Faculty of Mathematical Sciences

University of Twente
The Netherlands

P.O. Box 217
7500 AE Enschede
The Netherlands
Phone: +31-53-4893400
Fax: +31-53-4893114
Email: memo@math.utwente.nl
www.math.utwente.nl/publications

Memorandum No. 1657

Information extraction

L. Zhang[1] and C. Hoede

November, 2002

ISSN 0169-2690

[1] Department of Computer Sciences, Northwest University, Xi'an, Shaanxi 710079, China

INFORMATION EXTRACTION

L. ZHANG

Department of Computer Sciences

Northwest University

Xi'an, Shaanxi 710069, China

C. HOEDE

Department of Applied Mathematics

University of Twente

P.O.Box 217

7500AE Enschede, The Netherlands

Abstract

In this paper we present a new approach to extract relevant information by knowledge graphs from natural language text. We give a multiple level model based on knowledge graphs for describing template information, and investigate the concept of partial structural parsing. Moreover, we point out that expansion of concepts plays an important role in thinking, so we study the expansion of knowledge graphs to use context information for reasoning and merging of templates.

Key Words: information extraction, partial structural parsing, knowledge graph expansion, template

AMS Subject Classifications: 05C99, 68F99

1 Introduction

Over the last decade there has been a growing interest in developing systems for *information extraction* (*IE*). In a broad view, information extraction refers to any process that creates structured representation of selected information drawn from one or more texts. Usually this process involves the identification of instances of a class of *events* or relationships and the extraction of the relevant arguments or relationships in a natural language text. The output of the extraction process, although varying in every case, is finally transformed to the content of some type of database.

An enormous amount of information exists only in natural language form. The idea of reducing the information in a document to a tabular structure goes back to the early days of NLP applications [DeJong, 1979, Schank & Abelson, 1977, Sager, 1987]. However, the specific notion of information extraction was relatively new in the series of *Message Understanding Conferences* (*MUCs*). As a core of language technology, IE systems represent the need and ability to manipulate and analyze information automatically, by integrating a variety of natural language processing technologies. IE technology has not yet reached the market, but it was thought to be of great significance to information end-user industries of all kinds as, for example, finance companies, banks, publishers and other document-dependent managing agencies.

To be a trend of "understanding" by extracting information from texts, IE technology should not be confused with the more mature technology of Information Retrieval (IR) that selects a relevant subset of documents from a large volume set by query. IE extracts information from the actual text of a document. Any application of IE is usually preceded by an IR phase. Information Extraction is a more limited task than "full text understanding". In full text understanding, we expect the representation of *all* the information in a text in an explicit fashion. In information extraction, as a more focused and well-defined task, we restrict to the semantic range of the output: the relations we will represent, and the allowable fillers for each slot of a relation. For example, in the domain of terrorism, as the task given in the MUC-4 evaluation (1991), an IE system would extract the date, location, perpetrators, victims, targets and type of attack (bombing, arson, etc.). Since many phrases and even entire sentences can be ignored if they are not relevant to the domain, the IE process is computationally less expensive than in-depth natural language processing. In the last decade, IE has achieved notable success [MUC-3, MUC-4, MUC-5, MUC-6, MUC-7, Grishman & Sundheim, 1996, Cowie & Lehnert, 1996].

It should be noted that IE is not a wholly isolated information technology. For example, *MT (Machine Translation)* and IE are just two ways of producing information in applications and can be combined in different ways. One could translate a document and then extract information from the result or change the order of these procedures. Moreover, a simpler MT system might only be adequate to translate the contents of templates that resulted in an IE process. This means that the product of an IE system, i.e. the filled template, can be managed either as a compressed text itself, or as a form of database (with the fillers of the template slots corresponding to database fields).

2 The State of the Art of IE

Generally, the process of an IE system includes two major parts: (1) Extraction of the individual "facts" from the text through local text analysis, and (2) Integration of these facts to generate new facts through inference. Finally the pertinent facts are represented and translated into the required output format.

According to the terminology established by the MUC, a *scenario* is specific to particular events or relations to be extracted, and a *template* refers to the final tabular output format of the IE process. Recent research on IE was stimulated in large part by the MUC evaluations. Five separate component tasks, which illustrate the main functional capabilities of current IE systems, were specified by recent MUC evaluation (MUC-7).

(1) *Name Entity recognition* requires the recognition and classification of named entities such as organizations, persons, locations, dates and monetary amounts.

(2) *Coreference resolution* requires the identification of expressions in the text that refer to the same object, set or activity. These include variant forms of name expression, definite noun phrases and their antecedents and pronouns and their antecedents.

(3) *Template Element filling* requires the filling of small scale templates (slot-filler structures) for specified classes of entities in the text, such as organizations, persons, certain artifacts, and their locations, with slots such as name (plus name variants), description as supplied in the text, and subtype.

(4) *Template Relation filling* requires filling a two-slot template representing a binary relation with pointers to template elements standing in the relation.

(5) *Scenario Template filling* requires the detection of relations between template elements as participants in a particular type of event, or scenario, and the construction of an object-oriented structure recording the entities and various details of the relation [Humphreys, 2000].

Practically, IE development relies on recent advances in empirical NLP techniques. Many relatively independent modules within some general knowledge-based AI program have achieved significant success for a range of linguistic tasks, such as word sense tagging, syntactic parsing, sentence alignment and so on. Currently the most successful systems use a finite automatic approach, with patterns being derived from training data and corpora, or specified by computational linguistics. Recent research [Church *et al.*, 1996] has also shown that a number of quite independent modules of analysis by learning and statistical methods can be built up independently from data, rather than coming from either intuition or some dependence on other parts of a linguistic theory.

