the "[]" sign represents a gap of zero, one or several words. The "⇔" sign means that the two words can be in any order.
4 All the examples related to the stock market domain have actually been retrieved by **Xtract**.

Dow industrials," "The Dow Jones average of 30 industrial stocks," "the Dow Jones industrial average," and *"The Dow Jones industrials"* represent several ways to express a single concept. As we have seen before, these rigid noun phrases do not seem to follow any simple construction rule, as, for example, the examples given in sentences 6–8 at the beginning of the paper are all incorrect.

3.3 Phrasal Templates

Phrasal templates consist of idiomatic phrases containing one, several, or no empty slots. They are phrase-long collocations. Figure 4 lists some examples of phrasal templates in the stock market domain. In the figure, the empty slots must be filled in by a number (indicated by **NUMBER** in the figure). More generally, phrasal templates specify the parts of speech of the words that can fill the empty slots. Phrasal templates are quite representative of a given domain and are very often repeated in a rigid way in a given sublanguage. In the domain of weather reports, for example, the sentence *"Temperatures indicate previous day's high and overnight low to 8 a.m."* is actually repeated before each weather report.[5]

Unlike rigid noun phrases and predicative relations, phrasal templates are specifically useful for language generation. Because of their slightly idiosyncratic structure, generating them from single words is often a very difficult task for a language generator. As pointed out by Kukich (1983), in general, their usage gives an impression of fluency that could not be equaled with compositional generation alone.

4. Related Work

There has been a recent surge of research interest in corpus-based computational linguistics methods; that is, the study and elaboration of techniques using large real text as a basis. Such techniques have various applications. Speech recognition (Bahl, Jelinek, and Mercer 1983) and text compression (e.g., Bell, Witten, and Cleary 1989; Guazzo 1980) have been of long-standing interest, and some new applications are currently being investigated, such as machine translation (Brown et al. 1988), spelling correction (Mays, Damerau, and Mercer 1990; Church and Gale 1990), parsing (Debili 1982; Hindle and Rooth 1990). As pointed out by Bell, Witten, and Cleary (1989), these applications fall under two research paradigms: statistical approaches and lexical approaches. In the statistical approach, language is modeled as a stochastic process and the corpus is used to estimate probabilities. In this approach, a collocation is simply considered as a sequence of words (or n-gram) among millions of other possible sequences. In contrast, in the lexical approach, a collocation is an element of a dictionary among a few thousand other lexical items. Collocations in the lexicographic meaning are only dealt with in the lexical approach. Aside from the work we present in this paper, most of the work carried out within the lexical approach has been done in computer-assisted lexicography by Choueka, Klein, and Neuwitz (1983) and Church and his colleagues (Church and Hanks 1989). Both works attempted to automatically acquire true collocations from corpora. Our work builds on Choueka's, and has been developed contemporarily to Church's.

Choueka, Klein, and Neuwitz (1983) proposed algorithms to automatically retrieve idiomatic and collocational expressions. A collocation, as defined by Choueka, is a sequence of adjacent words that frequently appear together. In theory the sequences can be of any length, but in actuality, they contain two to six words. In Choueka

5 Taken from the daily reports transmitted daily by The Associated Press newswire.

(1988), experiments performed on an 11 million–word corpus taken from the *New York Times* archives are reported. Thousands of commonly used expressions such as *"fried chicken," "casual sex," "chop suey," "home run,"* and *"Magic Johnson"* were retrieved. Choueka's methodology for handling large corpora can be considered as a first step toward computer-aided lexicography. The work, however, has some limitations. First, by definition, only uninterrupted sequences of words are retrieved; more flexible collocations such as *"make-decision,"* in which the two words can be separated by an arbitrary number of words, are not dealt with. Second, these techniques simply analyze the collocations according to their observed frequency in the corpus; this makes the results too dependent on the size of the corpus. Finally, at a more general level, although disambiguation was originally considered as a performance task, the collocations retrieved have not been used for any specific computational task.

Church and Hanks (1989) describe a different set of techniques to retrieve collocations. A collocation as defined in their work is a pair of correlated words. That is, a collocation is a pair of words that appear together more often than expected. Church et al. (1991) improve over Choueka's work as they retrieve interrupted as well as uninterrupted sequences of words. Also, these collocations have been used by an automatic parser in order to resolve attachment ambiguities (Hindle and Rooth 1990). They use the notion of mutual information as defined in information theory (Shannon 1948; Fano 1961) in a manner similar to what has been used in speech recognition (e.g., Ephraim and Rabiner 1990), or text compression (e.g., Bell, Witten, and Cleary 1989), to evaluate the correlation of common appearances of pairs of words. Their work, however, has some limitations too. First, by definition, it can only retrieve collocations of length two. This limitation is intrinsic to the technique used since mutual information scores are defined for two items. The second limitation is that many collocations identified in Church and Hanks (1989) do not really identify true collocations, but simply pairs of words that frequently appear together such as the pairs *"doctor-nurse," "doctor-bill," "doctor-honorary," "doctors-dentists," "doctors-hospitals," etc.* These co-occurrences are mostly due to semantic reasons. The two words are used in the same context because they are of related meanings; they are not part of a single collocational construct.

The work we describe in the rest of this paper is along the same lines of research. It builds on Choueka's work and attempts to remedy the problems identified above. The techniques we describe retrieve the three types of collocations discussed in Section 2, and they have been implemented in a tool, **Xtract**. **Xtract** retrieves interrupted as well as uninterrupted sequences of words and deals with collocations of arbitrary length (1 to 30 in actuality). The following four sections describe and discuss the techniques used for **Xtract**.

5. Xtract: Introduction

Xtract consists of a set of tools to locate words in context and make statistical observation to identify collocations. In the upgraded version we describe here, **Xtract** has been extended and refined. More information is computed and an effort has been made to extract more functional information. **Xtract** now works in three stages.

The three-stage analysis is described in Sections 6, 7, and 8. In the first stage, described in Section 6, **Xtract** uses straight statistical measures to retrieve from a corpus pairwise lexical relations whose common appearance within a single sentence are correlated. A pair (or bigram) is retrieved if its frequency of occurrence is above a certain threshold and if the words are used in relatively rigid ways. The output of stage one is then passed to both the second and third stage in parallel. In the second

stage, described in Section 7, **Xtract** uses the output bigrams to produce collocations involving more than two words (or n-grams). It analyzes all sentences containing the bigram and the 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 rigid noun phrases in the training corpus. In the third stage, described in Section 8, **Xtract** adds syntactic information to collocations retrieved at the first stage and filters out inappropriate ones. For example, if a bigram involves a noun and a verb, this stage identifies it either as a subject-verb or as a verb-object collocation. If no such consistent relation is observed, then the collocation is rejected.

6. Xtract Stage One: Extracting Significant Bigrams

According to Cruse's definition (Cruse 1986), a syntagmatic lexical relation consists of a pair of words whose common appearances within a single phrase structure are correlated. In other words, those two words appear together within a single syntactic construct more often than expected by chance. The first stage of **Xtract** attempts to identify such pairwise lexical relations and produce statistical information on pairs of words involved together in the corpus.

Ideally, in order to identify lexical relations in a corpus one would need to first parse it to verify that the words are used in a single phrase structure. However, in practice, free-style texts contain a great deal of nonstandard features over which automatic parsers would fail.[6] Fortunately, there is strong lexicographic evidence that most syntagmatic lexical relations relate words separated by at most five other words (Martin, Al, and Van Sterkenburg 1983). In other words, most of the lexical relations involving a word w can be retrieved by examining the neighborhood of w, wherever it occurs, within a span of five (-5 and $+5$ around w) words.[7] In the work presented here, we use this simplification and consider that two words co-occur if they are in a single sentence and if there are fewer than five words between them.

In this first stage, we thus use only statistical methods to identify relevant pairs of words. These techniques are based on the assumptions that if two words are involved in a collocation then:

- the words must appear together significantly more often than expected by chance.
- because of syntactic constraints the words should appear in a relatively rigid way.[8]

These two assumptions are used to analyze the word distributions, and we base our filtering techniques on them.

6.1 Presentation of the Method

In this stage as well as in the two others, we often need part-of-speech information for several purposes. Stochastic part-of-speech taggers such as those in Church (1988) and

6 This fact is being seriously challenged by current research (e.g., Abney 1990; Hindle 1983), and might not be true in the near future.

7 Not crossing sentence boundaries.

8 This is obviously not true for nonconfigurational languages. Although we do believe that the methods described in this paper can be applied to many languages, we have only used them on English texts.

Garside and Leech (1987) have been shown to reach 95–99% performance on free-style text. We preprocessed the corpus with a stochastic part-of-speech tagger developed at Bell Laboratories by Ken Church (Church 1988).[9]

In the rest of this section, we describe the algorithm used for the first stage of **Xtract** in some detail. We assume that the corpus is preprocessed by a part of speech tagger and we note w_i a collocate of w if the two words appear in a common sentence within a distance of 5 words.

Step 1.1: Producing Concordances

Input: The tagged corpus, a given word w.
Output: All the sentences containing w.

Description: This actually encompasses the task of identifying sentence boundaries, and the task of selecting sentences containing w. The first task is not simple and is still an open problem. It is not enough to look for a period followed by a blank space as, for example, abbreviations and acronyms such as S.B.F., U.S.A., and A.T.M. often pose a problem. The basic algorithm for isolating sentences is described and implemented by a finite-state recognizer. Our implementation could easily be improved in many ways. For example, it performs poorly on acronyms and often considers them as end of sentences; giving it a list of currently used acronyms such as *N.B.A.*, *E.I.K.*, etc., would significantly improve its performance.

Step 1.2: Compile and Sort

Input: Output of Step 1.1, i.e., a set of tagged sentences containing w.
Output: A list of words w_i with frequency information on how w and w_i co-occur. This includes the raw frequency as well as the breakdown into frequencies for each possible position. See Table 2 for example outputs.

