Might a semantic lexicon support hypertextual authoring?

Roberto Basili, Fabrizio Grisoli, Maria Teresa Pazienza

Dipartimento di Ingegneria Elettronica,
Universita' di Roma, Tor Vergata
Via della Ricerca Scientifica, 00133 Roma (ITALY)
e-mail: {rbas,pazienza}@tovvx1.ccd.utovrm.it

Abstract

It is common opinion that current hypertextual systems do not allow to express objectively the information content of documents, but only the view of the "author". The hyperlink building requires an heavy and highly specialised human intervention: this task is very expensive whenever possible!
A different approach, based on NLP methodologies, aiming at automatizing the development of an hypertext, is hereafter proposed. Anchorage points are inferred both from content and structure of documents. A semantic lexicon based on conceptual graph structures is used to guide text understanding. Contextual roles are introduced to model domain specific concepts relevant to the navigation. An off-line activation of useful links has been defined according to explicit user specifications. A simple declarative language (HyDeL) for the definition of such links is available to the user to create his own views on the document base. HERMES is a prototype system implementing our approach. The paper discusses the semantic processing of a document base and highlights the performance of different hypertextual systems derived by HERMES over different languages and knowledge domains.

Introduction

An intuitive representation of an *hypertext* is depicted as a directed graph, in which nodes are *documents*, and arcs are *links*. *Browsing* an hypertext means to move from a document (source node) to another (destination node) through a link. Specific words, named *anchors*, are considered more important than others, as they are source or destination of a link.
Many different criteria have been used to describe existing hypertextual systems. According to (Frisse, Cousin 92) three different "levels" in modelling an hypertext should be underlined: *physical, logical* and *browsing semantics levels.*
At *physical level,* the model addresses the data organization on the storage devices. At *logical level,* relationships among data are described. Finally, the *browsing semantics oriented models* are used to describe human interaction modalities.
Main actors in hypertext systems are the **author** and the **user**: the author organizes the overall hypertextual structure defining the number and type of links between documents, thus imposing a particular navigation modality in the *document base*[1]. The user may browse the hypertext, following only the defined hypertextual structure.
Authoring is a major problem in hypertextual systems: the requested human effort explodes with the document base size. An incremental approach is often impossible either because the insertion of a new document requires a reprocessing of the whole document base, or the new document may introduce new perspectives. The management of very large document bases would suitably profit of an automatic approach to authoring.
Furthermore it is simply unrealistic that *the author knows what is of interest in any document, at any time.* Manual authoring thus bases on a static source of information modelled once; it should be desirable a more objective and exhaustive processing.
Our idea of hypertext is as follows: considering that documents have their own meaning, strictly related both to their content and structure, which is independent from the "author" point of view, we affirm that *an anchor is a property of the document,* and its existence does not depend by any (potential) link existence.
HERMES[2] (Hypertextual Effective Role-based Multilingual End user oriented System) is our proposed hypertext management system that embodies such ideas. HERMES uses an hypertextual framework where the author "makes explicit" information (on document structure and content), and the user creates his own hypertext schema by means of a simple definition language.

---

[1] With "document base" we refer simply to the set of documents, while we use the term "hyperbase" or "hypertext" when we refer to documents and links together.

[2] The implemented version of HERMES system actually runs on a SUN Sparc workstation. Logic programming modules, based on ProLog 3.1 by BIM, are integrated with C libraries, under the SunView and X-Windows environment. Italian and english corpora on different domains have been used to test HERMES.

1. The role of lexical knowledge in authoring.

An automatic hypertext generation may follow two approaches: fully automatic (i.e. pattern matching and/or statistic methods) or knowledge based authoring. They replicate a very famous dichotomy about symbolic vs. sub symbolic processing of information. As experimented in Information Retrieval, numerical methods are efficient and have a wide coverage, even though they lack expressive power and are not prevented from silence/noise drawbacks. But is reasonable to affirm that *word* (or phrase) *relevance in a text is independent either by its frequency or by text size* . It is our opinion that *black boxes* are not enough expressive to do explicit the meaning of what is implicit in the language. For example in the domain of Remote Sensing we acquired evidences that the information content of the word *satellite* should not be shared by sentences like " *the satellite flies over* ...." and "*Earth satellite (i.e. moon)* ...". Without any representation level other than string cooccurrences there is no way to capture such distinction. In sentences like " *Earth Observation systems* ...." the notion of satellitary platform is expressed in a different fashion. Capturing equivalence among words is a final target. *Scalability, robustness* and *efficiency* are important characteristics for any systems.