There have been IE systems developed in groups stimulated by the MUC, such as POETIC [Mellish *et al.*, 1992], MITRE [Aberdeen *et al.*, 1995], FASTUS [Appelt *et al.*, 1995], SRA [Krupka, 1995], UMASS [Fisher *et al.*, 1995], LASIE [Gaizauskas *et al.*, 1995], NYU Proteus system [Yangarber & Grishman, 1998)], with alternative features to the IE task by applying NLP techniques. Many developers of IE systems have opted for robust shallow processing approaches that do not employ a general framework for "knowledge representation". In other words, there may even be an attempt, without building a meaning representation of the overall text, nor representing and using world and domain knowledge in a general way to help in resolving ambiguities of attachment, word sense, quantifier scope, coreference, and so on. Such shallow approaches typically rely on collecting a large number of lexically triggered patterns for partial filling templates, domain-specific heuristics for merging partially filled templates to yield a final, maximally filled template, as exemplified in the systems of FASTUS [Appelt *et al.*, 1995] and the SRA and MITRE MUC-6 systems [Krupka, 1995, Aberdeen *et al.*, 1995]. However, there have been attempts to derive a richer meaning representation of the text with less task- and template-specific approaches, such as the discourse model and intermediate representation used in the design of the LASIE system [Gaizauskas *et al.*, 1995]. Such approaches were motivated by the belief that high level of precision in the IE task will not be achieved without attempting a deeper understanding of at least parts of the text. It was claimed that the MUC-6 evaluation showed that such an approach, with richer meaning representation, overall performed not worse than the shallow processing approaches [Cowie & Wilks, 2000].

A typical discussion about the NLP techniques used for IE is that of Hobbs [Hobbs, 1993]. According to Hobbs' paper, the functionalities shared by most systems alternatively include the following [Cowie & Wilks, 2000]:

1. a Text Zoner, which turns a text into a set of segments.
2. a Preprocessor, which turns a text into a sequence of sentences.
3. a Filter, which turns a sequence of sentences into a smaller set of sentences by filtering out irrelevant ones.
4. a Preparser, which takes a sequence of lexical items and tries to identify reliably determinable small-scale structures.
5. a Parser, which takes a set of lexical items (words and phrases) and gives a set of parse-tree fragments as output.
6. a Fragment Combiner, which attempts to combine parse-tree or logical-form fragments into a structure of the same type for the whole sentence.
7. a Semantic Interpreter, which generates semantic structures or logical forms from parse-tree fragments.
8. a Lexical Disambiguator, which indexes lexical items to one and only one lexical sense, or can be viewed as reducing the ambiguity of the predicates in the logical form fragments.
9. a Coreference Resolver, which identifies different descriptions of the same entity in different parts of a text.
10. a Template Generator, which fills the IE templates from the semantic structures.

It should be noted that there exist disputes on the practice of IE with the above organization. For example, module 8 could be performed early on lexical items, or later on semantic structures. Within the process under module 5, some people use a syntactic parser but the majority uses some form of corpus-derived finite-state patterns to represent the lexical sequences, which process would be called "semantic parsing" [Cowie *et al.*, 1993].

For an IE task, defining templates is difficult, which involves the selection of the information elements required, and the definition of their relationships. The definition consists of two parts: a syntactic description of the structure of the template (often given in a standard form known as BNF-Backus Naur Form), and a written description of the rules on filling the templates and instructions on determining the content of the slots. The actual structure of the templates used has varied from the flat record structure of MUC-4 to a more complex object oriented definition used for

MUC-5 and MUC-6. For example, a person object might contain name, title, age and an employer slot, which is a pointer to an organization object. Such newer object style templates make it easier to handle multiple entities which share one slot, as they group together the information related to each entity in the corresponding object. However, the readability in printed form suffers a lot, as much of it consists of pointers.

Although many IE systems have been proved effective for information extraction on limited domains, there are difficulties in construction of a large number of domain-specific patterns. Manual creation of patterns is time consuming and error prone, even for a small application domain. In the following parts, a new IE approach is presented, which is using a domain-independent model described by knowledge graphs. This method is to Extract information by Knowledge Graphs (KGExtract) from the natural language texts.

3 Overview of the Approach

The development of language engineering applications, information extraction (IE) in particular, has demonstrated a need for the full range of NLP and AI techniques, from syntactic part-of-speech tagging through to knowledge representation and reasoning. The task of information extraction can be seen as a problem of semantic matching between a user-defined template and a piece of information written in natural language. To this purpose, the structural parsing oriented semantic processing will be applied to IE, and we will show that such a new IE technique has considerable advantages in comparison to traditional approaches to information extraction from the texts. For example, the method presented here, which is to encode the input pieces of information and the filled template into knowledge graphs, is a kind of graphic representation and it is domain-independent.

With respect to the advantages of the approach, our main points are:

- a simple (but semantically rigorous) model;
- the possibility of semantic checks guided by the model;
- a domain-independent representation;
- an automatic pattern acquisition;

Let us now discuss an approach that is a process consisting of the following phases:

Lexicon and Morphology. In the procedure of extracting information from NL texts, the precise duties of the lexical and morphological processing depend on the language that is being processed. For example, Chinese, without orthographically distinguished word boundaries, will require that some word segmentation procedure will be applied, but English can skip this procedure.

The most important problem faced in this phase is to handle the proper names in a text. Because most extraction tasks require the recognition of persons, companies, government organizations, locations, etc.

In addition to name identification, this phase must tag word types to words. We have discussed 8 word types in English in [Hoede & Zhang, 2001b], such as noun, verb, adjective, pronoun, numeral, preposition, adverb and determiner. For each word type, we have created its syntactic word graph that represents the syntactic function of a word type. In Chinese we also chose 8 word types, but we replaced the "determiner" type by the "classifier" type.

Semantic chunk tagging. The role of this phase is to split the different sentences of the text into semantic chunks according to the chunk indicators, such as pairs of commas and/or period signs, auxiliary verbs, reference words, prepositions, "jumps", etc. Here no more details about chunk indicators will be repeated since we have discussed the problem in [Hoede & Zhang, 2001b].

Partial Structural Parsing. Some IE systems do not have any separate phase of syntactic analysis. Others attempt to build a complete parse of a sentence. Most systems fall in between, and build a series of parse fragments. In general, they only build structures about which they can be quite certain, either from syntactic or from semantic evidence. In our approach, a partial structural parsing method will be applied to every sentence of the input to build a series of semantic chunk knowledge graphs. We will then combine these knowledge graphs of chunks to derive the information that is to be extracted. The partial structural parsing can make the patterns, that will be mentioned in the next phase, to be created easily.