Description: For each input sentence containing w, we make a note of its collocates and store them along with their position relative to w, their part of speech, and their frequency of appearance. More precisely, for each prospective lexical relation, or for each potential collocate w_i, we maintain a data structure containing this information. The data structure is shown in Figure 5. It contains $freq_i$, the frequency of appearance of w_i with w so far in the corpus, PP, the part of speech of w_i, and p_j^i, $(-5 \leq j \leq 5, j \neq 0)$, the frequency of appearance of w_i with w such that they are j words apart. The p_j^is represent the histogram of the frequency of appearances of w and w_i in given positions. This histogram will be used in later stages.

As an example, if sentence (9) is the current input to step 1.2 and w = *takeover*, then, the prospective lexical relations identified in sentence (9) are as shown in Table 3.

9. *" The pill would make a takeover attempt more expensive by allowing the retailer's shareholders to . . . "*

In Table 3, *distance* is the distance between *"takeover"* and w_i, and PP is the part of speech of w_i. The closed class words are not considered at this stage and the other

9 We are grateful to Ken Church and to Bell Laboratories for providing us with this tool.

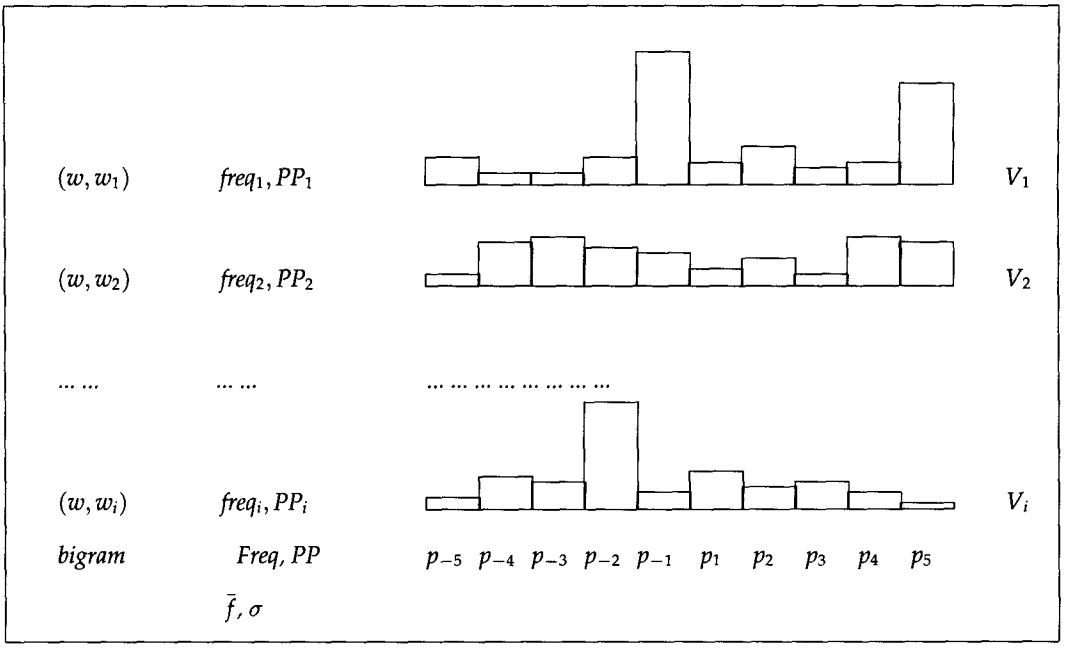

Figure 5
Data structure maintained at stage one by **Xtract**.

words, such as *"shareholders,"* are rejected because they are more than five words away from *"takeover."* For each of the above word pairs, we maintain the associated data structure as indicated in Figure 5. For *takeover pill,* for example, we would increment $freq_{pill}$, and the p_4 column in the histogram. Table 2 shows the output for the adjective collocates of the word *"takeover."*

Step 1.3: Analyze

Input: Output of Step 1.2, i.e., a list of words w_i with information on how often and how w and w_i co-occur. See Table 2 for an example input.
Output: Significant word pairs, along with some statistical information describing how strongly the words are connected and how rigidly they are used together. A separate (but similar) statistical analysis is done for each syntactic category of collocates. See Table 4 for an example output.

Description: At this step, the statistical distribution of the collocates of w is analyzed, and the interesting word pairs are automatically selected. If part of speech information is available, a separate analysis is made depending on the part of speech of the collocates. This balances the fact that verbs, adjectives, and nouns are simply not equally frequent.

For each word w, we first analyze the distribution of the frequencies $freq_i$ of its collocates w_i, and then compute its average frequency $\bar{f}$ and standard deviation σ around $\bar{f}$. We then replace $freq_i$ by its associated z-score k_i. k_i is called the *strength* of the word pair in Figure 4; it represents the number of standard deviation above the

Table 2
Output of stage 1, step 3. Noun–adjective associations.

w	w_i	*Freq*	p_{-5}	p_{-4}	p_{-3}	p_{-2}	p_{-1}	p_1	p_2	p_3	p_4	p_5
takeover	possible	178	0	13	4	23	138	0	0	0	0	0
takeover	corporate	93	2	2	2	1	63	3	2	9	4	5
takeover	unsolicited	83	5	30	5	0	42	0	0	1	0	0
takeover	several	81	2	6	6	6	45	0	0	12	0	4
takeover	recent	76	5	4	6	5	17	0	0	36	2	1
takeover	new	75	4	3	6	28	27	0	1	4	2	0
takeover	unwanted	53	5	0	0	2	46	0	0	0	0	0
takeover	expensive	52	1	0	0	0	2	0	23	23	3	0
takeover	potential	50	1	0	1	3	42	0	0	0	2	1
takeover	big	47	0	0	0	4	15	0	0	5	21	2
takeover	friendly	41	0	3	3	1	25	0	0	2	3	4
takeover	unsuccessful	40	0	1	5	6	27	0	0	0	0	1
takeover	biggest	35	1	2	1	4	20	0	0	0	5	2
takeover	largest	32	0	1	3	20	3	0	0	0	0	5
takeover	old	28	0	8	6	0	14	0	0	0	0	0
takeover	unfriendly	26	0	0	0	0	18	0	0	0	0	8
takeover	rival	26	0	1	3	0	3	0	8	5	5	1
takeover	inadequate	26	5	10	2	0	0	0	0	9	0	0
takeover	initial	25	0	6	0	0	13	0	0	4	0	2
takeover	unwelcome	24	4	0	0	0	20	0	0	0	0	0
takeover	previous	24	0	2	0	4	18	0	0	0	0	0
takeover	federal	22	4	2	2	0	0	0	2	2	8	2
takeover	bitter	22	0	0	0	7	14	0	0	0	1	0
takeover	strong	19	0	4	3	5	4	0	0	1	0	2
takeover	hostile	16	0	6	0	0	10	0	0	0	0	0
takeover	attractive	16	1	0	5	3	7	0	0	0	0	0
takeover	unfair	13	0	0	0	0	13	0	0	0	0	0

Table 3
The collocates of *"takeover"* as retrieved from sentence (9).

w	w_i	*distance*	*PP*
takeover	pill	4	N
takeover	make	2	V
takeover	attempt	-1	N
takeover	expensive	-3	J
takeover	allowing	-5	V

average of the frequency of the word pair w and w_i and is defined as:

$$k_i = \frac{freq_i - \bar{f}}{\sigma} \tag{1a}$$

Then, we analyze the distribution of the p_j^is and produce their average $\bar{p}_i$ and variance U_i around $\bar{p}_i$. In Figure 4 *spread* represents U_i on a scale of 1 to 100. U_i characterizes the shape of the p_i^j histogram. If U_i is small, then the histogram will tend to be flat,

which means that w_i can be used equivalently in almost any position around w. In contrast, if U_i is large, then the histogram will tend to have peaks, which means that w_i can only be used in one (or several) specific position around w. U_i is defined by:

$$U_i = \frac{\sum_{j=1}^{10} (p_i^j - \bar{p}_i)^2}{10} \tag{1b}$$

These analyses are then used to sort out the retrieved data. First, using (1a), collocates with *strength* smaller than a given threshold k_0 are eliminated. Then, using (1b), we filter out the collocates having a variance U_i smaller than a given threshold U_0. Finally, we keep the interesting collocates by pulling out the peaks of the $p^i{}_j$ distributions. These peaks correspond to the js such that the z-score of $p^i{}_j$ is bigger than a given threshold k_1. These thresholds have to be determined by the experimenter and are dependent on the use of the retrieved collocations. As described in Smadja (1991), for language generation we found that $(k_0, k_1, U_0) = (1, 1, 10)$ gave good results, but for other tasks different thresholds might be preferable. In general, the lower the threshold the more data are accepted, the higher the recall, and the lower the precision of the results. Section 10 describes an evaluation of the results produced with the above thresholds.

More formally, a peak, or lexical relation containing w, at this point is defined as a tuple $(w_i, distance, strength, spread, j)$ verifying the following set of inequalities:

$$(C) \quad \left\{ \begin{array}{ll} strength = \frac{freq - \bar{f}}{\sigma} \geq k_0 & (C_1) \\ spread \geq U_0 & (C_2) \\ p_j^i \geq \bar{p}_i + (k_1 \times \sqrt{U_i}) & (C_3) \end{array} \right\}$$

Some example results are given in Table 4.

As shown in Smadja (1991), the whole first stage of **Xtract** as described above can be performed in $O(S \log S)$ time, in which S is the size of the corpus. The third step of counting frequencies and maintaining the data structure dominates the whole process and as pointed out by Ken Church (personal communication), it can be reduced to a sorting problem.