Scalability is a problem from a linguistic and computational point of view. To scale up to real sized document base it is necessary the use of shallow semantics to guide text understanding. Navigation points are in general 'local' properties of documents: they are usually triggered by limited contexts. A surface semantic approach, like the representation language used in ARIOSTO (Basili et al. 92 a,b 93 a,b), a system for corpus driven lexical acquisition, is sufficient to detect and extract such meaningful contexts.

The use of ARIOSTO will favour the portability and scalability of the HERMES methodology. As in our approach an anchor is a document property, existence of links should depend on the document set and on the user needs. Such distinction improves the flexibility of the hypertextual representation (see section 2) and the efficiency of the overall document processing.

Robustness of the linguistic processing is improved by the availability of a robust semantic lexicon generated by ARIOSTO within the corpus sublanguage. As information local to sentences is soundly detected and extracted by the Shallow Syntactic Analizer (SSA) processor of ARIOSTO, it has been helpfully used in HERMES for the basic text processing. Morphologic knowledge required by SSA is related to POS tagging. The available automatic tools, POS taggers (Church 88), or on-line dictionaries favour the portability of this special purpose syntax throughout different sublanguages. Syntax-based processing has also allowed a unified processing of different domains, and languages. Rule-based processing favours a common representation for texts of different structures, style and prose.

The use of acquisition tools is also helpful in improving the *efficiency* of the NLP preprocessing. The availability of a semantic lexicon and of coarse grained selectional restrictions allows disambiguation at syntactic as well as semantic levels. This implies a reduced need of memory. The efficiency of the overall hypertextual strategy is good as NLP is not an on-line activity, but it is performed only once over a text. When the meaning of a document has been stored, many successive processing steps can be activated (i.e. anchors detection and extraction, activation of links during the integration of the document in a (existing) document base, meaningful link visualization during the navigation phase). Moreover different hypertexts may be defined starting from the same semantic representation of the document base.

1.1 Automatic authoring and NL understanding.

The core notion used to capture parts of documents relevant to the navigation is the context of anchor. Anchors are a document feature and they are "possible" hints for hypertextual navigation, independently by the existence of incident links. This idea supports an incremental approach to hyperbase generation. In fact document content is to be recognised just once during the linguistic preprocessing: *updating the hyperbase* requires only the comparison of the new one with several document representations.

The main steps for the creation of an hypertext are :

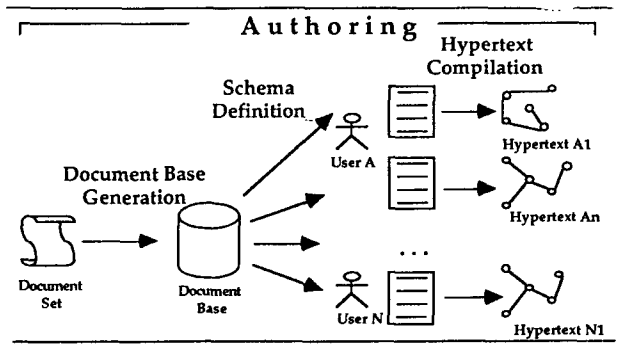
[1] **Document base generation**, rewriting of documents in a structured representation doing explicit meaningful components, i.e:
   a. structure *processing* (i.e. identification of textual vs. non textual parts ...),
   b. semantic *processing* (i.e. syntactic processing, detection and extraction of meaningful anchor points)

[2] **Schema Definition**, that allows end users to intentionally define, by a definition language, the nature and constraints of the links of interest for his own hypertext.

[3] **Hypertext Compilation**, that performs the generation of all those links matching the Schema Definition specifications.

Whenever a new document is available steps [1] ad [3] are applied to it. Steps [1] and [2] are asynchronous, several [2] phases can follow an unique [1]. This modularity allows to implement

different hypertexts as 'views' on the same document base. Fig. 1 graphically represents an overview of system functionalities.
In such an approach it is evident that the roles of "author" and "user" are conceptually different from traditional ones.



