# AUTOMATICALLY EXTRACTING AND REPRESENTING COLLOCATIONS FOR LANGUAGE GENERATION*

**Frank A. Smadja**†
and
**Kathleen R. McKeown**
Department of Computer Science
Columbia University
New York, NY 10027

**ABSTRACT**

Collocational knowledge is necessary for language generation. The problem is that collocations come in a large variety of forms. They can involve two, three or more words, these words can be of different syntactic categories and they can be involved in more or less rigid ways. This leads to two main difficulties: collocational knowledge has to be acquired and it must be represented flexibly so that it can be used for language generation. We address both problems in this paper, focusing on the acquisition problem. We describe a program, **Xtract**, that automatically acquires a range of collocations from large textual corpora and we describe how they can be represented in a flexible lexicon using a unification based formalism.

## 1 INTRODUCTION

Language generation research on lexical choice has focused on syntactic and semantic constraints on word choice and word ordering. *Collocational constraints*, however, also play a role in how words can co-occur in the same sentence. Often, the use of one word in a particular context of meaning will require the use of one or more other words in the same sentence. While phrasal lexicons, in which lexical associations are pre-encoded (*e.g.*, [Kukich 83], [Jacobs 85], [Danlos 87]), allow for the treatment of certain types of collocations, they also have problems. Phrasal entries must be compiled by hand which is both expensive and incomplete. Furthermore, phrasal entries tend to capture rather rigid, idiomatic expressions. In contrast, 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, while in others they can be separated by a varying number of other words.

*The research reported in this paper was partially supported by DARPA grant N00039-84-C-0165, by NSF grant IRT-84-51438 and by ONR grant N00014-89-J-1782.

†Most of this work is also done in collaboration with Bell Communication Research, 445 South Street, Morristown, NJ 07960-1910

In this paper, we identify a range of collocations that are necessary for language generation, including open compounds of two or more words, predicative relations (*e.g., subject-verb*), and phrasal templates representing more idiomatic expressions. We then describe how **Xtract** automatically acquires the full range of collocations using a two stage statistical analysis of large domain specific corpora. Finally, we show how collocations can be efficiently represented in a flexible lexicon using a unification based formalism. This is a word based lexicon that has been macrocoded with collocational knowledge. Unlike a purely phrasal lexicon, we thus retain the flexibility of word based lexicons which allows for collocations to be combined and merged in syntactically acceptable ways with other words or phrases of the sentence. Unlike pure word based lexicons, we gain the ability to deal with a variety of phrasal entries. Furthermore, while there has been work on the automatic retrieval of lexical information from text [Garside 87], [Choueka 88], [Klavans 88], [Amsler 89], [Boguraev & Briscoe 89], [Church 89], none of these systems retrieves the entire range of collocations that we identify and no real effort has been made to use this information for language generation [Boguraev & Briscoe 89].

In the following sections, we describe the range of collocations that we can handle, the fully implemented acquisition method, results obtained, and the representation of collocations in Functional Unification Grammars (FUGs) [Kay 79]. Our application domain is the domain of stock market reports and the corpus on which our expertise is based consists of more than 10 million words taken from the Associated Press news wire.

## 2 SINGLE WORDS TO WHOLE PHRASES: WHAT KIND OF LEXICAL UNITS ARE NEEDED?

Collocational knowledge indicates which members of a set of roughly synonymous words co-occur with other words and how they combine syntactically. These affinities can not be predicted on the basis of semantic or syntactic rules, but can be observed with some regularity in text [Cruse 86]. We have found a range of collocations from word pairs to whole phrases, and as we shall show,

this range will require a flexible method of representation.