6.2 What Exactly Is Filtered Out?

The inequality set (C) is used to filter out irrelevant data, that is pairs of words supposedly not used consistently within a single syntactic structure. This section discusses the importance of each inequality in (C) on the filtering process.

$$strength = \frac{freq - \bar{f}}{\sigma} \geq k_0 \tag{C_1}$$

Condition (C_1) helps eliminate the collocates that are not frequent enough. This condition specifies that the frequency of appearance of w_i in the neighborhood of w must be at least one standard deviation above the average. In most statistical distributions, this thresholding eliminates the vast majority of the lexical relations. For example, for $w =$ *"takeover,"* among the 3385 possible collocates only 167 were selected, which gives a proportion of 95% rejected. In the case of the standard normal distribution, this would reject some 68% of the cases. This indicates that the actual distribution of the

Table 4
Output of stage 1, step 4.

w_i	w_j	*distance*	*strength*	*spread*
hostile	takeovers	1	13	97
hostile	takeover	1	13	90
corporate	takeovers	1	8	90
possible	takeover	1	6	73
hostile	takeovers	2	2	70
corporate	takeover	1	3	63
unwanted	takeover	1	1	83
potential	takeover	1	1	80
several	takeover	1	2	50
unsolicited	takeover	1	2	53
his	takeover	1	3	44
unsuccessful	takeover	1	1	63
takeover	recent	3	2	46
unsolicited	takeover	4	2	53
takeover	last	2	2	46
friendly	takeover	1	1	60
takeover	expensive	3	1	60
takeover	expensive	2	1	60
new	takeover	2	2	46
new	takeover	1	2	46
takeover	big	4	1	47
takeovers	other	2	1	43
big	takeover	1	1	46
takeovers	major	4	1	46
biggest	takeover	1	.93	53
largest	takeover	2	.82	60

collocates of *"takeover"* has a large kurtosis.[10] Among the eliminated collocates were *"dormant, dilute, ex., defunct,"* which obviously are not typical of a takeover. Although these rejected collocations might be useful for applications such as speech recognition, for example, we do not consider them any further here. We are looking for recurrent combinations and not casual ones.

$$spread \geq U_0 \qquad (C_2)$$

Condition (C_2) requires that the histogram of the 10 relative frequencies of appearance of w_i within five words of w (or p_j^is) have at least one spike. If the histogram is flat, it will be rejected by this condition. For example, in Figure 5, the histogram associated with w_2 would be rejected, whereas the one associated with w_1 or w_i would be accepted. In Table 2, the histogram for *"takeover-possible"* is clearly accepted (there is a spike for p_{-1}), whereas the one for *"takeover-federal"* is rejected. The assumption here is that, if the two words are repeatedly used together within a single syntactic construct, then they will have a marked pattern of co-appearance, i.e., they will not appear in all the possible positions with an equal probability. This actually eliminates pairs such as *"telephone-television," "bomb-soldier," "trouble-problem," "big-small,"* and

10 The kurtosis of the distribution of the collocates probably depends on the word, and there is currently no agreement on the type of distribution that would describe them.

"doctor-nurse" where the two words co-occur with no real structural consistency. The two words are often used together because they are associated with the same context rather than for pure structural reasons. Many collocations retrieved in Church and Hanks (1989) were of this type, as they retrieved *doctors-dentists, doctors-nurses, doctor-bills, doctors-hospitals, nurses-doctor, etc.*, which are not collocations in the sense defined above. Such collocations are not of interest for our purpose, although they could be useful for disambiguation or other semantic purposes. Condition (C_2) filters out exactly this type of collocations.

$$p_j^i \geq \bar{p}_i + (k_1 \times \sqrt{U_i}) \qquad (C_3)$$

Condition (C_3) pulls out the interesting relative positions of the two words. Conditions (C_2) and (C_1) eliminate rows in the output of Step 1.2. (See Figure 2). In contrast, Condition (C_3) selects columns from the remaining rows. For each pair of words, one or several positions might be favored and thus result in several p_j^i selected. For example, the pair *"expensive-takeover"* produced two different peaks, one with only one word in between *"expensive"* and *"takeover,"* and the other with two words. Example sentences containing the two words in the two possible positions are:

> *" The provision is aimed at making a hostile* **takeover** *prohibitively* **expensive** *by enabling Borg Warner's stockholders to buy the . . . "*
>
> *"The pill would make a* **takeover** *attempt more* **expensive** *by allowing the retailer's shareholders to buy more company stock . . . "*

Let us note that this filtering method is an original contribution of our work. Other works such as Church and Hanks (1989) simply focus on an evaluation of the correlation of appearance of a pair of words, which is roughly equivalent to condition (C_1). (See next section). However, taking note of their pattern of appearance allows us to filter out more irrelevant collocations with (C_2) and (C_3). This is a very important point that will allow us to filter out many invalid collocations and also produce more functional information at stages 2 and 3. A graphical interpretation of the filtering method used for **Xtract** is given in Smadja (1991).

7. Xtract Stage Two: From 2-Grams to N-Grams

The role of the second stage of **Xtract** is twofold. It produces collocations involving more than two words, and it filters out some pairwise relations. Stage 2 is related to the work of Choueka (1988), and to some extent to what has been done in speech recognition (e.g., Bahl, Jelinek, and Mercer 1983; Merialdo 1987; Ephraim and Rabiner 1990).

7.1 Presentation of the Method

In this second stage, **Xtract** uses the same components used for the first stage but in a different way. It starts with the pairwise lexical relations produced in stage 1 and produces multiple word collocations, such as rigid noun phrases or phrasal templates, from them. To do this, **Xtract** studies the lexical relations in context, which is exactly what lexicographers do. For each bigram identified at the previous stage, **Xtract** examines all instances of appearance of the two words and analyzes the distributions of words and parts of speech in the surrounding positions.

Input: Output of Stage 1. Similar to Table 4, i.e., a list of bigrams with their statistical information as computed in stage 1.

Output: Sequences of words and parts of speech. See Figure 8.

Stage 2 has three steps:

Step 2.1: Produce Concordances

Identical to Stage 1, Step 1.1. Given a pair of words w and w_i, and an integer specifying the distance of the two words.[11] This step produces all the sentences containing them in the given position. For example, given the bigram *takeover-thwart* and the distance 2, this step produces sentences like:

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

Step 2.2: Compile and Sort

Identical to Stage 1, Step 1.2. We compute the frequency of appearance of each of the collocates of w by maintaining a data structure similar to the one given in Figure 5.

Step 2.3: Analyze and Filter

Input: Output of Step 2.2.
Output: N-grams such as in Figure 8.

Discussion: Here, the analyses are simpler than for Stage 1. We are only interested in percentage frequencies and we only compute the moment of order 1 of the frequency distributions.

Tables produced in Step 2.2 (such as in Figure 5) are used to compute the frequency of appearance of each word in each position around w. For each of the possible relative distances from w, we analyze the distribution of the words and only keep the words occupying the position with a probability greater than a given threshold T.[12] If part of speech information is available, the same analysis is also performed with parts of speech instead of actual words. In short, a word w or a part of speech *pos* is kept in the final n-gram at position i if and only if it satisfies the following inequation:

$$p(word[i] = w_0) > T \tag{4a}$$

$p(e)$ denotes the probability of event e. Consider the examples given in Figures 6 and 7 that show the concordances (output of step 2.1) for the input pairs: *"average-industrial"* and *"index-composite."*

In Figure 6, the same words are always used from position −4 to position 0. However, at position +1, the words used are always different. *"Dow"* is used at position −3 in more than 90% of the cases. It is thus part of the produced rigid noun phrases. But *"down"* is only used a couple of times (out of several hundred) at position +1,

11 The distance is actually optional and can be given in various ways. We can specify the word order, the maximum distance, the exact distance, etc.

12 This threshold must also be determined by the experimenter. In the following we use $T = 0.75$. As discussed previously, the choice of the threshold is arbitrary, and the general rule is that the lower the threshold, the higher the recall and the lower the precision of the results. The choice of 0.75 is based on the manual observations of several samples and it has effected the overall results, as discussed in Section 10.

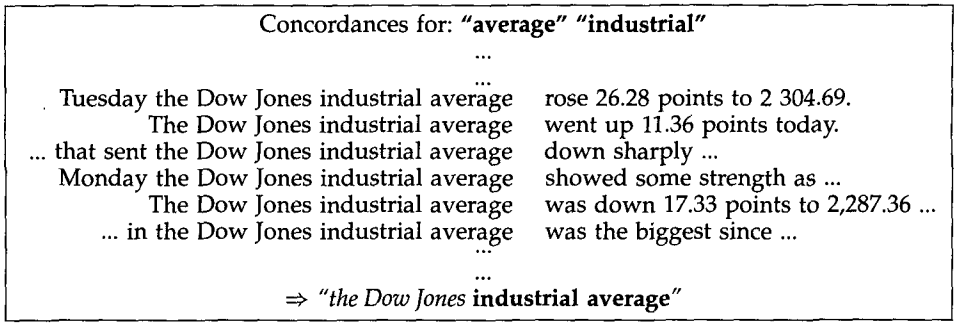

Figure 6
Producing: *"the Dow Jones* **industrial average***"*

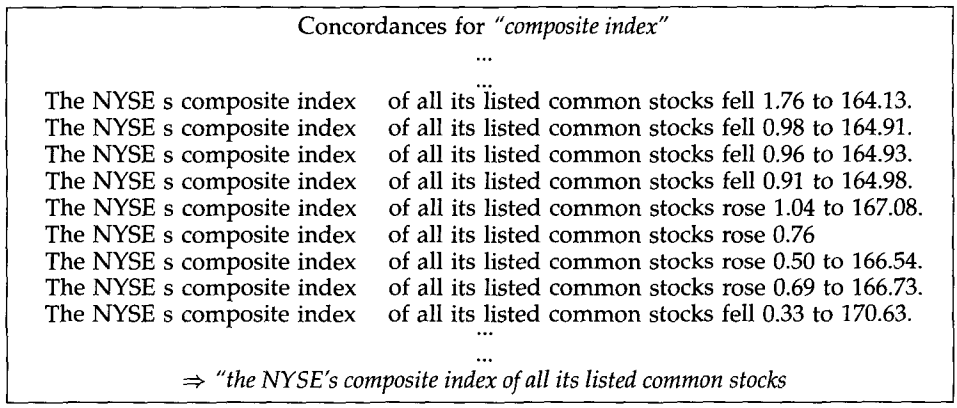

Figure 7
Producing: *"the NYSE's composite index of all its listed common stocks"*

and will not be part of the produced rigid noun phrases. From those concordances, **Xtract** produced the five-word rigid noun phrases: *"The Dow Jones Industrial Average."*

Figure 7 shows that from position −3 to position +7 the words used are always the same. In all the example sentences in which *"composite"* and *"index"* are adjacent, the two words are used within a bigger construct of 11 words (also called an 11-gram). However, if we look at position +8 for example, we see that although the words used are different, in all the cases they are verbs. Thus, after the 11-gram we expect to find a verb. In short, Figure 7 helps us produce both the rigid noun phrases *"The NYSE's composite index of all its listed common stocks,"* as well as the phrasal template *"The NYSE's composite index of all its listed common stocks *VERB* *NUMBER* to *NUMBER*."*

Figure 8 shows some sample phrasal templates and rigid noun phrases that were produced at this stage. The leftmost column gives the input lexical relations. Some other examples are given in Figure 3.