- Fig. 1: HERMES functionalities -

Step [1] poses strong requirements to the linguistic knowledge available to the system. ARIOSTO supports such a phase. Anchor points are induced from the raw text by means of a local semantic processing aiming at capturing sentences, propositions or phrases of interest. Whenever a shallow semantic classification of the content words is available, and a core set of conceptual relations allows the interpretation of syntactic structures in the sublanguage, parts of documents relevant for their content can be detected.
To look at hypertextual navigation as at a privileged search form as well as at a full text information retrieval, text understanding is a main activity. The problem is: what is an anchor according to its use during navigation? An anchor is related to a topic (expressed, mentioned or developed, within a document) that is relevant in the corresponding knowledge domain. Examples of such meaningful topics may be *satellite* or *IOL* in sentences like

"Earth observation satellites for future applications employ high resolution sensors (...)"
"Especially for the transmission of such high data rates on IOLs and ISLs optical systems are (...)"

Targets of queries to a bibliographic database are examples of information of interest for a user in the related domain. User interest focuses on (a limited set of) classes of arguments relevant to the knowledge domain. Symptoms, Medicines or Pathologies are qualitative examples of such classes within a (possible) medical domain. We will see other examples connected to Remote Sensing hereafter. Existence and properties of concepts like Symptoms vs. Pathologies may be ruled by some meta level knowledge able to guide retrieval as well as text comprehension. "*Cough*" for example may be a Pathology (when aspecific) but more often it is a Symptom of a more complex disease. Detecting such differences for a word requires the recognition of some aspects of their 'meaning' in the context. These different behaviours of a word are defined as *contextual roles* in our system, and will be used for an intelligent retrieval. The contextual roles relevant for the document base will be derived by an analysis of the related sublanguage. We shall outline the way contextual roles are detected in texts and expressed by suitable semantic primitives called *representative types* in section 1.2. Roles and lexical expectations are also central to other text processing systems, like SCISOR (Rau, Jacobs 88).
Automatic authoring, as many other text oriented disciplines, needs for a specialised, word oriented component of the knowledge, i.e. a lexicon, describing word meaning within the text. ARIOSTO automatizes the generation of one such lexicon related to the analysed corpus.
The core of lexical knowledge acquired by ARIOSTO is word association data augmented with syntactic and semantic markers. In ARIOSTO reliability is improved by statistical processing applied to syntactic information (that we call *elementary syntactic links* (esl)).
Statistically relevant associations between words with the same semantic tag are markers of typical semantic relationships between corresponding classes (or conceptual categories). The discovery of such selectional restrictions throughout different corpora is relevant to the engineering of the required specific lexicon.

1.2 NL Processing of documents in HERMES.

The need for a semantic interpreter is related to text intrinsic ambiguity at syntactic as well as at semantic levels. Main problems are related to multiple sense words, syntactic ambiguity (e.g. PP referents), long-range relationships, focus. As we have previously outlined ARIOSTO provides a specific domain lexicon based on a set of semantic classification of lemmata and a catalogue of conceptual relations: the former is a 'flat' type hierarchy (Sowa 84) that aims at improving word senses disambiguation, while the latter provides a set of canonical graphs (Sowa 84,89), that guides syntactic disambiguation as well as interpretation. In ARIOSTO this declarative form of knowledge is augmented by statistical figures expressing numerical preference factors.
Elementary syntactic links (*esl*) are interpreted by the lexicon of cooccurring words. For example a NL segment like
(1) " *the temperature measured by the sensor ...*"
that originates the *esl*
(2) `V_P_N(measure,by,sensor)`
may be interpreted by the following conceptual relation

[Act:measure]-(INSTRUMENT)->[INSTRUMENTALITY[3]:sensor],
The relation
[Act:measure] -(INSTRUMENT)-> [INSTRUMENTALITY:*]
in the lexicon of *measure* is here triggered by (2).
Conceptual graphs join (Sowa 84) is used to compose conceptual relations provided by *esl*'s derived by the same sentences. Local meanings of content words originate complex conceptual graph structures expressing semantic analysis of the related phrases. Contextual roles are domain dependent conceptual graphs *schemata* (Sowa 84) that are possibly filled in by relevant sentences and passages of the documents. Anchor derivation is realized by triggering schemata in an expectation driven fashion. An example of schema defined in the Remote Sensing domain is the following:

