Microcoding the Lexicon with Co-occurrence Knowledge

CUCS-448-89

Frank Smadja

Department of Computer Science
Columbia University
New York, NY 10027
Smadja@cs.columbia.edu

August 1989

Presented at the 1st International Workshop of Lexical Acquisition,
IJCAI'89, August 89, Detroit, Mi.

This work was partly carried out at Bar Ilan university, Department of English, and partly at Columbia University supported by DARPA under contract #N00039-84-C-0165 and NSF grant IRT-84-51438.

To be presented at the First Int'l Workshop of Lexical Acquisition, 08/21/89, Detroit, Michigan.

Microcoding the Lexicon with Co-Occurrence Knowledge

CUCS-448-89

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

Use: language generation. Contents: co-occurrence knowledge wired in.
Acquisition resource: textual corpora. Utilities: Co-occurrence compiler.

Introduction

Neither syntax nor semantics can justify the use of a certain class of English word combinations. This class contains word pairs that often appear together in a given context of meaning. Such pairs are called *co-occurrence relations* or *idiosyncratic collocations* [3]. To correctly understand or produce natural language, such lexical relations need to be specifically encoded in lexicons [6], [10], [1]. In this paper, we show how word-based lexicons can be enriched with automatically acquired lexical relations. We call this process *microcoding* the lexicon, since it corresponds to the addition of lexical associations in a regular lexicon. We are using our enriched lexicon for language generation.

Co-occurrence knowledge is particularly important for language generation, without it, awkward or incorrect sentences could be produced. In previous natural language work, co-occurrence knowledge was ignored or hand encoded. In contrast, we acquire it automatically from the analysis of large textual corpora. We describe the acquisition method based on EXTRACT [12], a co-occurrence compiler that retrieves lexical relations from the statistical analysis of a large corpus. We indicate how these lexical associations are entered in a word-based lexicon in a useful and coherent way for language generators. We then show how this information is used in COOK, a functional unification based language generator that correctly handles collocationally restricted sentences. Whenever possible, we use examples taken from the bank and stock market domains.

The Problem: Co-occurrence Knowledge and Generation Lexicons

The fact that people prefer saying *"drink a strong tea"* to *"a powerful tea"*, and prefer saying *"drive a powerful car"* to *"a strong car"* cannot be accounted for on purely syntactic or semantic grounds. Consider the following example sentences. The lexically incorrect sentences are marked by a $\star$.

(1) $\star$ *"John offered Mary a hint"* (2) *"John gave Mary a hint"*
(3) $\star$ *"Mary perpetrated suicide"* (4) *"Mary committed suicide"*

In these sentences, *"hint"* and *"suicide"* co-occur with the verbs *"give"* and *"commit"* rather than with other synonymous verbs, the sentences are *collocationally restricted*. This kind of lexical behavior is usually unpredictable. For example, *"murder"*, which is closely related to *"suicide"* in terms of both meaning and syntax, co-occurs both with *"perpetrate"* and *"commit"*, whereas *"suicide"* only co-occurs with *"commit"*. Without the knowledge of such behaviors a language generator cannot correctly produce sentences such as (2) and (4).

Encoding lexical relations in the lexicon provides a language generator with the information necessary for handling many lexical decisions that were previously ignored. Moreover, it allows for simpler input since the presence of one of the two words in a given situation requires the presence of the other, (like *"suicide"* requires *"commit"*) [11].