Domain-independent pattern creation. In order to extract information from texts, we have to have patterns representing entities and events occurring in the texts. Many IE systems use a pattern-matching approach, but the set of patterns has to be created for each target task or target domain. If we are using a "pattern-matching" method, most work will probably be focused on the development of the set of patterns. However, for different domains changes will also be needed to the semantic hierarchy, to the set of inference rules, and to the rules for creating the output templates.

To summarize, other methods are domain-dependent, but our approach is domain-independent. This is the most important improvement of our approach in comparison with other systems.

Pattern Merging. This is the main part of our approach. The procedure of merging patterns is actually to integrate two semantic chunk graphs into a bigger one. It can be repeated until the number of semantic chunk graphs becomes 1.

Template Generation. Once a text has been fully processed and a domain-independent representation has been derived, this representation can be used to generate template structures.

4 Description of KG-extraction

This section addresses the theoretical issues related to the design and use of such a method for information extraction. We present some basic principles, and we illustrate a preliminary proposal for a model developed according to such principles.

Definition 1 A *KG-extraction* is the mapping of unstructured natural language texts onto predefined, structured representations, or templates, which, when filled, represent an extract of key information from the original text, with special regard to a model based on knowledge graph theory.

4.1 Partial structural parsing

It is very important to realize that the role of parsing in an information extraction system is not to perform full text understanding, but to perform parsing on the relevant parts of the text. The shallow parsing techniques tend to be imprecise, although efficient and transportable, whereas the full parsing approaches tend to be very precise but not robust and efficient.

Our approach is to shift from a full structural parsing (which has been described in [Hoede & Zhang, 2001b]) to partial structural parsing. The partial structural parsing has the role not only of identifying syntactic structure but also of making the extracted information syntax independent "regularizing" or "standardizing" by constructing the semantic chunk graph. For more detail about regularizing, we refer to the example in Section 4.2.

Definition 2 *Partial structural parsing* is the mapping of a sentence that is in the input text onto a set of semantic chunk graphs of this sentence.

The goal of partial structural parsing is creating the scenario patterns of information to be extracted, not obtaining the full sentence graph. It is performed along almost the same phases as structural parsing, that is the mapping of a sentence on a semantic sentence graph, but for the last phase of structural parsing, which combines the various bigger semantic chunk graphs into a sentence graph.

4.2 An example of representing patterns with knowledge graphs: KG-Structure

In order to extract the information from text, we have to have patterns representing entities of interest in the application domain (e.g., company takeovers, management successions) and relations between such entities in the texts.

In many other current extraction systems most of the text analysis is performed by matching text against a set of patterns. If the pattern matches a segment of the text, the segment of the text is assigned a label, and one or possibly more associated features. These patterns are domain specific. For the example of “executive succession”, there will be such patterns as:

<person> retires as <position>, <person> is succeeded by <person>.

In another text about “joint venture”, there will be the following pattern:

<company> forms joint venture with <company>.

In general, each application of extraction will be related to a different scenario. Most work will probably be focused on the development of the set of patterns, because it is difficult and inconvenient to operate directly on the patterns. These patterns are varying as the system turns from domain to domain.

The approach described in this section is to represent such patterns by knowledge graphs. This provides a graphical representation of patterns (i.e., knowledge graphs), which is called KG-Structure. One can then operate conveniently on the knowledge graphs. The KG-Structure, which is domain-independent, aims at easing the burden of pattern creation.

In particular, different clause forms, such as active and passive forms, relative clause, reduced relatives, etc., are mapped onto essentially the same semantic structure (i.e. a semantic chunk graph). This regularization simplifies the scenario pattern creation.

For example, we do not need separate patterns for

Cars that are manufactured by GM.

... GM, which manufactures cars ...

... cars, which are manufactured by GM ...

... cars manufactured by GM ...

GM is to manufacture cars.

Cars are to be manufactured by GM.

GM is a car manufacturer.

etc.

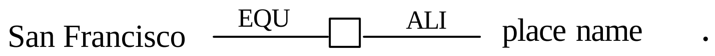
Although these various clauses have different syntactic structure, they have the same meaning. This is why there is only one pattern represented by a KG-Structure that is to be constructed as follows.

4.3 Named entity recognition

Named entities (like organization, person, location, and position) are very important in the information extraction task. It is necessary to recognize the proper names in the text and express them with a corresponding KG-structure.

Example

In the sentence "I want to go to San Francisco", we recognize "San Francisco" as a place name, its representation by a KG-structure is as follows:



Other named entities can be represented with a similar structure, and we do not list them one by one.

4.4 Automatic pattern acquisition

In this section, we will propose an almost automatic method to acquire patterns useful for IE from the text input. This method does not require humans to design complicated patterns. The basic idea involves the following:

- Semantic chunk taggings are performed for each sentence in the text.
- Semantic chunks are extracted to be the source of the patterns, which are represented with a KG-structure.
- The source KG-structures of the tagged sentence are integrated on basis of their similarity.
- Inferencing and merging are performed by knowledge graph operations.
- Templates are then filled through matching a KG-definition with a KG-structure.

One of the strengths of this approach is that the KG-structure, being domain-independent, based on partial structural parsing, supports the generation of the templates or summaries in a language different from that of the input texts.

At the initial stage, each tagged sentence is regarded to be a pattern consisting only of semantic chunks and named entities. The merging can be done by combining the most similar pair of patterns into one pattern.

There are several methods to define similarity of structures that can be found in the literature. In this paper we will not use any specific similarity measure.

4.5 Inference and merging

In many situations, partial information about an event may be spread over several sentences; this information needs to be combined before a template can be generated. In other cases, some of the information is only implicit, and needs to be made explicit through an inference process.

4.6 Generating templates

An IE system does not require full generation capabilities from the intermediate representation (the knowledge graph), and the task will be well-specified by a limited "domain model" rather than a full unrestricted "world model". This makes a generation feasible for IE, because it will not involve finding solutions to all the problems of such a KG-structure.