(3) Remote_Sensing_Machinery(x) iff
[INSTRUMENTALITY:*x]
-(OBJECT)->[LOCATION:*y]
<-(INSTRUMENT)-[Act:*z]

(3) provides the assignment of the role Remote_Sensing_Machinery to *system* in "*Earth observation systems* ..." or to *Sar* in " ... *ERS-1 Sar enlights land surface with a resolution of 30 m*".
Word sense ambiguity is partially solved by the word classification. Syntactic ambiguity is approached by the use of the lexicon of canonical graphs. The remaining genuine ambiguities are solved by maximizing the probabilities of the competing canonical relations of the lexicon.
After the shallow syntactic processing of texts, the semantic interpretation produces the list of relevant words, tagged by their related contextual roles in the document. The couple <word,contextual role> is a potential anchor in HERMES. CoDHIR system (Marega, Pazienza 94) uses the contextual roles for semantic driven Information Retrieval in the Remote Sensing domain.

2. System architecture

In this section we describe the HERMES functional architecture. Implementative details may be found in (Grisoli 94). Main problems in hypertextual authoring are: real text understanding, objectivity and incrementality. Correspondingly we have defined in HERMES three different activities: Document Preprocessing, Schema Definition and Hypertext Compilation. The Schema Definition allows the end user to get a "self tailored system" as well as a "view" of the document base. While document preprocessing captures document semantics, hypertext compilation performs the effective generation of the hyperbase.

[3] INSTRUMENTALITY is a semantic class of the 'flat' hierarchy semi-automatically derived from WORDNET .

2.1 Document base generation.

We identified in document preprocessing two different phases: full text and structural processing. In the document structure processing basic document components are recognised, while the semantic information of interest (i.e. contextual roles) is extracted from the textual parts during the full text processing.
In a document base several types of documents, identified by an explicit structure, may be collected. For example, the textual component of an abstract has an implicit structure. Scientific papers show generally a more explicit structure, e.g. title, authors, sections, ... Recently several standards for documents interchange have been defined to impose an overall fixed structure to documents. The structure brings part of the information related to a document. Hypertextual systems are able to manage such kind of information (Essence 92).
In HERMES this aspect has been exploited too: explicit CF grammars have been defined to recognise different document types.

2.2 Hypertext Schema Definition

Hypertext schema definition is based on an intentional description of: document structure, rules for defining collections, rules for defining links. HERMES provides the user with a definition language called HyDeL (Hypertext Definition Language) (Grisoli 94), to directly define his own hypertexts. HyDeL is a declarative language whose syntax and semantics are quite simple.
In the *schema definition* the user declares the types of documents that wants be inserted in the document space. HyDeL allows the definition of a corresponding set of *document types* . Basic elements of a document type are *attributes*. Attributes may be simple as strings, numbers and text. Complex attributes may be introduced as a combination of attributes. For example we can say that a *scientific paper* is a document with a list of authors, a title, an abstract, a list of sections, and references. An author can be thought of as a string or, alternatively, as a complex attribute (noun, e-mail, address, and so on) according to the user wishes. Full text components are subject to a different representation, in terms of the list of extracted anchors.
A *collection* is a set of documents that share some properties. Properties can be expressed by declarative rules on document attributes. For example we can define a collection of scientific papers that have "title beginning with a" or "including an anchor with a fixed role" etc... In this way hypertext

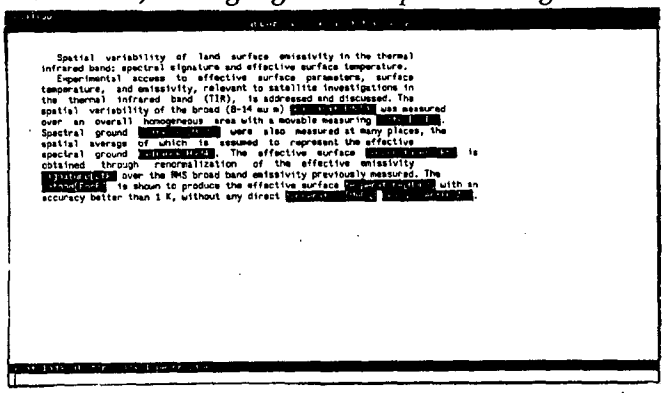
definitions cluster together documents according to some application oriented criteria.
Links are also defined by declarative rules. They express contraints on couples of anchors belonging to different documents. Whenever constraints are satisfied the corresponding links are activated. We stress that a rule for link definition in HyDeL is independent either by the user knowledge on the document content or by its activation. Rules in HyDeL provide a sort of query language on anchors. Declarativity here may be used to deductively generate links of semantic 'flavour'.
Links activation and collection generation proceed at the hypertext compilation time (section 2.3).