**Open Compounds** . Open compounds 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, adjectives, and closed class words and are similar to the type of collocations retrieved by [Choueka 88] or [Amsler 89]. An open compound generally functions as a single constituent of a sentence. More open compound examples are given in figure 1.[1]

**Predicative Relations** consist of two (or several) words repeatedly used together in a similar syntactic relation. These lexical relations are harder to identify since they often correspond to interrupted word sequences in the corpus. They are also the most flexible in their use. This class of collocations is related to Mel'čuk's Lexical Functions [Mel'čuk 81], and Benson's L-type relations [Benson 86]. Within this class, **Xtract** retrieves *subject-verb, verb-object, noun-adjective, verb-adverb, verb-verb* and *verb-particle* predicative relations. Church [Church 89] also retrieves *verb-particle* associations. Such collocations require a representation that allows for a lexical function relating two or more words. Examples of such collocations are given in figure 2.[2]

**Phrasal templates:** consist of idiomatic phrases containing one, several or no empty slots. They are extremely rigid and long collocations. These almost complete phrases are quite representative of a given domain. Due to their slightly idiosyncratic structure, we propose representing and generating them by simple template filling. Although some of these could be generated using a word based lexicon, in general, their usage gives an impression of fluency that cannot be equaled with compositional generation alone. **Xtract** has retrieved several dozens of such templates from our stock market corpus, including:

*"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*"*

[1] All the examples related to the stock market domain have been actually retrieved by Xtract.

[2] 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.

# 3 THE ACQUISITION METHOD: Xtract

In order to produce sentences containing collocations, a language generation system must have knowledge about the possible collocations that occur in a given domain. In previous language generation work [Danlos 87], [Iordanskaja 88], [Nirenburg 88], collocations are identified and encoded by hand, sometimes using the help of lexicographers (e.g., Danlos' [Danlos 87] use of Gross' [Gross 75] work). This is an expensive and time-consuming process, and often incomplete. In this section, we describe how **Xtract** can automatically produce the full range of collocations described above.

**Xtract** has two main components, a concordancing component, **Xconcord**, and a statistical component, **Xstat**. Given one or several words, **Xconcord** locates all sentences in the corpus containing them. **Xstat** is the co-occurrence compiler. Given **Xconcord**'s output, it makes statistical observations about these words and other words with which they appear. Only statistically significant word pairs are retained. In [Smadja 89a], and [Smadja 88], we detail an earlier version of **Xtract** and its output, and in [Smadja 89b] we compare our results both qualitatively and quantitatively to the lexicon used in [Kukich 83]. **Xtract** has also been used for information retrieval in [Maarek & Smadja 89]. In the updated version of **Xtract** we describe here, statistical significance is based on four parameters, instead of just one, and a second stage of processing has been added that looks for combinations of word pairs produced in the first stage, resulting in multiple word collocations.

**Stage one:** In the first phase, **Xconcord** is called for a single open class word and its output is pipelined to **Xstat** which then analyzes the distribution of words in this sample. The output of this first stage is a list of tuples ($w_1, w_2$, *distance, strength, spread, height, type*), where ($w_1, w_2$) is a lexical relation between two open-class words ($w_1$ and $w_2$). Some results are given in Table 1. *"Type"* represents the syntactic categories of $w_1$ and $w_2$.[3]. *"Distance"* is the relative distance between the two words, $w_1$ and $w_2$ (*e.g.*, a distance of 1 means $w_2$ occurs immediately after $w_1$ and a distance of -1 means it occurs immediately before it). A different tuple is produced for each statistically significant word pair and distance. Thus, if the same two words occur equally often separated by two different distances, they will appear twice in the list. *"Strength"* (also computed in the earlier version of **Xtract**) indicates how strongly the two words are related (see [Smadja 89a]). *"Spread"* is the distribution of the relative distance between the two words; thus, the larger the *"spread"* the more rigidly they are used in combination to one another. *"Height"* combines the factors of *"spread"*

[3] In order to get part of speech information we use a stochastic word tagger developed at AT&T Bell Laboratories by Ken Church [Church 88]

Table 1: Some binary lexical relations.