Definition 3 A *KG-definition* is a knowledge graph that expresses the semantics of a template relevant to the information extracted.

Once a text has been fully processed and a KG-structure pattern, of those aspects of it required for the IE task, has been created, this pattern can be used to fill template structures. These template structures will include pointers to the entries (each entry is a knowledge graph expression of a location slot) in the lexicon, which forms the KG-definition of the template. The KG-definitions of the templates are pre-defined and stored in the lexicon. Once a KG-definition of a template matches with a KG-structure of a pattern, a slot in the template will be filled up.

4.7 A worked out example

We will extract the information from the following text:

> "George Grorrick, 40 years old, president of the famous hotdog manufacturer Hupplewhite, was appointed CEO of Lafarge Corporation, one of the leading construction material companies in North America. He will be succeeded by Mr. John."

Input slots that stand for the information that people want to extract are:

EVENT

PERSON

AGE

OLD POSITION

NEW POSITION

NEW COMPANY

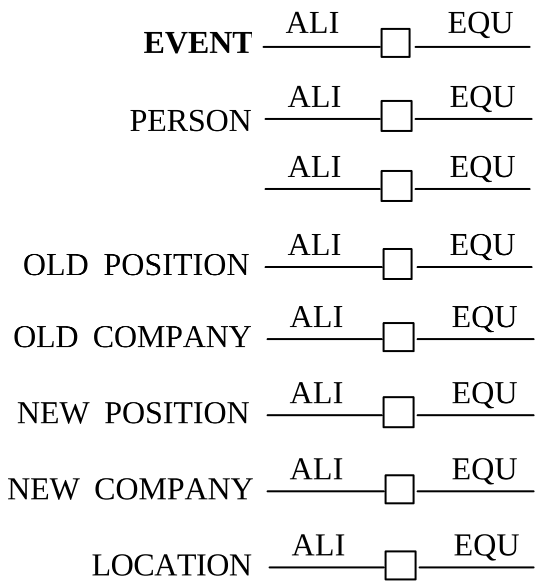
LOCATION**.**

As a result, we will fill each slot, and the output, that should be obtained, is shown as:

EVENT	Appointment
PERSON	George Grorrick
AGE	40
OLD POSITION	President
OLD COMPANY	Hupplewhite
NEW POSITION	CEO
NEW COMPANY	Lafarge Corporation
LOCATION	North America

.

The set of knowledge graphs corresponding to each slot is the following:



.

There are two sentences in our text, and we chunk each sentence step by step according to chunk indicators. There are 4 types of indicator, which have been mentioned in [Hoede & Zhang, 2001b], that are used in this example. Besides them, we introduce a new indicator, which is the named entity (like organization, person, location, position), because they are very important in the information extraction task. Totally, we obtain the following 5 indicators:

- Indicator 0: comma or period signs.
- Indicator 1: auxiliary verbs, such as “will” in the second sentence.

- Indicator 2: reference words, such as “he” and “one”.
- Indicator 3: prepositions, such as “of”, “in”, as well as “by”.
- Indicator 4: names and numbers, such as “CEO” and “George Grorrick”.

Easiest is Indicator 0. We get the chunks:

1 : George Grorrick

2 : 40 years old

3 : president of the famous hotdog manufacturer Hupplewhite

4 : was appointed CEO of Lafarge Corporation

5 : one of the leading construction material companies in North America

6 : He will be succeeded by Mr. John

7 : effective October 1**.**

Note: Oct. 1 was abbreviated. The computer can replace Oct. 1 by October 1.

Indicator 3 is easy too. The prepositions cut the sentence just before the preposition. Just try to speak the sentence with natural pauses to see why we did this. We now find:

31 : president

32 : of the famous hotdog manufacturer Hupplewhite

41 : was appointed CEO

42 : of Lafarge Corporation

51 : one

52 : of the leading construction material companies

53 : in North America

61 : He will be succeeded

62 : by Mr. John**,**

next to chunks 1, 2 and 7.

Indicator 1 is about auxiliary verb forms and these, like prepositions, cut before the form. So

611 : He

612 : will be succeeded**.**

Note that “was appointed” has an auxiliary verb, but the sentence was already cut before “was” by the comma indicator.

Indicator 2 concerns “one” and “He”, but these chunks already stand alone in chunk 51 and chunk 611. So far we got:

1 : George Grorrick

2 : 40 years old

31 : president

32 : of the famous hotdog manufacturer Hupplewhite

41 : was appointed CEO

42 : of Lafarge Corporation

51 : one

52 : of the leading construction material companies

53 : in North America

611 : He

612 : will be succeeded

62 : by Mr. John

7 : effective October 1.

We do not chunk up further in view of what we did so far. In particular “of”, “in” and “by” are linking **two** slots, indicating a relational template “in North America” is a chunk with one slot filled in, which may make it easier to find out the value of the other slot.

We want an **automatic** extraction procedure, to be followed by a computer. But a computer **cannot** make the jumps we make when we say “Now we make the semantic chunk graphs”. This is precisely the difficulty in artificial intelligence. We have to give very detailed instructions to go ahead in the information extraction process.

Now back to the names and numbers. We see CEO (Chief Executive Officer) as name, but it is the name of a position and so is “president”. “Officer” and “president” are both positions. The computer must know that and must know that CEO is short for Chief Executive Officer. When we prescribe/give slots like POSITION, then we must have a huge list of “values” for this slot and if “president” is not on that list, the computer cannot make semantic chunk graph 3. That is why it might be easier to generate names of slots ourselves. Suppose some lexicon gives: president: officer of a company, then we would introduce the slot OFFICER, and also for CEO we would do that. If the lexicon gives: president: position in a company, then we would generate the slot POSITION for “president” and OFFICER for “CEO”. Only when the computer knows that POSITION and OFFICER are similar, there is the possibility of reduction to just one slot, say POSITION. So this has its disadvantages too.