2.3 Hypertext compilation

Compilation consists of the generation of all the links and collections as they have been defined in the hypertext schema. This activity is performed document by document: creating an hypertext, thus, consists of updating an empty hypertext. Incrementality follows consequently. Creating/updating activities are realized in a completely automatic way.
A document can be thought of as a point in a vector space. Components are here related to the document semantic content, i.e. its anchors: a previous research on this model has been carried out in (Marega, Pazienza 94). A distance metrics has been defined in such a semantic space. To integrate an incoming document means to appropriately collocate it in this space as suggested by its semantic representation. Contextual as well as structural information are used in this phase. Collections are clusters of points in this space. Links connect documents, i.e. distinct points, even far in the space.

3 Linguistic evidences in HERMES

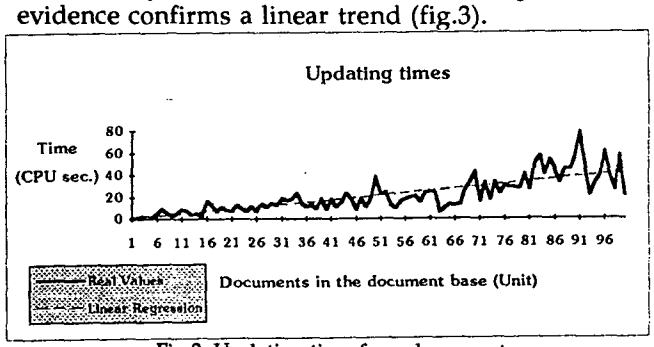
Figure 2 shows the screendump of an excerpt of an English hypertext. Anchors (i.e. words with their role labels) are highlighted. Simple browsing



- Fig. 2: HERMES: hypertext over Remote Sensing documents[4] -

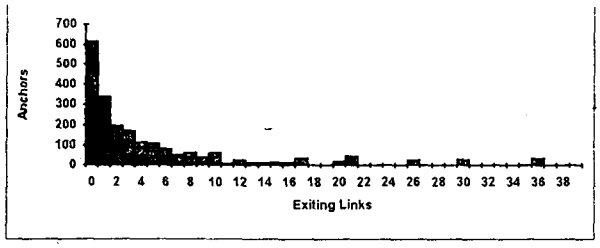
[4] Legenda of contextual role label in figure 4.

operations (i.e. index, history or backtracking) are available. The current version has been also implemented under the X-Windows environment. Parallel navigation sessions on the same document base are allowed.
In order to appreciate the advantages of automatic generation and upgrade activities, details on system performance may be of interest. Hypertext compilation, in fact, provides important information on the linguistic processing to be derived. The following analysis is based on the hypertext developed for documents (abstracts and D.I.F.) on Remote Sensing (210 documents). 10743 anchors (of 133 different words), and 5908 links have been globally derived. Fig. 3 describes the time needs during the compilation phase. We estimated a linear trend in time because the updating activity has a complexity O(n) [5], where n is the number of document yet in the document base. Experimental evidence confirms a linear trend (fig.3).



- Fig.3: Updating time for a document -

Note the difference from the linear regression (dotted line). This is a clear marker of relevant difference in document *information density*. This difference is much more evident when comparing abstracts with DIF (the last documents are composed by a large number of descriptors and a textual component used as a comment). We defined an information density score D as the ratio: number of anchors / document length (in byte). DIF and abstracts are very different in length: the average length value is 1022 bytes for abstracts and 6691 bytes for DIF documents.



- Fig.4: Exiting links versus anchors -

MeP Measured Parameter, ExI Experimental Instrument, StP Studied Parameter, Foc Focus, ExA Experimental Activity.
[5] To insert a document means to compare the new document with previously ones checking for link estabilishing.