| word1 | word2 | distance | strength | spread | height | Type |
|---|---|---|---|---|---|---|
| stock | market | 1 | 47.018 | 28.5 | 11457.1 | NN |
| president | vice | -1 | 40.6496 | 29.7 | 10757 | NN |
| trade | deficit | 1 | 30.3384 | 28.4361 | 7358.87 | NN |
| directors | board | -2 | 22.6038 | 28.7682 | 5611.84 | NN |
| merger | agreement | 1 | 20.62 | 28.7682 | 5119.32 | NN |
| attempt | takeover | -1 | 21.1464 | 28.407 | 5118.02 | NN |
| average | industrial | -1 | 13.1674 | 29.3682 | 3406.85 | NJ |
| index | composite | -1 | 12.3874 | 29.0682 | 3139.89 | NJ |
| chip | blue | -1 | 10.078 | 30 | 2721.06 | NJ |
| shares | totaled | -4 | 20.7815 | 29.3682 | 5376.87 | NV |
| price | closing | -1 | 23.0465 | 25.9415 | 4615.48 | NV |
| stocks | listed | -2 | 27.354 | 23.8696 | 4583.57 | NV |
| volume | totaled | 1 | 16.8724 | 29.7 | 4464.89 | NV |
| takeover | bid | -1 | 19.3312 | 28.1071 | 4580.39 | NN |
| takeovers | hostile | 1 | 13.5184 | 29.3682 | 3497.67 | NJ |
| takeover | offer | -1 | 5.43739 | 25.7917 | 1084.05 | NN |
| takeovers | thwart | 2 | 2.61206 | 30 | 705.256 | NV |

Table 2: Concordances for *"average industrial"*

| | |
|---|---|
| On Tuesday the Dow Jones industrial average | rose 26.28 points to 2 304.69. |
| The Dow Jones industrial average | went up 11.36 points today. |
| ... a selling spurt that sent the Dow Jones industrial average | down sharply in the first hour of trading. |
| On Wednesday the Dow Jones industrial average | showed some strength as ... |
| The Dow Jones industrial average | was down 17.33 points to 2,287.36 ... |
| The Dow Jones industrial average | had the biggest one day gain of its history ... |
| ... Thursday with the Dow Jones industrial average | soaring a record 69.89 points to ... |
| ... swelling the Dow Jones industrial average | by more than 475 points in the process ... |
| The rise in the Dow Jones industrial average | was the biggest since a 54.14 point jump on ... |

Table 3: Concordances for *"composite index"*

| | |
|---|---|
| The NYSE s composite index | of all its listed common stocks fell 1.76 to 164.13. |
| The NYSE s composite index | of all its listed common stocks fell 0.98 to 164.91. |
| The NYSE s composite index | of all its listed common stocks fell 0.96 to 164.93. |
| The NYSE s composite index | of all its listed common stocks fell 0.91 to 164.98. |
| The NYSE s composite index | of all its listed common stocks rose 1.04 to 167.08. |
| The NYSE s composite index | of all its listed common stocks rose 0.76 |
| The NYSE s composite index | of all its listed common stocks rose 0.50 to 166.54. |
| The NYSE s composite index | of all its listed common stocks rose 0.69 to 166.73. |
| The NYSE s composite index | of all its listed common stocks fell 0.33 to 170.63. |

| type of collocation | examples |
|---|---|
| open compound | "**leading** *industrialized* **countries**" |
| open compound | *"the Dow Jones* **average** *of 30* **industrials**" |
| open compound | **"bear/bull market"** |
| open compound | *"the Dow Jones* **industrial average**" |
| open compound | *"The NYSE's composite index of all its listed common stocks"* |
| open compound | **"Advancing/winning/losing/declining issues"** |
| open compound | *"The NASDAQ* **composite** *index for the over the counter* **market**" |
| open compound | **"stock market"** |
| open compound | **"central bank** |
| open compound | **"leveraged buyout"** |
| open compound | *"the* **gross** *national* **product**" |
| open compound | *"***blue** *chip* stocks" |
| open compound | *"White* **House spokesman** *Marlin Fitzwater"* |
| open compound | **"takeover speculation/strategist/target/threat/attempt"** |
| open compound | **"takeover bid/battle/defense/efforts/fight/law/proposal/rumor"** |

Figure 1: Some examples of open compounds

| type of collocation | examples |
|---|---|
| noun adjective | **"heavy/light [] trading/smoker/traffic"** |
| noun adjective | **"high/low [] fertility/pressure/bounce"** |
| noun adjective | **"large/small [] crowd/retailer/client"** |
| subject verb | **"index [] rose** |
| subject verb | **"stock [] [rose, fell, closed, jumped, continued, declined, crashed, ...]"** |
| subject verb | **"advancers [] [outnumbered, outpaced, overwhelmed, outstripped]"** |
| verb adverb | **"trade ⇔ actively," "mix ⇔ narrowly," "use ⇔ widely," "watch ⇔ closely"** |
| verb object | **"posted [] gain** |
| verb object | **"momentum [] [pick** *up,* **build, carry** *over,* **gather, loose, gain]"** |
| verb particle | **"take [] from," "raise [] by," "mix [] with"** |
| verb verb | **"offer** *to* **[acquire, buy"]** |
| verb verb | **"agree** *to* **[acquire, buy"]** |