Assume that the computer knows all the prescribed slot names for the candidate words, so “president” is a word that can fill the slot POSITION. Because we prescribe the slots the lexicon of the computer may determine which words can fill one of the slots **EVENT**, PERSON, AGE, POSITION, COMPANY, LOCATION. The computer may know for example:

EVENT	: appointment, succession (recognition of verb forms, leading is not passing this test)
PERSON	: He, Mr.
AGE	: old (e.g. because the lexicon says something like “have age” for “old”)
POSITION	: president, CEO
COMPANY	: manufacturer, corporation, company
LOCATION	:

So for these words an interpretation as slot fillers is assumed to be directly possible. The computer may look up “appointment” and “succession” as values of **EVENT**, as these are to be described by nouns. There is one **EVENT** per sentence, so that has been settled (as only thing so far).

What other preliminary action can the computer take? The names like George, Grorrick, Hupplewhite, Lafarge, John, North America, October and the numbers 40 and 1 are supposed to be recognized as NAME and NUMBER respectively, but these are **not** among the given slots. What kind of names are they? The computer might find:

George	: name of PERSON
Grorrick	: name of PERSON or name of COMPANY or name of LOCATION

Hupplewhite	: name of PERSON or name of COMPANY or name of LOCATION
Lafarge	: name of PERSON or name of COMPANY
North America	: name of CONTINENT
John	: name of PERSON
October	: name of MONTH
40	: value of NUMBER
1	: value of NUMBER.

Before going over to the artificial intelligence part, let us remove adjectives: “famous”, “leading” and “effective”. Why? We are, given the slots, only interested in nouns, and more in particular in names and values. We can also replace by slot names where possible.

Having done all this preparation we now have the following:

1 : PERSON: George | PERSON, COMPANY or LOCATION: Grorrick

2 : NUMBER: 40 | years | AGE: old

31 : POSITION: president

32 : of the hotdog | COMPANY: manufacturer | PERSON,COMPANY or LOCATION: Hupplewhite

41 : EVENT: appointment | POSITION: CEO

42 : of PERSON or COMPANY: Lafarge | COMPANY: corporation

51 : one

52 : of the construction material | COMPANY: companies

53 : in | CONTINENT: North America

611 : PERSON: He

612 : EVENT: succession

62 : by | PERSON: Mr. | PERSON: John

7 : MONTH: October | NUMBER: 1.

From this we have to extract the desired information. That is, the computer has to and here is where the reasoning gets tougher for getting the semantic chunk graphs.

CHUNK 1 There are two names consecutive, one for a PERSON and one for PERSON, COMPANY or LOCATION. The computer should know that it has to conclude PERSON: George Grorrick.

CHUNK 2 NUMBER: 40 and SET: years stand consecutive, so “40 years”. This is followed by AGE: old which has a measure, so “40 years” must be the value of that measure. Conclusion AGE: 40 years.

CHUNK 31 POSITION: president. Here the main problem arises. OLD or NEW POSITION? The computer must choose OLD POSITION because of the place in the sentence. A position is attributed to a person and “president” follows “George Grorrick”, so OLD POSITION: president.

CHUNK 32 hotdog is FOOD, and Hupplewhite is the name of a PERSON, COMPANY or LOCATION. So we extract COMPANY: Hupplewhite, as the other noun occurring in this chunk is of type COMPANY: manufacturer. The link implied by “of” is to president, but that means that this is the OLD COMPANY: Hupplewhite.

CHUNK 41 EVENT: appointment POSITION: CEO. See later.

CHUNK 42 of COMPANY or PERSON: Lafarge COMPANY: corporation. There is no problem here, it must be COMPANY: Lafarge.

CHUNK 51 one. This reference word still has to be dealt with, if necessary. The place in the sentence suggests reference to COMPANY: Lafarge.

CHUNK 52 of the construction material COMPANY: companies. This chunk does not contain information relevant to the given slots.

CHUNK 53 in CONTINENT: North America. Expansion of CONTINENT gives LOCATION, so LOCATION: North America is found.

CHUNK 611 PERSON: He. The reference must be to a person mentioned in the first sentence. The only person is George Grorrick. Hupplewhite and Lafarge turned out to be companies.

CHUNK 612 EVENT: succession. See later.

CHUNK 62 by PERSON: Mr. PERSON: John. As “Mr.” is not a name the computer should combine to: by PERSON: John or Mr. John.

CHUNK 7 MONTH: October is a TIME-concept which is not one of the slots. So the computer should forget about this chunk.

As output we so far have for sentence 1:

EVENT	Appointment
PERSON	George Grorrick
AGE	40 years
OLD POSITION	president
OLD COMPANY	Hupplewhite
NEW POSITION	
NEW COMPANY	
LOCATION	

.

We used the chunks 1,2, 31, 32 and 41 partly. For sentence 2 we so far have:

EVENT	Succession
PERSON	
AGE	
OLD POSITION	
OLD COMPANY	
NEW POSITION	
NEW COMPANY	
LOCATION	

.

Age and location are not mentioned at all in sentence 2, but COMPANY and PERSON and POSITION do occur. This has to be decided by solving the OLD/NEW problem.

The chunks 41 and 612 are the vital ones. The computer has to know what appoint and succeed mean.

The lexicon might give “appoint” = “give POSITION to”. The only position mentioned in chunk 41 is CEO. This must therefore, implied by “give”, be the NEW POSITION. From chunk 42 then follows that Lafarge is the NEW COMPANY and the first template is filled after filling in the location: “North America”.

The lexicon might give “succeed” = “get POSITION of”. The preposition “by” leads to the proper choice. PERSON: John gets the NEW POSITION. This is implied by “gets”. The position is that of “He”, who is George Grorrick, so it is president and of NEW COMPANY: Hupplewhite. For OLD POSITION and OLD COMPANY nothing is found.