The main reason why co-occurrence knowledge has generally been ignored in language generation is probably because of the inadequacy of the lexicons used; neither word-based nor phrasal lexicons are well suited for its economic inclusion. In phrasal lexicons, the basic entry is not a lexeme but a phraseme, *i.e.* a template for a whole syntactic structure. Phrasal entries can be seen as lexical associations *wired*, or microcoded in the lexicon. Although such wiring is questionable for semantically transparent constructs, it is necessary when dealing with semantically opaque[1] ones. For this reason, word-based lexicons cannot properly handle collocations. There are

[1] English substructures can be classified according to their opacity [3]. A structure is transparent if its meaning can be successfully divided into the meaning of its constituents, the structure is said opaque. Idioms provide natural examples of opaque structures, *"to have your cake and eat it"* very seldom literally means *"to have"* *"your cake"* *"and eat it"*. On the contrary, *"John opens the door"* is a transparent structure. Collocations, although somehow transparent (*"to commit suicide"*) offer a varying degree of transparency.

however prohibitive drawbacks with the use of phrasal lexicons in large applications. First, encoding transparent structures in phrasal entries is not economic since phrasal entries are polynomially more numerous than word entries. Second and more important, the information they contain is generally provided manually by the lexicon builders [8]. Such a manual processing is painstaking and hardly reaches a sufficient coverage. A way around these problems is to have a team of lexicographers provide information on combinational properties of English [4]. However, there is no such extensive linguistic expertise of English.

Here, we enrich word-based lexicons with automatically acquired lexical relations. Our acquisition program, EXTRACT decides on what must be microcoded in the lexicon. This allows tackling the issue of lexicon wiring without suffering from the problems of phrasal lexicons.

Acquiring Lexical Relations from Corpora

The acquisition of information from large textual corpora has already addressed in the past [2], but with different interest. Choueka was more interested in retrieving frequently used idiomatic expressions than idiosyncratic collocations. An idiosyncratic collocation is reflected in the language by a correlation of common appearance[2] of several items. EXTRACT identifies lexical relations in a large sample of natural language data by making statistical observations. As a first step we only retrieve noun-noun, noun-verb and noun-adjective lexical relations. For example, noun-noun combinations would include, *"stock market"*, *"credit card"*, etc. Noun-verb combinations would include, *"to charge a card"*, *"to write a check"*, *"the market plummeted"*, etc. Noun-adjective combinations would include, *"a high rate"*, *"a fast turnaround"*, *"a bounced check"*, etc. Such collocations represent meaningful lexical entities in the considered domain, and require specific entries in the lexicon.

EXTRACT takes as input a corpus and a dictionary specifying only part of speech. It produces a list of tuples $(w_1, w_2, \textit{cook-info})$, where (w_1, w_2) is a lexical relation between two open-class words (w_1 and w_2), and *cook-info* is a set of statistical figures representing the lexical relation within the distribution of words collocating with w_1. For example, *cook-info* contains an evaluation of the correlation factor of w_1 and w_2, and additional information on their relative positions in the corpus. *Cook-info* is used to filter out irrelevant associations and to retrieve global syntactic information. A more detailed description of EXTRACT along with some results can be found in [12].

A first version of EXTRACT has been tested on a 300,000 words corpus taken from the UNIX Usenet and on a 2,000,000 words corpus taken from The Jerusalem Post archives. It has retrieved more than three hundred lexical relations such as, *"to clamp"* with *"curfew"*, as in *"the authorities clamped a curfew ..."* *"To plant"* with *"bomb"*, as in *"the terrorists planted a bomb in ..."* *"Violent"* with *"clash"*, as in *"there has been a violent clash today ...,"* etc. An experiment using EXTRACT for software reuse is described in [9]. We are currently working on corpora in the domains of stock market and bank reports, in order to retrieve domain dependent collocations.

How the Lexicon is Used while Generating

Following Halliday [6], we include collocations in the lexicon such that they can easily be used from within the grammar. Lexical relations are lexical associations that are triggered in a given context by the grammar. As a demonstration, we have implemented COOK, a generator using a microcoded lexicon and handling these constraints in the grammar. COOK works on a simple banking domain, it uses FUF [5] a functional formalism based on functional unification grammars [7]. COOK correctly handles collocationally restricted sentences while using a simplified input structure. Syntactic and lexical constraints are encoded as functional descriptions, the grammar handling their interaction. Let us briefly describe some results obtained by COOK.