Figure 2: Some examples of predicative collocations

and *"strength"* resulting in a ranking of the two words for their *"distances"*. Church [Church 89] produces results similar to those presented in the table using a different statistical method. However, Church's method is mainly based on the computation of the *"strength"* attribute, and it does not take into account *"spread"* and *"height"*. As we shall see, these additional parameters are crucial for producing multiple word collocations and distinguishing between open compounds (words are adjacent) and predicative relations (words can be separated by varying distance).

**Stage two:** In the second phase, **Xtract** first uses the same components but in a different way. It starts with the pairwise lexical relations produced in Stage one to produce multiple word collocations, then classifies the collocations as one of three classes identified above, and finally attempts to determine the syntactic relations between the words of the collocation. To do this, **Xtract** studies the lexical relations in context, which is exactly what lexicographers do. For each entry of Table 1, **Xtract** calls **Xconcord** on the two words $w_1$ and $w_2$ to produce the concordances. Tables 2 and 3 show the concordances (output of **Xconcord**) for the input pairs: *"average-industrial"* and *"index-composite"*. **Xstat** then compiles information on the words surrounding both $w_1$ and $w_2$ in the corpus. This stage allows us to filter out incorrect associations such as *"blue-stocks"* or *"advancing-market"* and replace them with the appropriate ones, *"blue chip stocks," "the broader market in the NYSE advancing issues."* This stage also produces phrasal templates such as those given in the previous section. In short, stage two filters inapropriate results and combines word pairs to produce multiple word combinations. To make the results directly usable for language generation we are currently investigating the use of a bottom-up parser in combination with stage two in order to classify the collocations according to syntactic criteria. For example if the lexical relation involves a noun and a verb it determines if it is a *subject-verb* or a *verb-object* collocation. We plan to do this using a deterministic bottom up parser developed at Bell Communication Research [Abney 89] to parse the concordances. The parser would analyze each sentence of the concordances and the parse trees would then be passed to **Xstat**.

Sample results of Stage two are shown in Figures 1, 2 and 3. Figure 3 shows phrasal templates and open compounds. **Xstat** notices that the words *"composite* and *"index"* are used very rigidly throughout the corpus. They almost always appear in one of the two

| 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 3: Example collocations output of stage two.

sentences. The lexical relation *composite-index* thus produces two phrasal templates. For the lexical relation *average-industrial* **Xtract** produces an open compound collocation as illustrated in figure 3. Stage two also confirms pairwise relations. Some examples are given in figure 2. By examining the parsed concordances and extracting recurring patterns, **Xstat** produces all three types of collocations.

# 4 HOW TO REPRESENT THEM FOR LANGUAGE GENERATION?

Such a wide variety of lexical associations would be difficult to use with any of the existing lexicon formalisms. We need a flexible lexicon capable of using single word entries, multiple word entries as well as phrasal templates and a mechanism that would be able to gracefully merge and combine them with other types of constraints. The idea of a flexible lexicon is not novel in itself. The lexical representation used in [Jacobs 85] and later refined in [Desemer & Jabobs 87] could also represent a wide range of expressions. However, in this language, collocational, syntactic and selectional constraints are mixed together into phrasal entries. This makes the lexicon both difficult to use and difficult to compile. In the following we briefly show how FUGs can be successfully used as they offer a flexible declarative language as well as a powerful mechanism for sentence generation.

We have implemented a first version of **Cook**, a surface generator that uses a flexible lexicon for expressing co-occurrence constraints. **Cook** uses FUF [Elhadad 90], an extended implementation of FUGs, to uniformly represent the lexicon and the syntax as originally suggested by Halliday [Halliday 66]. Generating a sentence is equivalent to unifying a semantic structure (Logical Form) with the grammar. The grammar we use is divided into three zones, the *"sentential,"* the *"lexical"* and *"the syntactic zone."* Each zone contains constraints pertaining to a given domain and the input logical form is unified in turn with the three zones. As it is, full backtracking across the three zones is allowed.