4.8 Chunk graphs for the example

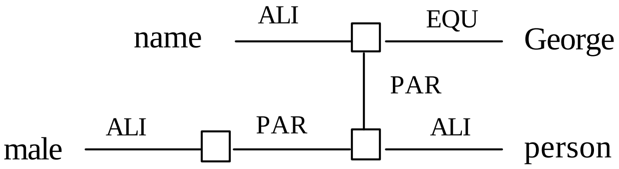
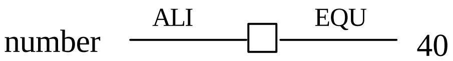
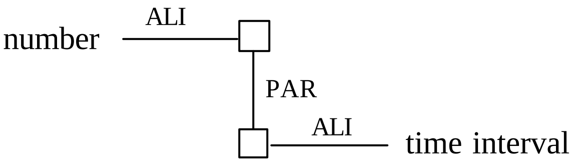
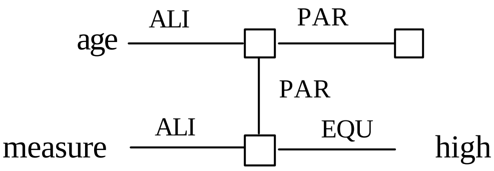
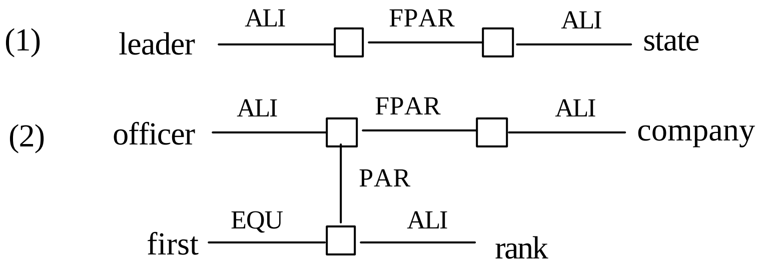
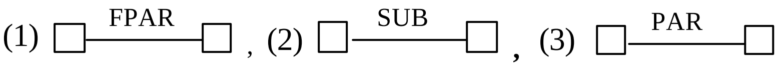
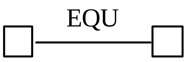
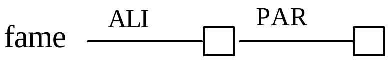
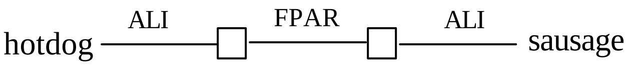
There is a difference between the status of the slot “**EVENT**” and the status of the other slots like “PERSON”, “AGE”, etc. The event is given by the whole text. To describe the event we choose to give nouns derived from the verbs used in the sentences. Thus we obtain “appointment” from “appointed” and “succession” from “succeeded”. For filling the other slots we should now discuss the role of semantic chunk graphs, as these form the essential parts of structural parsing. We will describe four phases illustrating the discussion given sofar.

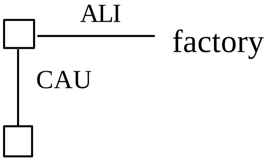
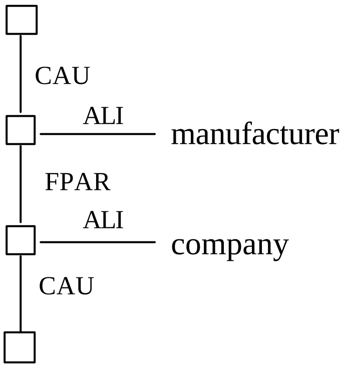
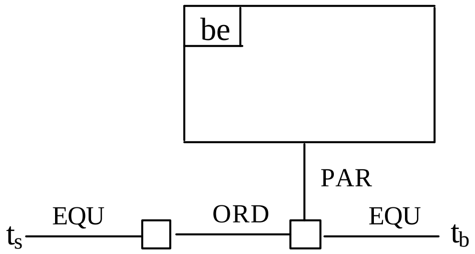
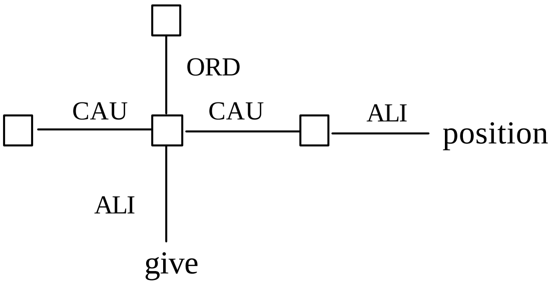
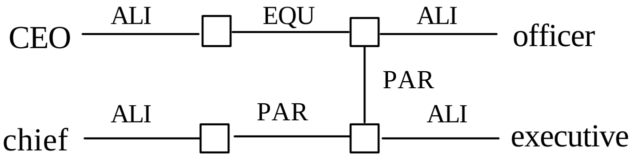
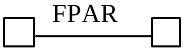
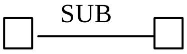
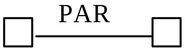
The first phase gives the word graphs of the words occurring in the two sentences. We discuss the construction of these word graphs from an imaginary lexicon. The information included in the lexicon might be as follows:

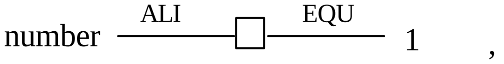
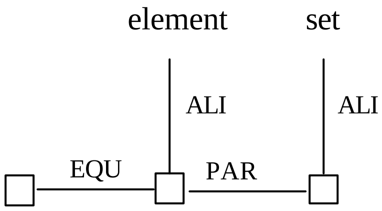
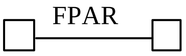
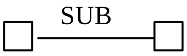
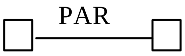
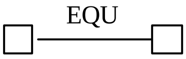
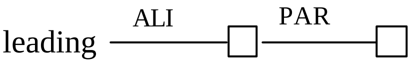
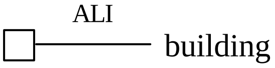
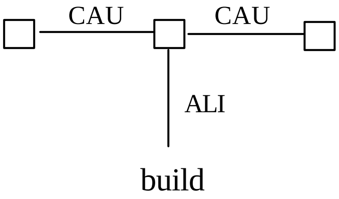
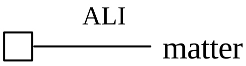
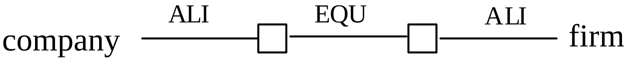
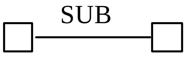
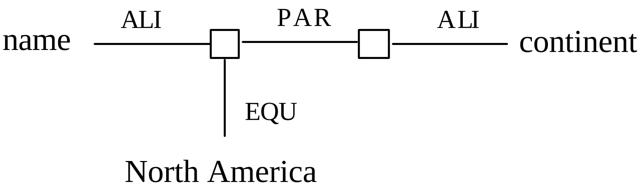
1.	George	: Name of a male person.
	Grorrick	: No information in the lexicon.
2.	40	: Number.
	Year	: Measure of time interval.
	Old	: Of high age.
3.	President	(1) Leader of a state, (2) First officer of a company.
	Of	: Preposition, used for describing a property, part or attribute.
	The	: Determiner.
	Famous	: Having fame.
	Hotdog	: Kind of sausage.
	Manufacturer	: (1) Factory, (2) Kind of company.
	Hupplewhite	: No information in the lexicon.