7.2 Discussion

The role of stage 2 is to filter out many lexical relations and replace them by valid ones. It produces both phrasal templates and rigid noun phrases. For example, associations such as *"blue-stocks," "air-controller,"* or *"advancing-market"* were filtered out

lexical relation	collocation
composite-index	*"The NYSE's composite index of all its listed common stocks fell *NUMBER* to *NUMBER*"*
composite-index	*"the NYSE's composite index of all its listed common stocks rose *NUMBER* to *NUMBER*."*
"close-industrial"	*"Five minutes before the close the Dow Jones average of 30 industrials was up/down *NUMBER* to/from *NUMBER*"*
"average industrial"	*"the Dow Jones industrial average."*
"advancing-market"	*"the broader market in the NYSE advancing issues"*
"block-trading"	*"Jack Baker head of block trading in Shearson Lehman Brothers Inc."*
"blue-stocks"	*"blue chip stocks"*
"cable-television"	*"cable television"*
"consumer index"	*"The consumer price index"*

Figure 8
Example output collocations of stage two.

and respectively replaced by: *"blue chip stocks," "air traffic controllers,"* and *"the broader market in the NYSE advancing issues."*

Thus stage 2 produces n-word collocations from two-word associations. Producing n-word collocations has already been done (e.g., Choueka 1988).[13] The general method used by Choueka is the following: for each length n, ($1 \leq n \leq 6$), produce all the word sequences of length n and sort them by frequency. On a 12 million–word corpus, Choueka retrieved 10 collocations of length six, 115 collocations of length five, 1,024 collocations of length four, 4,777 of length three, and some 15,973 of length two. The threshold imposed was 14. The method we presented in this section has three main advantages when compared to a straight n-gram method like Choueka's.

1. Stage 2 retrieves phrasal templates in addition to simple rigid noun phrases. Using part of speech information, we allow categories and words in our templates, thus retrieving a more flexible type of collocation. It is not clear how simple n-gram techniques could be adapted to obtain the same results.

2. Stage 2 gets rid of subsumed m-grams of a given n-gram ($m < n$). Since stage 2 works from bigrams, and produces the biggest n-gram containing it, there is no m-gram ($m < n$) produced that is subsumed by it. For example, although *"shipments of arms to Iran"* is a collocation of length five, *"arms to Iran"* is not an interesting collocation. It is not opaque, and does not constitute a modifier–modified syntactic relation. A straight n-gram method would retrieve both, as well as many other subsumed m-grams, such as *"of arms to Iran."* A sophisticated filtering method would then be necessary to eliminate the invalid ones (See Choueka 1988). Our method avoids this problem and only produces the biggest possible n-gram, namely: *"shipment of arms to Iran."*

3. Stage 2 is a simple way of compiling n-gram data. Retrieving an 11-gram by the methods used in speech, for example, would require a great deal

13 Similar approaches have been done for several applications such as Bahl, Jelinek, and Mercer (1983) and Cerf-Danon et al. (1989) for speech recognition, and Morris and Cherry (1975), Angell (1983), Kukich (1990), and Mays, Damerau, and Mercer (1990) for spelling correction (with letters instead of words).

of CPU time and space. In a 10 million–word corpus, with about 60,000 different words, there are about 3.6×10^9 possible bigrams, 2.16×10^{14} trigrams, and 3×10^{33} 7-grams. This rapidly gets out of hand. Choueka, for example, had to stop at length six. In contrast, the rigid noun phrases we retrieve are of arbitrary length and are retrieved very easily and in one pass. The method we use starts from bigrams and produces the biggest possible subsuming n-gram. It is based on the fact that if an n-gram is statistically significant, then the included bigrams must also be significant. For example, to identify *"The Dow Jones average of 30 industrials,"* a traditional n-gram method would compare it to the other 7-grams and determine that it is significant. In contrast, we start from an included significant bigram (for example, *"Dow-30"*) and we directly retrieve the surrounding n-grams.[14]

8. Xtract Stage Three: Adding Syntax to the Collocations

The collocations as produced in the previous stages are already useful for lexicography. For computational use, however, functional information is needed. For example, the collocations should have some syntactic properties. It is not enough to say that *"make"* goes with *"decision"*; we need to know that *"decision"* is used as the direct object of the verb.

The advent of robust parsers such as **Cass** (Abney 1990) and **Fidditch** (Hindle 1983) has made it possible to process large text corpora with good performance and thus combine statistical techniques with more symbolic analysis. In the past, some similar attempts have been done. Debili (1982) parsed corpora of French texts to identify nonambiguous predicate argument relations. He then used these relations for disambiguation. Hindle and Rooth (1990) later refined this approach by using bigram statistics to enhance the task of prepositional phrase attachment. Church et al. (1989, 1991) have yet another approach; they consider questions such as *what does a boat typically do?* They are preprocessing a corpus with the **Fidditch** parser (Hindle 1983) in order to produce a list of verbs that are most likely associated with the subject *"boat."*

Our goal here is different, as we analyze collocations automatically produced by the first stage of **Xtract** to either add syntactic information or reject them. For example, if a lexical relation identified at stage 1 involves a noun and a verb, the role of stage 3 is to determine whether it is a *subject–verb* or a *verb–object* collocation. If no such consistent relation is observed, then the collocation is rejected. Stage 3 uses a parser but it does not require a complete parse tree. Given a number of sentences, **Xtract** only needs to know pairwise syntactic (modifier–modified) relations. The parser we used in the experiment reported here is **Cass** (Abney 1989, 1990), a bottom-up incremental parser. **Cass**[15] takes input sentences labeled with part of speech and attempts to identify syntactic structure. One of the subtasks performed by **Cass** is to identify predicate argument relations, and this is the task we are interested in here. Stage 3 works in the following three steps.

14 Actually, this 7-gram could be retrieved several times, one for each pair of open class word it contains. But a simple sorting algorithm gets rid of such repetitions.

15 The parser developed at Bell Communication Research by Steve Abney, **Cass** stands for Cascaded Analysis of Syntactic Structure. We are grateful to Steve for helping us with the use of **Cass** and customizing its output for us.

label	bigram	label	bigram
VO	faced test	VO	awaited signs
SV	investors awaited	SV	Street faced
NN	year market	NN	week selloff
NN	stock traders	NN	bull market
JN	old market	JN	major test
JN	last selloff	JN	epic selloff

Figure 9
All the syntactic labels produced by **Cass** on sentence (10).

Step 3.1: Produce Tagged Concordances

Identical to what we did at Stage 2, Step 2.1. Given a pair of words w and w_i, a distance of the two words (optional), and a tagged corpus, **Xtract** produces all the (tagged) sentences containing them in the given position specified by the distance.

Step 3.2: Parse

Input: Output of Step 3.1. A set of tagged sentences each containing both w and w_i.
Output: For each sentence, a set of syntactic labels such as those shown in Figure 9.

Discussion: Cass is called on the concordances. From **Cass** output, we only retrieve binary syntactic relations (or labels) such as *"verb–object"* or *"verb–subject," "noun–adjective,"* and *"noun–noun."* To simplify, we abbreviate them respectively: VO, SV, NJ, NN. For sentence (10) below, for example, the labels produced are shown in Figure 9.

10. *"Wall Street faced a major test with stock traders returning to action for the first time since last week's epic selloff and investors awaited signs of life from the 5-year-old bull market."*

Step 3.3: Label Sentences

Input: A set of sentences each associated with a set of labels as shown in Figure 9.
Output: Collocations with associated syntactic labels as shown in Figure 10.

Discussion: For any given sentence containing both w and w_i, two cases are possible: either there is a label for the bigram (w, w_i), or there is none. For example, for sentence (10), there is a syntactic label for the bigram *faced-test*, but there is none for the bigram *stock-returning*. *Faced-test* enters into a verb object relation, and *stock-returning* does not enter into any type of relation. If no label is retrieved for the bigram, it means that the parser could not identify a relation between the two words. In this case we introduce a new label: U (for undefined) to label the bigram. At this point, we associate with the sentence the label for the bigram (w, w_i). With each of the input sentences, we associate a label for the bigram (w, w_i). For example, the label associated with sentence (10) for the bigram *faced-test* would be VO. A list of labeled sentences for the bigram $w =$ *"rose"* and $w_i =$ *"prices"* is shown in Figure 10.