- The *sentential zone* contains the phrasal templates against which the logical form is unified first. A sentential entry is a whole sentence that should be used in a given context. This context is specified by subparts of the logical form given as input. When there is a match at this point, unification succeeds and generation is reduced to simple template filling.
- The *lexical zone* contains the information used to lexicalize the input. It contains collocational information along with the semantic context in which to use it. This zone contains predicative and open compound collocations. Its role is to trigger phrases or words in the presence of other words or phrases. Figure 5 is a portion of the lexical grammar used in **Cook**. It illustrates the choice of the verb to be used when *"advancers"* is the subject. (See below for more detail).
- The *syntactic zone* contains the syntactic grammar. It is used last as it is the part of the grammar ensuring the correctness of the produced sentences.

An example input logical form is given in Figure 4. In this example, the logical form represents the fact that on the New York stock exchange, the advancing issues (semantic representation or `sem-R`: *c:winners*) were ahead (predicate *c:lead*) of the losing ones (`sem-R`: *c:losers*) and that there were 3 times more winning issues than losing ones *ratio*). In addition, it also says that this ratio is of *degree* 2. A degree of 1 is considered as a slim lead whereas a degree of 5 is a commanding margin. When unified with the grammar, this logical form produces the sentences given in Figure 6.

As an example of how **Cook** uses and merges co-occurrence information with other kind of knowledge consider Figure 5. The figure is an edited portion of the lexical zone. It only includes the parts that are relevant to the choice of the verb when *"advancers"* is the subject. The `lex` and `sem-R` attributes specify the lexeme we are considering (*"advancers"*) and its semantic representation (*c:winners*).

The semantic context (`sem-context`) which points to the logical form and its features will then be used in order