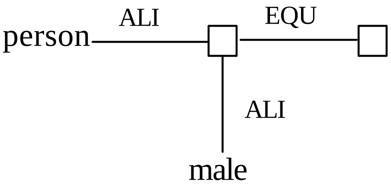
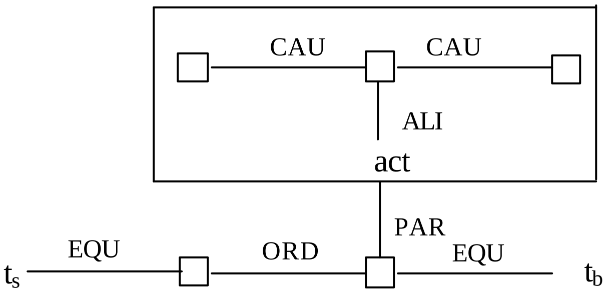
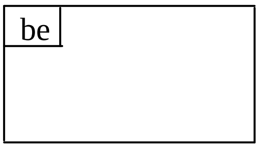
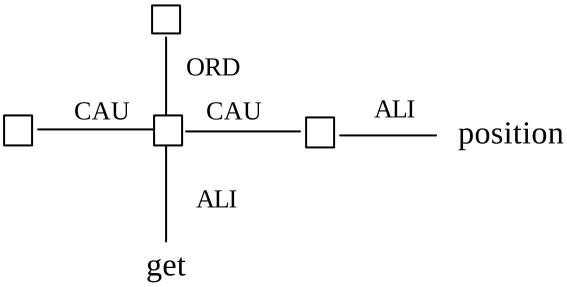
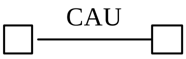
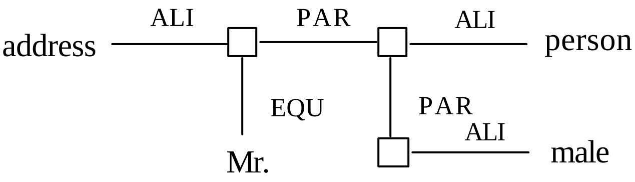
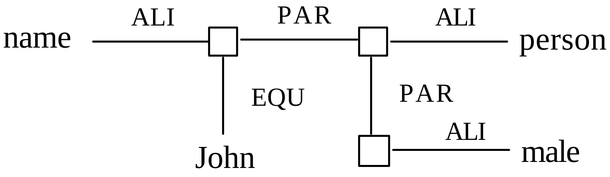
4.	Was	: Form of the verb “be”.
	Appoint	: Give a position to.
	CEO	: Shorthand for Chief Executive Officer.
	Of	: Preposition, used for describing a property, part or attribute.
	Lafarge	: No information in the lexicon.
	Corporation	: Kind of a company.
5.	One	: (1) Number, (2) Pronoun, referring to an element of a set.
	Of	: Preposition, used for describing a property, part or attribute.
	The	: Determiner.
	Leading	: Adjective, built from the verb “lead”.
	Construction	: (1) Building (2) The act of building.
	Material	: Matter.
	Company	: Synonym of “firm”.
	In	: Preposition, used for describing that something is part of something else.
	North America	: Name of a continent.
6.	He	: Pronoun, referring to a male person.
	Will	: Form of the auxiliary verb “will”, used to express acts in the future.
	Be	: Auxiliary verb, used to express a situation.
	Succeed	Get the position of.
	By	: Preposition, used for describing an actor or a cause of a verb.
	Mr.	: Address form for a male person.
	John	: Name of a male person.
7.	Effective	: (1) Causing effect, (2) Starting.
	October	: Name of a month.
	1	: Number.

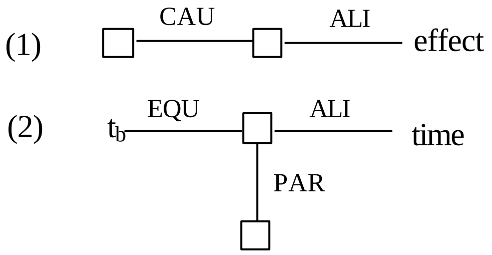
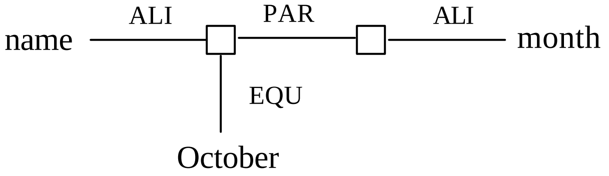
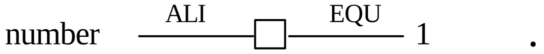
The word graphs for these words can now be constructed

1.	George:	 name ALI EQU George PAR male ALI PAR ALI person
	Grorrick:	
2.	40:	
	year:	
	old:	
3.	president:	 (1) leader ALI FPAR ALI state (2) officer ALI FPAR ALI company PAR first EQU ALI rank
	of:	
	the:	
	famous:	
	hotdog:	

	manufacturer:	(1) , (2) 
	Hupplewhite:	
4.	was:	 be PAR EQU ORD EQU t_s t_b
	appoint:	
	CEO:	 CEO ALI EQU ALI officer PAR chief ALI PAR ALI executive
	of:	(1) , (2) , (3) 
	Lafarge:	