Some Concordances for **(rose, prices)**	label
... when they **rose** pork **prices** 1.1 percent ...	**VO**
Analysts said stock **prices rose** because of a rally in Treasury bonds.	**SV**
Bond **prices rose** because many traders took the report as a signal ...	**SV**
Stock **prices rose** in moderate trading today with little news ...	**SV**
Bond **prices rose** in quiet trading	**SV**
Stock **prices rose** sharply Friday in response to a rally in ...	**SV**
... soft drink **prices rose** 0.5 percent ...	**SV**
Stock **prices rose** broadly in early trading today as a rising dollar ...	**SV**

Figure 10
Producing the "prices [] rose," **SV** predicative relation at stage 3.

Step 3.4: Filter and Label Collocation

Input: A set of sentences containing w and w_i each associated with a label as shown in Figure 10.
Output: Labeled collocations as shown in Figure 11.

Discussion on Step 3.4: At this step, we count the frequencies of each possible label identified for the bigram (w, w_i) and perform a statistical analysis of order two for this distribution. We compute the average frequency for the distribution of labels: $\bar{f}_t$ and the standard deviation σ_t. We finally apply a filtering method similar to (C_2). Let t be a possible label. We keep t if and only if it satisfies inequality (4b) similar to (4a) given before:

$$p(label[i] = t) > T \tag{4b}$$

A collocation is thus accepted if and only if it has a label g satisfying inequality (4b), and $g \neq U$. Similarly, a collocation is rejected if no label satisfies inequality (4b) or if U satisfies it.

Figure 10 shows part of the output of Step 3.3 for $w =$ *"rose"* and $w_i =$ *"prices."* As shown in the figure, **SV** labels are a large majority. Thus, we would label the relation price-rose as an **SV** relation. An example output of this stage is given in Figure 11. The bigrams labeled U were rejected at this stage.

Stage 3 thus produces very useful results. It filters out collocations and rejects more than half of them, thus improving the quality of the results. It also labels the collocations it accepts, thus producing a more functional and usable type of knowledge. For example, if the first two stages of **Xtract** produce the collocation *"make-decision,"* the third stage identifies it as a verb–object collocation. If no such relation can be observed, then the collocation is rejected. The produced collocations are not simple word associations but complex syntactic structures. Labeling and filtering are two useful tasks for automatic use of collocations as well as for lexicography. The whole of stage 3 (both as a filter and as a labeler) is an original contribution of our work. Retrieving syntactically labeled collocations is a relatively new concern. Moreover, filtering greatly improves the quality of the results. This is also a possible use of the emerging new parsing technology.

8.1 Xtract: The Toolkit

Xtract is actually a library of tools implemented using standard C-Unix libraries. The toolkit has several utilities useful for analyzing corpora. Without making any effort

w	w_i	label	w	w_i	label
savings	ailing	U	securities	dealer	U
savings	appears	U	denominated	securities	VO
savings	continue	U	securities	firms	NN
savings	dip	U	securities	fixed	U
savings	dipped	U	securities	fraud	NN
savings	failing	U	securities	industry	NN
savings	fell	SV	securities	law	NN
manufacturing	sector	NN	securities	lawmakers	U
manufacturing	sector	NN	securities	lawyer	NN
securities	business	NN	securities	lawyers	NN

Figure 11
Some examples of syntactically labeled bigrams.

to make **Xtract** efficient in terms of computing resources, the first stage as well as the second stage of **Xtract** only takes a few minutes to run on a ten-megabyte (pre-tagged) corpus. **Xtract** is currently being used at Columbia University for various lexical tasks. And it has been tested on many corpora, among them: several ten-megabyte corpora of news stories, a corpus, consisting of some twenty megabytes of *New York Times* articles, which has already been used by Choueka (1988), the Brown corpus (Francis and Kučera 1982), a corpus of the proceedings of the Canadian Parliament, also called the Hansards corpus, which amounts to several hundred megabytes. We are currently working on packaging **Xtract** to make it available to the research community. The packaged version will be portable, reusable, and faster than the one we used to write this paper.[16]

We evaluate the filtering power of stage 3 in the evaluation section, Section 10. Section 9 presents some results that we obtained with the three stages of **Xtract**.

9. Some Results

Results obtained from *The Jerusalem Post* corpus have already been reported (e.g., Smadja 1991). Figure 12 gives some results for the three-stage process of **Xtract** on a 10 million-word corpus of stock market reports taken from the Associated Press newswire. The collocations are given in the following format. The first line contains the bigrams with the distance, so that *"sales fell −1"* says that the two words under consideration are *"sales"* and *"fell,"* and that the distance we are considering is −1. The first line is thus the output of stage 1. The second line gives the output of stage 2, i.e., the n-grams. For example, *"takeover-thwart"* is retrieved as *"44 . . . . . to thwart AT takeover NN . . . . . "* AT stands for article, NN stands for nouns, and 44 is the number of times this collocation has been retrieved in the corpus. The third line gives the retrieved tags for this collocation, so that the syntactic relation between *"takeover"* and *"thwart"* is an **SV** relation. And finally, the last line is an example sentence containing the collocation. Output of the type of Figure 12 is automatically produced. This kind of output is about as far as we have gone automatically. Any further analysis and/or use of the collocations would probably require some manual intervention.

16 Please contact the author if you are interested in getting a copy of the software.

```
sales fell -1
158 . . . . . sales fell . . . . 158
TAG: SV
3 4
New home sales fell 2.7 percent in February following an 8.6 percent drop in January
the Commerce Department reported.

study said -1
40 . . . . . AT study said . . . . . 40
TAG: SV
5 6
A private study said Americans are eating about the same amount of red meat they did
four years ago.

sense makes 1
26 . . . . . makes sense . . . . 26
TAG: VO
20 19
Murray Drabkin of Washington lawyer for the Dalkon Shield claimants committee said
now that Robins has agreed it makes sense to sell the company we are finally down
to the real questions How much will the company bring in the open market and
how much of that amount will the claimants
allow to go to shareholders?

steps take 1
75 . . . . . take steps TO VB . . . . . 75
TAG: VO
15 14
Officials also are hopeful that individual nations particularly West Germany and Japan will
take steps to stimulate their own economies.

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

telephone return 1
53 . . . . . return telephone calls . . . . . 53
22 21
TAG: VO
Mesa did not indicate the average price it paid for its 4.4 percent stake and Mesa
officials did not immediately return telephone calls seeking comment.
```

Figure 12
Some complete output on the stock market corpus.

For the 10 million–word stock market corpus, there were some 60,000 different word forms. **Xtract** has been able to retrieve some 15,000 collocations in total. We would like to note, however, that **Xtract** has only been effective at retrieving collocations for words appearing at least several dozen times in the corpus. This means that low-frequency words were not productive in terms of collocations. Out of the 60,000 words in the corpus, only 8,000 were repeated more than 50 times. This means that for a target

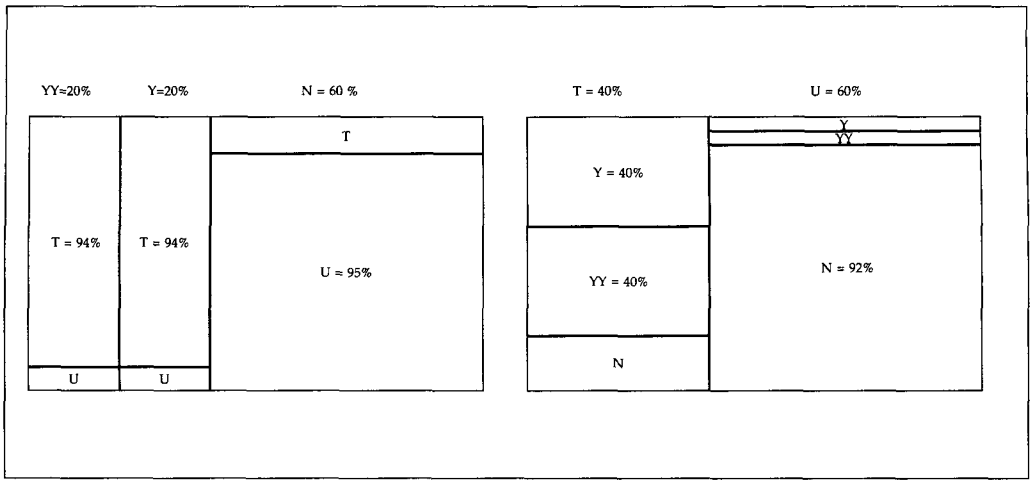

Figure 13
Overlap of the manual and automatic evaluations

lexicon of size $N = 8,000$, one should expect at least as many collocations to be added, and **Xtract** can help retrieve most of them.

10. A Lexicographic Evaluation

The third stage of **Xtract** can thus be considered as a retrieval system that retrieves valid collocations from a set of candidates. This section describes an evaluation experiment of the third stage of **Xtract** as a retrieval system as well as an evaluation of the overall output of **Xtract**. 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**.

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 the *Oxford English Dictionary* (OED) coordinating the North American Readers program of OED at Bell Communication Research. Jeffery Triggs agreed to go over manually several thousands of collocations.[17]

In order to have an unbiased experiment we had to be able to evaluate the performance of **Xtract** against a human expert. We had to have the lexicographer and **Xtract** perform the same task. To do this in an unbiased way we randomly selected a subset of about 4,000 collocations after the first two stages of **Xtract**. This set of collocations thus contained some good collocations and some bad ones. This data set was then evaluated by the lexicographer and the third stage of **Xtract**. This allowed

17 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.

us to evaluate the performances of the third stage of **Xtract** and the overall quality of the total output of **Xtract** in a single experiment. The experiment was as follows:

We gave the 4,000 collocations to evaluate to the lexicographer, asking him to select the ones that he 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** include good collocations, and **N** includes 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 compiling new entries 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, fewer than 400 are serious candidates 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 8, there are several valid labels (VO, VS, NN, etc.). In this experiment, we grouped them under a single label: T. There is only one nonvalid 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 13 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 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% and recall = 94%.