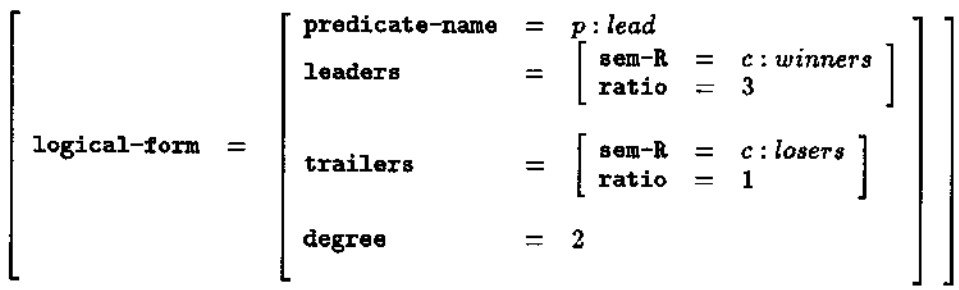


Figure 4: LF: An example logical form used by Cook

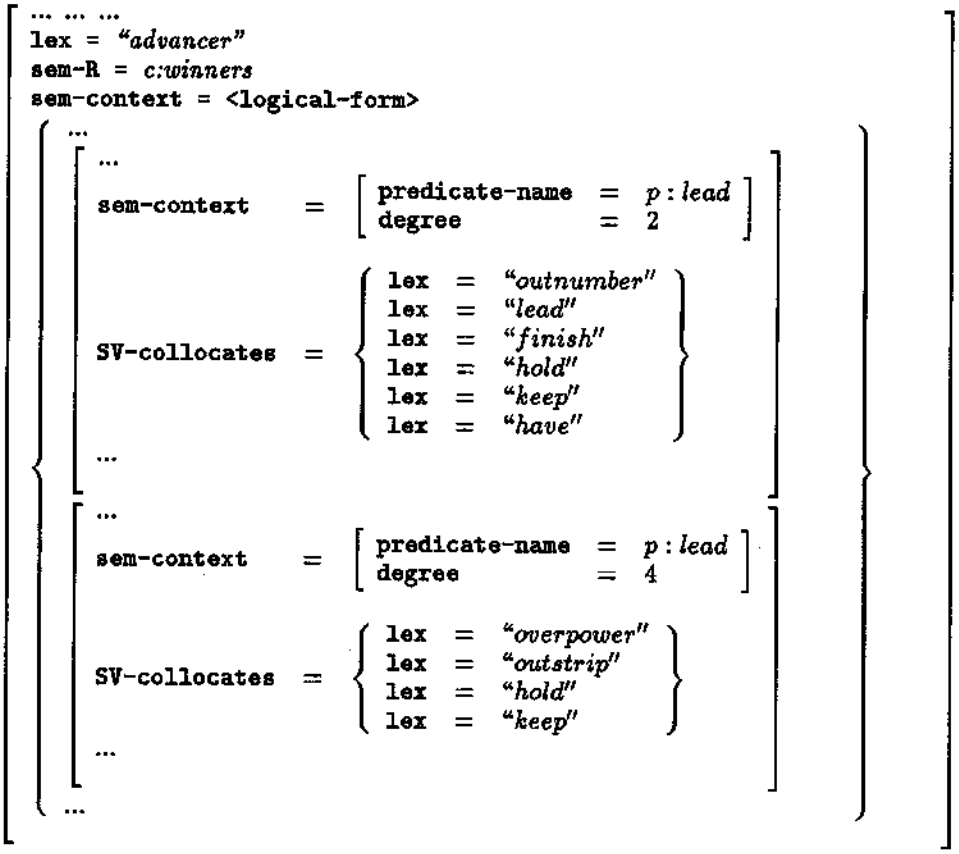


Figure 5: A portion of the lexical grammar showing the verbal collocates of "advancers".

*"Advancers outnumbered declining issues by a margin of 3 to 1."*
*"Advancers had a slim lead over losing issues with a margin of 3 to 1."*
*"Advancers kept a slim lead over decliners with a margin of 3 to 1"*

Figure 6: Example sentences that can be generated with the logical form LF

to select among the alternatives classes of verbs. In the figure we only included two alternatives. Both are relative to the predicate `p:lead` but they are used with different values of the *degree* attribute. When the *degree* is 2 then the first alternative containing the verbs listed under `SV-collocates` (*e.g. "outnumber"*) will be selected. When the degree is 4 the second alternative containing the verbs listed under `SV-collocates` (*e.g. "overpower"*) will be selected. All the verbal collocates shown in this figure have actually been retrieved by **Xtract** at a preceding stage.

The unification of the logical form of Figure 4 with the lexical grammar and then with the syntactic grammar will ultimately produce the sentences shown in Figure 6 among others. In this example, the sentencial zone was not used since no phrasal template expresses its semantics. The verbs selected are all listed under the `SV-collocates` of the first alternative in Figure 5.

We have been able to use **Cook** to generate several sentences in the domain of stock market reports using this method. However, this is still on-going research and the scope of the system is currently limited. We are working on extending **Cook**'s lexicon as well as on developing extensions that will allow flexible interaction among collocations.

## 5 CONCLUSION

In summary, we have shown in this paper that there are many different types of collocations needed for language generation. Collocations are flexible and they can involve two, three or more words in various ways. We have described a fully implemented program, **Xtract**, that automatically acquires such collocations from large textual corpora and we have shown how they can be represented in a flexible lexicon using FUF. In FUF, cooccurrence constraints are expressed uniformly with syntactic and semantic constraints. The grammar's function is to satisfy these multiple constraints. We are currently working on extending **Cook** as well as developing a full sized from **Xtract**'s output.

## ACKNOWLEDGMENTS

We would like to thank Karen Kukich and the Computer Systems Research Division at Bell Communication Research for their help on the acquisition part of this work.

## References

[Abney 89] S. Abney, *"Parsing by Chunks"* in C. Tenny, ed., The MIT Parsing Volume, 1989, to appear.

[Amsler 89] R. Amsler, *"Research Towards the Development of a Lexical Knowledge Base for Natural Language Processing"* Proceedings of the 1989 SIGIR Conference, Association for Computing Machinery. Cambridge, Ma, June 1989.

[Benson 86] M. Benson, E. Benson and R. Ilson, *Lexicographic Description of English.* John Benjamins Publishing Company, Philadelphia, 1986.

[Boguraev & Briscoe 89] B. Boguraev & T. Briscoe, in *Computational Lexicography for natural language processing.* B. Boguraev and T. Briscoe editors. Longmans, NY 1989.