The inputs used here all describe a transfer of money, they have roles such as PREDICATE; ACTOR the person initiating the transfer; AMOUNT the amount of money transferred; FORM the actual form of the transfer; OBJECT what is being exchanged for the money (if any) and TO or FROM describing where the money goes or comes from. The verb is deduced using this information as can be seen below. Sentences (1), (2), and (3) have been generated using the logical forms (lf1), (lf2), and (lf3). The verb was not specified in the logical input but was deduced by COOK using collocational and semantic knowledge.

(lf1) ((PREDICATE DEBIT)(ACTOR ((LEX john)))(AMOUNT ((LEX $100)))(FORM ((LEX interest))))
(lf2) ((PREDICATE CREDIT)(ACTOR ((LEX john)))(AMOUNT ((LEX $100)))(FORM ((LEX interest))))
(lf3) ((PREDICATE DEBIT)(ACTOR ((LEX john)))(AMOUNT ((LEX $100))(OBJECT ((LEX insurance))))
(1) *"John earns $100 in interest."*

[2] That is, within a single syntactic unit *e.g.*, noun phrase, verb phrase, etc.

(2) *"John pays $100 in interest."*
(3 *"John takes out insurance for $100."*
In sentences (1), (2) and (3), the lexical relations *"interest-earn"*, *"interest-pay"* and *"insurance-take-out"* have been used. Note the influence of the predicate and of the logical roles (OBJECT, FORM) used. In some cases the lexical choice cannot be done effectively without a more complex interaction among constraints. We have examined cases involving the interactions of syntactic, semantic and argumentative constraints.

Conclusion

Co-occurrence knowledge is necessary for language generation. It provides useful indications for choosing lexical items and is indispensable for generating collocationally restricted sentences. We automatically acquire lexical relations from the statistical analysis of a large training corpus and use this information to enrich word-based lexicons for language generation.

We have implemented the above ideas in EXTRACT, a co-occurrence compiler that has been tested on several corpora. A first version of COOK, has also been implemented in a simplified banking domain. In the near future we plan on running EXTRACT on a corpus containing stock market reports, and to use the acquired information in an extended version of COOK. The ultimate goal of this approach is to make use of a training corpus in any given domain in order to automatically help microcode a lexicon for language generation.

Acknowledgments

Many thanks are due to Kathy McKeown for her advice on both the paper and the research. The research reported was partially supported by DARPA grant N00039-84-C-0165 and NSF grant IRT-84-51438. I am grateful to the The Jerusalem Post for graciously giving me access to their archive and to Yaacov Choueka and Joel Walters at Bar Ilan University, Ramat Gan, Israel. for providing me with a rich and pleasant environment during my stay there.

References

[1] M. Benson, E. Benson, R. Ilson, *The BBI combinatory dictionary of English: A guide to word combinations.* Amsterdam: John Benjamin, 1986.

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

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

[4] L. Danlos, *The linguistic basis of text generation.* Series. Studies in Natural Language Processing, Cambridge University Press, 1986.

[5] M. Elhadad, *Extended Functional Unification Programmars*, Columbia University, Department of Computer Science, technical report CUCS-420-89, 1989.

[6] 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.

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

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

[9] Y.S Maarek & F.A. Smadja, *Full Text Indexing Based on Lexical Relations, An Application: Software Libraries.* To appear in the Proceedings of SIGIR'89, 12th International Conference on Research and Development in Information Retrieval, Cambridge, MA, June 1989

[10] R. Mackin, *On collocations: words shall be known by the company they keep.* In *In honour of A.S. Hornby.* Oxford University Press, Oxford, England, 1978.

[11] F.A. Smadja, *Dictionaries for Language Generation Accounting for Co-occurrence knowledge*, Columbia University, Department of Computer Science, New York 10027, Technical report number CUCS-418-89.

[12] F.A. Smadja, *Lexical Co-occurrence: The Missing link.* To appear in the Journal of the Association for Literary and Linguistic computing, 1989.