NYT	d	w	DJ	d	w	AP	d	w
pay	2	568	closing	1	4615	gouging	−1	1713
rises	−1	568	rose	−1	3704	get	3	551
raise	2	527	fell	−1	3161	kindle	4	422
cutting	−1	522	tumbled	−1	865	increases	−1	357
declines	−1	492	moved	−1	850	pay	2	335
freeze	−1	481	declined	−1	811	sell	5	293
offered	1	443	finished	−3	710	finished	−3	293
increases	−1	338	closed	−1	648	declining	2	293
closing	1	231	measures	1	644	rose	−5	291
fell	2	224	edged	−1	620	trading	3	207

Figure 14
Top associations with *"price"* in NYT, DJ, and AP.

11. Influence of the Corpus on the Results

In this section, we discuss the extent to which the results are dependent on the corpus used. To illustrate our purpose here, we are using results collected from three different corpora. The first one, DJ, for Dow Jones, is the corpus we used in this paper; it contains (mostly) stock market stories taken from the Associated Press newswire. DJ contains 8–9 million words. The second corpus, NYT, contains articles published in the *New York Times* during the years 1987 and 1988. The articles are on various subjects. This is the same corpus that was used by Choueka (1988). NYT contains 12 million words. The third corpus, AP, contains stories from the Associated Press newswire on various domains such as weather reports, politics, health, finances, etc. AP is 4 million words. Figure 14 represents the top 10 word associations retrieved by **Xtract** stage 1 for the three corpora with the word *"price."* In this figure, d represents the distance between the two words and w represents the weight associated with the bigram. The weight is a combined index of the statistical distribution as discussed in Section 6, and it evaluates the collocation. There are several differences and similarities among the three columns of the figure in terms of the words retrieved, the order of the words retrieved, and the values of w. We identified two main ways in which the results depend on the corpus. We discuss them in turn.

11.1 Results Are Dependent on the Size of the Corpus

From the different corpora we used, we noticed that our statistical methods were not effective for low-frequency words. More precisely, the statistical methods we use do not seem to be effective on low frequency words (fewer than 100 occurrences). If the word is not frequently used in the corpus or if the corpus is too small, then the distribution of its collocates will not be big enough. For example, from AP, which contains about 1,000 occurrences of the word *"rain,"* **Xtract** produced over 170 collocations at stage 1 involving it. In contrast, DJ only contains some 50 occurrences of *"rain"*[18] and **Xtract** could only produce a few collocations with it. Some collocations with *"rain"* and *"hurricane"* extracted from AP are listed in Figure 15. Both words are high-frequency words in AP and low-frequency words in DJ.

18 The corpus actually contains some stories not related to Wall Street.

In short, to build a lexicon for a computational linguistics application in a given domain, one should make sure that the important words in the domain are frequent enough in the corpus. For a subdomain of the stock market describing only the fluctuations of several indexes and some of the major events of the day at Wall Street, a corpus of 10 million words appeared to be sufficient. This 10 million–token corpus contains only 5,000 words each repeated more than 100 times.

11.2 Results Are Dependent on the Contents of the Corpus

Size and frequency are not the only important criteria. For example, even though *"food"* is a high-frequency word in DJ, *"eat"* is not among its collocates, whereas it is among the top ones in the two other corpora. Food is not eaten at Wall Street but rather traded, sold, offered, bought, etc. If the corpus only contains stories in a given domain, most of the collocations retrieved will also be dependent on this domain. We have seen in Section 2 that in addition to jargonistic words, there are a number of more familiar terms that form collocations when used in different domains. A corpus containing stock market stories is obviously not a good choice for retrieving collocations related to weather reports or for retrieving domain independent collocations such as *"make-decision."*

For a domain-specific application, domain-dependent collocations are of interest, and a domain-specific corpus is exactly what is required. To build a system that generates stock market reports, it is a good choice to use a corpus containing only stock market reports.

There is a danger in choosing a too specific corpus however. For example, in Figure 14, we see that the first collocate of *"price"* in AP is *"gouging,"* which is not retrieved in either DJ or in NYT. *"Price gouging"* is not a current practice at Wall Street and this collocation could not be retrieved even on some 20,000 occurrences of the word. An example use of *"price gouging"* is the following:

> *"The Charleston City Council passed an emergency ordinance barring price gouging later Saturday after learning of an incident in which 5 pound bags of ice were being sold for 10."*

More formally, if we compare the columns in Figure 14, we see that the numbers are much higher for DJ than for the other two corpora. This is not due to a size/frequency factor, since *"price"* occurs about 10,000 times in both NYT and DJ, whereas it only occurs 4,500 times in AP. It rather says that the distribution of collocates around *"price"* has a much higher variance in DJ than in the other corpora. DJ has much bigger *weights* because it is focused; the stories are almost all about Wall Street. In contrast, NYT contains a large number of stories with *"price,"* but they have various origins. *"Price"* has 4,627 collocates in NYT, whereas it only has 2,830 in DJ.

Let us call θ_{corpus} the *variety* of a given corpus. One way to measure the variety is to use the information theory measure of entropy for a given language model. Entropy is defined (Shannon 1948) as:

$$\theta_{corpus} = -\sum_{w} p(w) \log p(w)$$

where $p(w)$ is the probability of appearance of a given word, w. Entropy measures the *predictability* of a corpus, in other words, the bigger the entropy of a corpus the less predictable it is.

In an ideal language model, the entropy of a corpus should not depend on its size. However, word probabilities are difficult to approximate (see, for example, Bell

. . . . . CD inches of rain . . . .
. . . . . acid rain . . . .
. . . . . CD inches of rain fell . . . . .
. . . . . heavy rain . . . .
. . . . . the Atlantic hurricane season . . . . .
. . . . . hurricane force winds . . . . .
. . . . . rain forests . . . . .
. . . . . to reduce acid rain . . . .
. . . . . a major hurricane . . . . .
. . . . . light rain . . . .
. . . . . the most powerful hurricane to hit the . . . .
. . . . . an inch of rain . . . .
. . . . . to save the world s rain forests . . . . .
. . . . . wind and rain . . . .
. . . . . a cold rain . . . . .

Figure 15
Some collocations retrieved from AP.

[1987] for a thorough discussion on probability estimation), and in most cases entropy grows with the size of the corpus. In this section, we use a simple unigram language model trained on the corpus and we approximate the variety of a given corpus by:

$$\theta_{corpus} = -\sum_{w}(f(w)/S)\log(f(w)/S)$$

in which $f(w)$ is the frequency of appearance of the word w in the corpus and S is the total number of different word forms in the corpus. In addition, to be fair in our comparison of the three corpora, we have used three (sub)corpora of about one million words for DJ, NYT, and Brown. The 1 million–word Brown corpus (Francis and Kučera 1982) contains 43,300 different words, of which only 1091 are repeated more than 100 times. The θ of the Brown corpus is: $\theta_{Brown} = 10.5$. In comparison, the size of DJ is 8,000,000. It contains 59,233 different words of which 5,367 are repeated more than 100 times. DJ θ ratio is: $\theta_{DJ} = 9.6$. And the θ ratio of NYT which contains stories pertaining to various domains has been estimated at $\theta_{NYT} = 10.4$. According to this measure, DJ is much more focused than both the Brown Corpus and NYT because the difference in *variety* is 1 in the logarithmic scale. This is not a surprise since the subjects it covers are much more restricted, the genre is of only one kind, and the setting is constant. In contrast, the Brown corpus has been designed to be of mixed and rich composition, and NYT is made up of stories and articles related to various subjects and domains. Let us note that several factors might also influence the overall entropy of a given corpus; for example the number of writers, the time span covered by the corpus, etc. In any case, the success of statistical methods such as the ones described in this report also depends on the sublanguage used in the corpus.

For a sublanguage-dependent application, the training corpus must be focused, mainly because its vocabulary being restricted, the important words will be more frequent than in a nonrestricted corpus (of equivalent size), and thus the collocations will be easier to retrieve. Other applications might require less focused corpora. For those applications, the problem is even more touchy, as a perfectly balanced corpus is very difficult to compile. A sample of the 1987 DJ text is certainly not a good sample

of general English; however, a *balanced* sample, such as the Brown Corpus, may also be a poor sample. It is doubtful that even a balanced corpus contains enough data on all possible domains, and the very effort of artificially balancing the corpus might also bias the results.

12. Some Applications

Corpus-based techniques are still rarely used in the fields of linguistics, lexicography, and computational linguistics, and the main thrust of the work presented here is to promote its use for any text based application. In this section we discuss several uses of **Xtract**.

12.1 Language Generation

Language generation is a novel application for Corpus-Based Computational Linguistics (Boguraev 1989). In Smadja (1991) we show how collocations enhance the task of lexical selection in language generation. Previous language generation works did not use collocations mainly because they did not have the information in compiled form and the lexicon formalisms available did not handle the variability of collocational knowledge. In contrast, we use **Xtract** to produce the collocations and we use Functional Unification Grammars (FUGs) (Kay 1979) as a representation formalism and a unification engine. We show how the use of FUGs allows us to properly handle the interactions of collocational and various other constraints. We have implemented **Cook**, a surface sentence generator that uses a flexible lexicon for expressing collocational constraints in the stock market domain. Using **Ana** (Kukich 1983) as a deep generator, **Cook** is implemented in FUF (Elhadad 1990), an extended implementation of FUG, and uniformly represents the lexicon and syntax as originally suggested by Halliday (1966). For a more detailed description of **Cook** the reader is referred to Smadja (1991).

12.2 Retrieving Grammatical Collocations

According to Benson, Benson, and Ilson (1986a), collocations fall into two major groups: lexical collocations and grammatical collocations. The difference between these two groups lies in the types of words involved. Lexical collocations roughly consist of syntagmatic affinities among open class words such as verbs, nouns, adjectives, and adverbs. In contrast, grammatical collocations generally involve at least one closed class word among particles, prepositions, and auxiliary verbs. Examples of grammatical collocations are: *put-up*, as in *"I can't put up with this anymore,"* and *fill-out*, as in *"You have to fill out your 1040 form."*[19]

Consider the sentences below:

1. "The **comparison to** job hunting is certainly a valid one."
2. ⋆ "The **comparison with** job hunting is certainly a valid one."
3. "The **association with** job hunting is certainly a valid one."
4. ⋆ "The **association to** job hunting is certainly a valid one."
5. "...a new initiative in the **aftermath of** the PLO's evacuation from Beirut."