[Choueka 88] Y. Choueka, *Looking for Needles in a Haystack.* In Proceedings of the RIAO, p:609-623, 1988.

[Church 88] K. Church, *A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text* In Proceedings of the Second Conference on Applied Natural Language Processing, Austin, Texas, 1988.

[Church 89] K. Church & K. Hanks, *Word Association Norms, Mutual Information, and Lexicography.* In Proceedings of the 27th meeting of the Association for Computational Linguistics, Vancouver, B.C, 1989.

[Cruse 86] D.A. Cruse, *Lexical Semantics.* Cambridge University Press, 1986.

[Danlos 87] L. Danlos, *The Linguistic Basis of Text Generation.* Cambridge University Press, 1987.

[Desemer & Jabobs 87] D. Desemer & P. Jacobs, *FLUSH: A Flexible Lexicon Design.* In proceedings of the 25th Annual Meeting of the ACL, Stanford University, CA, 1987.

[Elhadad 90] M. Elhadad, *Types in Functional Unification Grammars,* Proceedings of the 28th meeting of the Association for Computational Linguistics, Pittsburgh, PA, 1990.

[Garside 87] R. Garside, G. Leech & G. Sampson, editors, *The computational Analysis of English, a corpus based approach.* Longmans, NY 1987.

[Gross 75] M. Gross, *Méthodes en Syntaxe.* Hermann, Paris, France, 1975.

[Halliday 66] M.A.K. Halliday, *Lexis as a Linguistic Level.* In C.E. Bazell, J.C. Catford, M.A.K Halliday and R.H. Robins (eds.), *In memory of J.R. Firth* London: Longmans Linguistics lla Library, 1966, pp: 148-162.

[Iordanskaja 88] L. Iordanskaja, R. Kittredge, A. Polguere, *Lexical Selection and Paraphrase in a Meaning-Text Generation Model.* Presented at the fourth International Workshop on Language Generation, Catalina Island, CA, 1988.

[Jacobs 85] P. Jacobs, *PHRED: a generator for natural language interfaces,* Computational Linguistics, volume 11-4, 1985

[Kay 79] M. Kay, *Functional Grammar,* in Proceedings of the 5th Meeting of the Berkeley Linguistic Society, Berkeley Linguistic Society, 1979.

[Klavans 88] J. Klavans, *"COMPLEX: a computational lexicon for natural language systems."* In proceeding of the 12th International Conference on Computational Linguistics, Budapest, Hungary, 1988.

[Kukich 83] K. Kukich, *Knowledge-Based Report Generation: A Technique for Automatically Generating Natural Language Reports from Databases.* Proceedings of the 6th International ACM SIGIR Conference, Washington, DC, 1983.

[Maarek & Smadja 89] Y.S Maarek & F.A. Smadja, *Full Text Indexing Based on Lexical Relations, An Application: Software Libraries.* Proceedings of the 12th International ACM SIGIR Conference, Cambridge, Ma, June 1989.

[Mel'čuk 81] I.A Mel'čuk, *Meaning-Text Models: a Recent Trend in Soviet Linguistics.* The annual review of anthropology, 1981.

[Nirenburg 88] S. Nirenburg *et. al.*, *Lexicon building in natural language processing.* In program and abstracts of the $15^{th}$ International ALLC, Conference of the Association for Literary and Linguistic Computing, Jerusalem, Israel, 1988.

[Smadja 88] F.A. Smadja, *Lexical Co-occurrence: The Missing link.* In program and abstracts of the $15^{th}$ International ALLC, Conference of the Association for Literary and Linguistic Computing, Jerusalem, Israel, 1988. Also in the Journal for Literary and Linguistic computing, Vol. 4, No. 3, 1989, Oxford University Press.

[Smadja 89a] F.A. Smadja, *Microcoding the Lexicon for Language Generation*, First International Workshop on Lexical Acquisition, IJCAI'89, Detroit, Mi, August 89. Also in *"Lexical Acquisition: Using on-line resources to build a lexicon"*, MIT press, Uri Žernik editor, to appear.

[Smadja 89b] F.A. Smadja, *On the Use of Flexible Collocations for Language Generation.* Columbia University, technical report, TR# CUCS-507-89.