Despite of this we obtained an average value D of 129 for abstracts and 341 for DIF.
Experimental evidence confirms abstracts are much more information dense than DIF.
In figure 4 a link versus anchor plot has been shown. This graph shows the link density of anchors. In referred prototype, running on a Remote Sensing domain, there are more than 300 anchors triggering only 1 link, while only less than 100 with 6 links have been activated. Note that there are more than 600 anchors than have no links leaving from them thus showing peculiar concepts for a document and irrelevant in the domain sublanguage (we call them *inactive anchors*, later insertion of a document may activate them). A monotonic decreasing trend on the number of the leaving links is evident. Isolated peaks on higher values of links number are evident. Those peaks are related to very domain dependent anchors, as for example, the couples <method-Focus>, <information-Kind of processed Information>, <information-Focus>, <image-Focus>, <result-Focus>, <image-Kind of processed information>. These anchors are present in most of the documents. Thus they represent common topics in the related domain and may be considered as concepts extremely important in the sublanguage. As descriptors of the underlying knowledge domain, the more frequently activated anchors are very information. One more evaluation provided by this plot is the evidence of computational validity of our approach. In fact the quickly decreasing curve shows that a very large amount of documents generates only a few links thus avoiding an exponential explosion of physical connections. We stress such a result because a completely automated production of anchors (as provided by our system) could have generated an unforeseen amount of links.

4. Concluding remarks

In this paper we have proposed a new approach to hypertexts based on a NLP methodology. Moreover a general description of HERMES, an hypertextual system allowing automatic authoring, has been provided. The growth in use and dimension of hypertextual systems, makes automatic authoring a must. In this activity, the efficacy of semantics, acquired by lexical acquisition tools (i.e. ARIOSTO), has been stressed. The semantic description guided by the lexicon is worth expressive for automatic authoring. As a deep text understanding is not required, semantic interpretation is feasible and cost-effective. A conceptual rather then just structural representation of documents suggests semantic rules for anchors detection and links activation.
The proposed method enhances the figure of the user, as opposed to the author. User becomes the main actor in the hypertextual management, as he can browse inside the space of documents with confidence. The hypertextual space is generated according to his suggestions, as provided in the definition schema. The author suggests only how to semantically represent documents.
As concluding remarks, the system evaluation has produced, as an unforeseen side-effect, some important linguistic evidences about the underlying sublanguages. This relates to HERMES capability of automatically deriving meaningful anchors. HERMES portability as well as low resource requirements are also improved by the use of lexical acquisition tools.

References

R. Basili, M.T. Pazienza, P. Velardi. 1992. A shallow syntactic analyzer to extract word associations from corpora, Literary and Linguistic Computing, vol. 7, n. 2, 114-124.
R. Basili, M.T. Pazienza, P. Velardi. 1992. "Computational Lexicon: the Neat Examples and the Odd Exemplars", Proc. of 3rd Conf. on Applied NLP.
R. Basili, M.T. Pazienza, P. Velardi. 1993. Acquisition of selectional patterns, Journal of Machine Translation, 8:175-201.
R. Basili, M.T. Pazienza, P. Velardi. 1993. What can be Learned from Raw Text? An Integrated Tool for the Acquisition of Case Rules, Taxonomic Relations and Disambiguation Criteria, Journal of Machine Traslation, 8:147-173.
K. Church. 1988. A stochastic parts program and noun phrase parser for unrestricted text, in Proc. of 2nd Conf. on Applied Natural Language Processing, Morristown, February 1988.
Essence Intelligent Document Analysis System. 1992. Technical Documentation, Toshiba, Australia.
M.F. Friesse, S.B. Cousin. 1992. Models for Hypertext. Journal od the American Society for Information Science, 43(2).
F. Grisoli. 1994. Definizione e realizzazione di sistemi ipertestuali mediante tecniche di NLP, Thesis, Università di Roma "Tor Vergata".
R. Marega, M.T. Pazienza. 1994. Co.D.H.I.R.: an Information Retrival System Based on Semantic Document Representation, Journal of Information Science, vol. 6.
L. F. Rau, P. S. Jacobs. 1988. Integrating Top-Down and Bottom-Up strategies in a Text Processing system, Proc. of the Second Applied Natural Language Processing, MorrisTown, NJ.
J. F. Sowa. 1984. Conceptual Structures: Information Processing in Mind and Machine, Addison-Wesley.
J. F. Sowa. 1988. Using a Lexicon of Canonical Graphs in a Semantic Interpreter, in Relational Models of the Lexicon, M.Evens, Ed., Cambridge University Press, 1988.