19 Note that British English uses rather *"to fill in a form."*

6.* "...a new initiative in the **aftermath from** the PLO's evacuation from Beirut."

7. "...a new initiative in the **aftershocks from** the PLO's evacuation from Beirut."

8.* "...a new initiative in the **aftershocks of** the PLO's evacuation from Beirut."

These examples clearly show that the choices of the prepositions are arbitrary. Sentences (1)–(2) and (3)–(4) compare the word associations *comparison with/to* with *association with/to*. Although very similar in meaning, the two words select different prepositions. Moreover, the difference of meaning of the two prepositions does not account for the wording choices. Similarly, sentences (5)–(6) and (7)–(8) illustrate the fact that *"aftermath"* selects the preposition *"of"* and *"aftershock"* selects *"from."*

Grammatical collocations are very similar to lexical collocations in the sense that they also correspond to arbitrary and recurrent word co-occurrences (Benson 1990). In terms of structure, grammatical collocations are much simpler: since many of the grammatical collocations only include one open class word, the separation base-collocator becomes trivial. The open class word is the meaning bearing element, it is the base; and the closed class word is the collocator. For lexicographers, grammatical collocations are somehow simpler than lexical collocations. A large number of dictionaries actually include them. For example, *The Random House Dictionary of the English Language* (RHDEL) (Flexner 1987) gives: *"abreast of, accessible to, accustomed to, careful about, conducive to, conscious of, equal to, expert at, fond of, jealous of,"* etc. However, a large number are missing and the information provided is inconsistent and spotty. For example, RHDEL does not include: *appreciative of, available to, certain of, clever at, comprehensible to, curious about, difficult for, effective against, faithful to, friendly with, furious at, happy about, hostile to,* etc. As demonstrated by Benson, even the most complete learners' dictionaries miss very important grammatical collocations and treat the others inconsistently.[20]

Xtract can be used without modification to retrieve noun–preposition collocations. Figure 16 lists such collocations as retrieved by **Xtract**. Many of the associations retrieved are effectively collocations: *"absence of, accordance with, accuracy of, advantage of, aftershock from, agreement on, allegations of, anxiety about, aspect of,"* etc.

12.3 Some Determiner–Noun Problems

Determiners are lexical elements that are used in conjunction with a noun to bring into correspondence with it a certain sector of reality (Ducrot and Todorov 1979). A noun without determiner has no referent. The role of determiner can be played by several classes of items: articles, (e.g., *"a," "the"*), possessives (e.g., *"my," "your"*), indefinite adjectives (e.g., *"some," "many," "few," "certain"*), demonstratives (e.g., *"this," "those"*), numbers, etc. Determiner–noun combinations are often based simply on semantic or syntactic criteria. For example in the expression *"my left foot,"* the determiner *"my"* is here for semantic reasons. Any other determiner would fail to identify the correct object (my left foot). Classes of nouns such as *mass* and *count* are supposed to determine the type of determiners to be used in conjunction with the nouns (Quirk et al. 1972). Mass nouns often refer to objects or ideas that can be divided into smaller parts without losing their meaning. In contrast, count nouns refer to objects that are not dividable. For example, *"water"* is a mass noun, if you spill half a glass of water you still have

20 For a detailed case study the reader is referred to Benson (1989b).

Noun	part	Noun	part	Noun	part
ability	of	afternoon	from	arbitrage	in
absence	of	aftershocks	from	area	of
acceleration	of	age	of	area	with
acceptance	of	agency	for	areas	of
accordance	with	agency	with	argument	by
account	of	agreement	by	arguments	in
accounts	in	agreements	with	article	in
accuracy	of	alarm	about	articles	on
acquisition	of	alternatives	for	aspects	of
acres	of	amount	of	assault	on
action	by	amounts	of	assessment	of
actions	by	analysis	for	association	with
actions	of	analysis	of	assumption	of
advance	from	announcement	by	attempts	by
advance	of	announcement	of	attention	on
advancers	with	anxiety	about	attorney	for
advances	in	appetite	for	attractiveness	of
advances	on	applications	for	auction	for
advantage	of	appointment	of	auction	in
adviser	in	appraisal	of	auction	of
aftermath	of	approval	from	author	of
aftershocks	from	approval	of	authority	for

Figure 16
Some noun–preposition associations retrieved by **Xtract**.

some *water* left in your glass. In contrast if you cut a book in two halves and discard one half, you do not have a book any more; *"book"* is a count noun. Count nouns are often used with numbers and articles, and mass nouns are often used with no articles (or the *zero article* noted $\emptyset$) (Quirk et al. 1972).

As with other types of word combinations, noun–determiner combinations often lead to collocations. Consider the table given in Table 5. In the table, some noun–determiner combinations are compared. The first four determiners (*a, the,* $\emptyset$*, some*) represent a singular use of the noun, and the last four (*many, few, a lot of, a great deal of*) represent a plural use. 1 and 300 are numbers. $\emptyset$ is the zero article. In the table, a '+' sign means that the combination is frequent and normal; a '−' sign means that the combination is very rare if not forbidden. A '?' sign means that the combination is very low probability and that it would probably require an unusual context. For example, one does not say $\star$*"a butter,"* one says *"some butter,"* and the combination *butter-many* is rather unusual and would only occur in unusual contexts. For example, if one refers to several types of butter, one could say: *"Many butters are based on regular butter and an additional spice or flavor, such as rosemary, sage, basil, garlic, etc."*

"Book" is a typical count noun in that it can combine with *"a"* and *"many."* *"Butter"* is a typical mass noun in that it combines with the zero determiner and *"a great deal."* However, words such as *"police, people, traffic, opinion, weather,"* etc. share some characteristics of both mass nouns and count nouns. For example, *"weather"* is neither a count noun—$\star$*"a weather"* is incorrect—nor a mass noun—$\star$*"a lot of weather"* is incorrect (Quirk et al. 1972). However, it shares some characteristics of both types of nouns. Mass noun features include the premodified structures *"a lot of good weather," "some bad weather,"* and *"what lovely weather."* Count noun features include the plural *"go out in all weathers," "in the worst of weathers."*

Table 5
Some noun–determiner collocations.

Noun/Det	a	the	∅	some	many	few	a lot of	a great deal of	1	300
butter	−	+	+	+	−?	−?	+	+	−	−
book	+	+	−	−	+	+	+	−	+	+
economics	−	+	+	−?	−	−	+	+	−	−
police	−	+	+	+	+	+	+	−	−	+
people	+	+	+	+	+	+	+	+	+	+
opinion	+	+	−	−	+	+	+	−?	+	+
traffic	−	+	+	+	−	−	+	+	−	−
weather	−	+	−	−	−	−	−	+?	−	−

The problem with such combinations is that, if the word is irregular then the information will probably not be in the dictionary.[21] Moreover, even if the word is regular, the word itself might not be in the dictionary or the information could simply be difficult to retrieve automatically.

Simple tools such as **Xtract** can hopefully provide such information. Based on a large number of occurrences of the noun, **Xtract** will be able to make statistical inferences as to the determiners used with it. Such analysis is possible without any modification to **Xtract**. Actually, only a subpart of **Xtract** is necessary to retrieve them.

12.4 Multilingual Lexicography

We have seen that collocations are difficult to handle for non-native speakers, and that they require special handling for computational applications. In a multilingual environment the problems become even more complex, as each language imposes its own collocational constraints. Consider, for example, the English expressions *"House of Parliament"* and *"House painter."* The natural French translation for *"house"* is *"maison."* However, the two expressions do not use this translation, but respectively *"chambre"* (*"room"* in English) and *"bâtiment"* (*"building"* in English). Translations have to be provided for collocations, and should not be word-based but rather expression-based. Bilingual dictionaries are generally inadequate in dealing with such issues. They generally limit such context-sensitive translations to ambiguous words (e.g., *"number" or "rock"*) or highly complex words such as *"make," "have,"* etc. Moreover, even in these cases, coverage is limited to semantic variants, and lexical collocations are generally omitted. One possible application is the development of compilation techniques for bilingual dictionaries. This would require compiling two monolingual collocational dictionaries and then developing some automatic or assisted translation methods. Those translation methods could be based on the statistical analysis of bilingual corpora currently available. A simple algorithm for translating collocations is given in Smadja (1992).

Several other applications such as information retrieval, automatic thesauri compilation, and speech recognition are also discussed in Smadja (1991).

21 Note that it might be in some grammar book. For example, Quirk et al. in their extensive grammar book (1972) devote some 100 pages to such noun–determiner combinations. They include a large number of rules and list exceptions to those rules.

13. Summary and Conclusion

Corpus analysis is a relatively recent domain of research. With the availability of large samples of textual data and automated tools such as part-of-speech taggers, it has become possible to develop and use automatic techniques for retrieving lexical information from textual corpora. In this paper some original techniques for the automatic extraction of collocations have been presented. The techniques have been implemented in a system, **Xtract**, and tested on several corpora. Although some other attempts have been made to retrieve collocations from textual corpora, no work has been able to retrieve the full range of the collocations that **Xtract** retrieves. Thanks to our filtering methods, the collocations produced by **Xtract** are of better quality. And finally, because of the syntactic labeling, the collocations we produce are richer than the ones produced by other methods.

The number and size of available textual corpora is constantly growing. Dictionaries are available in machine-readable form, news agencies provide subscribers with daily reports on various events, publishing companies use computers and provide machine-readable versions of books, magazines, and journals. This amounts to a vast quantity of language data with unused and virtually unlimited, implicit and explicit information about the English language. These textual data can thus be used to retrieve important information that is not available in other forms. The primary goal of the research we presented is to provide a comprehensive lexicographic toolkit to assist in implementing natural language processing, as well as to assist lexicographers in compiling general-purpose dictionaries, as most of the work is still manually performed in this domain. The abundance of text corpora allows a shift toward more empirical studies of language that emphasize the development of automated tools. We think that more research should be conducted in this direction and hope that our work will stimulate research projects along these lines.

Acknowledgments

I would like to thank Steve Abney, Ken Church, Karen Kukich, and Michael Elhadad for making their software tools available to us. Without them, most of the work reported here would not have been possible. Kathy McKeown read earlier versions of this paper and was helpful in both the writing and the research. Finally, the anonymous reviewers for *Computational Linguistics* made insightful comments on earlier versions of the paper.

Part of this work has been done in collaboration with Bell Communication Research, and part of this work has been supported by DARPA grant N00039-84-C-0165, by NSF grant IRT-84-51438, and by ONR grant N00014-89-J-1782.

References

Abney, S. (1989). Parsing by Chunks. In *The MIT Parsing Volume*, edited by C. Tenny. MIT Press.

Abney, S. (1990). "Rapid incremental parsing with repair." In *Proceedings, Waterloo Conference on Electronic Text Research*, 1990.

Allerton, D. J. (1984). "Three or four levels of co-occurrence relations." *Lingua*, **63**, 17–40.

Amsler, B. (1989). "Research towards the development of a lexical knowledge base for natural language processing." In *Proceedings, 1989 SIGIR Conference*. Cambridge, MA.

Angell, R. C. (1983). "Automating spelling correction using a trigram similarity measure." *Information Processing and Management*, **19**, 255–261.

Bahl, L.; Jelinek, F.; and Mercer, R. (1983). "A maximum likelihood approach to continuous speech recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **5**(2), 179–190.

Bell, T.; Witten, I.; and Cleary, J. (1989). "Modelling for text compression." *ACM Computing Surveys*, **21**(4), 557–591.

Bell, T. (1987). "A unifying theory and improvement for existing approaches to text compression." Doctoral dissertation, University of Canterbury, Christchurch, New Zealand.

Benson, M.; Benson, E.; and Ilson, R. (1986a). *The BBI Combinatory Dictionary of English: A Guide to Word Combinations*. John Benjamins.

Benson, M.; Benson, E.; and Ilson, R. (1986b). *The Lexicographic Description of English*. John Benjamins.

Benson, M. (1989a). "The collocational dictionary and the advanced learner." In *Learner's Dictionaries: State of the Art*, edited by M. Tickoo, 84–93. SEAMEO.

Benson, M. (1989b). "The structure of the collocational dictionary." *International Journal of Lexicography*, **2**, 1–14.

Benson, M. (1990). "Collocations and general-purpose dictionaries." *International Journal of Lexicography*, **3**(1), 23–35.

Boguraev, B. (1989). "Introduction." In *Computational Lexicography for Natural Language Processing*, Chapter 1, edited by T. Boguraev and B. Briscoe. Longman.

Brown, P.; Cocke, J.; Della Pietra, V.; Della Pietra, S.; Jelinek, F.; Mercer, R.; and Roossin, P. (1988). "A statistical approach to language translation." In *Proceedings of the 13th International Conference on Computational Linguistics (COLING-88)*, 71–76.

Cerf-Danon, H.; Derouault, A. M.; Elbeze, M.; and Merialdo, B. (1989). "Speech recognition in French with a very large dictionary." In *Eurospeech*.

Choueka, Y.; Klein, T.; and Neuwitz, E. (1983). "Automatic retrieval of frequent idiomatic and collocational expressions in a large corpus." *Journal for Literary and Linguistic Computing*, **4**, 34–38.

Choueka, Y. (1988). "Looking for needles in a haystack." In *Proceedings, RIAO Conference on User-Oriented Context Based Text and Image Handling*, 609–623. Cambridge, MA.

Church, K., and Gale, W. (1990). "Poor estimates of context are worse than none." In *Darpa Speech and Natural Language Workshop*, Hidden Valley, PA.

Church, K., and Hanks, P. (1989). "Word association norms, mutual information, and lexicography." In *Proceedings, 27th Meeting of the ACL*, 76–83. Also in *Computational Linguistics*, **16**(1).

Church, K. W.; Gale, W.; Hanks, P.; and Hindle, D. (1989). "Parsing, word associations and typical predicate-argument relations." In *Proceedings of the International Workshop on Parsing Technologies*, 103–112. Carnegie Mellon University, Pittsburgh, PA.

Church, K.; Gale, W.; Hanks, P.; and Hindle, D. (1991). "Using statistics in lexical analysis." In *Lexical Acquisition: Using On-Line Resources to Build a Lexicon*, edited by Uri Žernik. Lawrence Erlbaum.

Church, K. (1988). "Stochastic parts program and noun phrase parser for unrestricted text." In *Proceedings, Second Conference on Applied Natural Language Processing*. Austin, TX.

Cowie, A. P. (1981). "The treatment of collocations and idioms in learner's dictionaries." *Applied Linguistics*, **2**(3), 223–235.

Cruse, D. A. (1986). *Lexical Semantics*. Cambridge University Press.

Debili, F. (1982). *Analyse Syntactico-Sémantique Fondée sur une Acquisition Automatique de Relations Lexicales Sémantiques*. Doctoral dissertation, Paris XI University, Orsay, France. Thèse de Doctorat D'état.

Dellenbaugh, D., and Dellenbaugh, B. (1990). *Small Boat Sailing, a Complete Guide*. Sports Illustrated Winner's Circle Books.

Ducrot, O., and Todorov, T. (1979). *Encyclopedic Dictionary of the Sciences of Language*. John Hopkins University Press.

Elhadad, M. (1990). "Types in functional unification grammars." In *Proceedings, 28th Meeting of the Association for Computational Linguistics*.

Ephraim, Y., and Rabiner, L. (1990). "On the relations between modeling approaches for speech recognition." *IEEE Transactions on Information Theory*, **36**(2), 372–380.

Fano, R. (1961). *Transmission of Information: A Statistical Theory of Information*. MIT Press.

Flexner, S., ed. (1987). *The Random House Dictionary of the English Language, Second Edition*. Random House.

Francis, W., and Kučera, H. (1982). *Frequency Analysis of English Usage*. Houghton Mifflin.

Garside, R., and Leech, G. (1987). *The Computational Analysis of English, a Corpus Based Approach*. Longman.

Guazzo, M. (1980). "A general minimum-redundancy source-coding algorithm." *IEEE Transactions on Information Theory*, IT-26(1), 15–25.

Halliday, M. A. K., and Hasan, R. (1976). *Cohesion in English*. Longman.

Halliday, M. A. K. (1966). "Lexis as a linguistic level." In *In Memory of J. R. Firth*, edited by C. E. Bazell, J. C. Catford, M. A. K. Halliday, and R. H. Robins, 148–162. Longmans Linguistics Library.

Hindle, D., and Rooth, M. (1990). "Structural ambiguity and lexical relations." In *DARPA Speech and Natural Language Workshop*, Hidden Valley, PA.

Hindle, D. (1983). "User manual for

fidditch, a deterministic parser." Technical Memorandum 7590-142, Naval Research Laboratory.

Kay, M. (1979). "Functional grammar." In *Proceedings, 5th Meeting of the Berkeley Linguistics Society*. Berkeley Linguistics Society.

Kukich, K. (1983). "Knowledge-based report generation: A technique for automatically generating natural language reports from databases." In *Proceedings, Sixth International ACM SIGIR, Conference on Research and Development in Information Retrieval*. Washington, D.C.

Kukich, K. (1990). "A comparison of some novel and traditional lexical distances metrics for spelling correction." In *Proceedings, International Neural Networks Conference (INNC)*. Paris, France.

Marcus, M. (1990). "Tutorial on tagging and processing large textual corpora." Presented at the 28th Annual Meeting of the ACL.

Martin, W. J. R.; Al, B. P. F.; and Van Sterkenburg, P. J. G. (1983). "On the processing of a text corpus: from textual data to lexicographical information." In *Lexicography: Principles and Practice*, Applied Language Studies Series, edited by R. R. K. Hartmann. Academic Press.

Mays, E.; Damerau, F.; and Mercer, R. (1990). "Context-based spelling correction." In *IBM Natural Language ITL*, Paris, France.

Mel'čuk, I. A. (1981). "Meaning-text models: a recent trend in Soviet linguistics." *The Annual Review of Anthropology*.

Merialdo, B. (1987). "Speech recognition with very large size dictionary." In *Proceedings, International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Dallas, TX.

Morris, R., and Cherry, L. L. (1975). "Computer detection of typographical errors." *IEEE Transactions on Professional Communications*, PC-**18**(1), 54–63.

Nakhimovsky, A. D., and Leed, R. L. (1979). "Lexical functions and language learning." *Slavic and East European Journal*, **23**(1).

Quirk, R.; Greenbaum, S.; Leech, G.; and Svartvik, J. (1972). *A Comprehensive Grammar of the English Language*. Longman.

Salton, J. (1989). *Automatic Text Processing, The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley.

Shannon, C. E. (1948). "A mathematical theory of communication." *Bell System Tech.*, **27**, 379–423, 623–656.

Smadja, F., and McKeown, K. (1990). "Automatically extracting and representing collocations for language generation." In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, Pittsburgh, PA.

Smadja, F. (1991). "Retrieving collocational knowledge from textual corpora. An application: Language generation." Doctoral dissertation, Computer Science Department, Columbia University.

Smadja, F. (1992). "How to compile a bilingual collocational lexicon automatically." In *Proceedings of the AAAI Workshop on Statistically-Based NLP Techniques*, San Jose, CA.