5.	one:	(1) , (2) 
	of:	(1) , (2) , (3) 
	the:	
	leading:	
	construction:	(1) , (2) 
	material:	
	company:	
	in:	
	North America:	 name ALI PAR ALI continent EQU North America

6.	he:	
	will:	 CAU CAU ALI act PAR t_s EQU ORD EQU t_b
	be:	
	succeed:	
	by:	
	Mr. :	 address ALI PAR ALI person EQU PAR ALI Mr. male
	John:	 name ALI PAR ALI person EQU PAR ALI John male

7.	effective:	 (1) CAU ALI effect (2) t_b EQU ALI time PAR
	October:	
	1:	

These word graphs contain only little relevant information. There are two persons: “George” and “ John”. “Age” is mentioned in “old”, but not specified of whom. “Position” and “Company” occur here and there, also without specification.

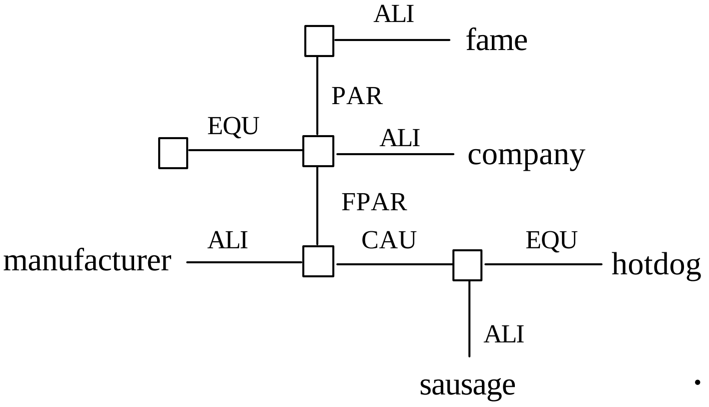
The second phase is to build chunk graphs from these word graphs. Note that we use *partial* structural parsing. The information that is to be extracted may be found from the chunks 1, 2, 31, 32, 41, 42, 51, 52, 53, 611, 612, 62, 7. Only if necessary, we combine these chunk graphs into graphs for larger chunks. If possible, we want to avoid complete structural parsing.

Chunk 1 : We only have at our disposal the word graph for “George”.

Chunk 2 : The three word graphs cannot yet be combined.

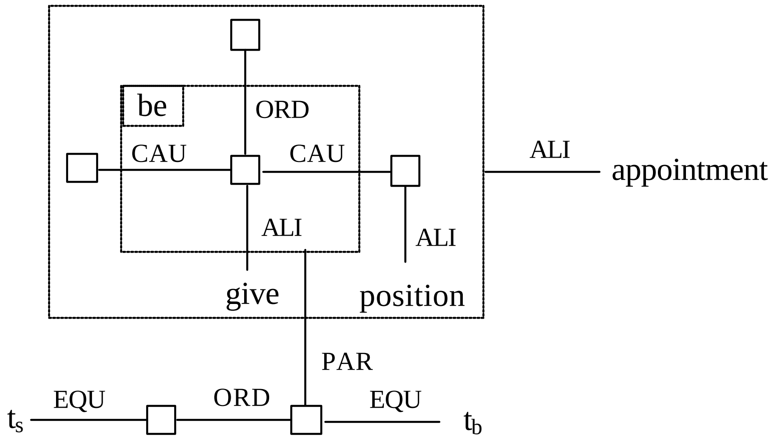
Chunk 31 : As this chunk has only one word, the chunk graph is just the word graph for “president”. We choose alternative (2).

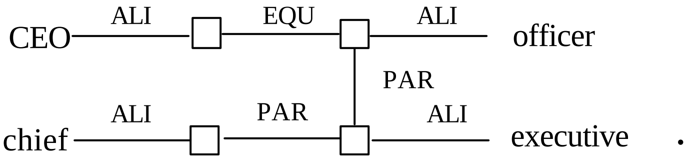
Chunk 32 : We choose alternative (2) for “manufacturer”. Using the same methods as in [Hoede & Zhang, 2001b], choosing alternative (1) for “of”, we obtained



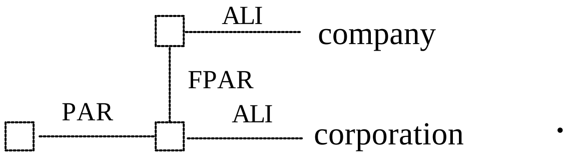
We cannot introduce "Hupplewhite" yet.

Chunk 41 :





Chunk 42 :

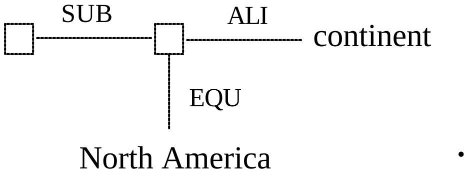


We cannot introduce "Lafarge" yet.

Chunk 51 : As this chunk has only one word the chunk graph is just the word graph for "one".

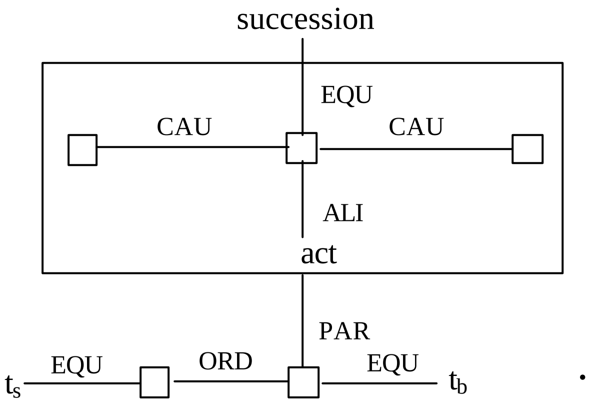
Chunk 52 : Without background knowledge, the word graphs cannot be combined.

Chunk 53 :



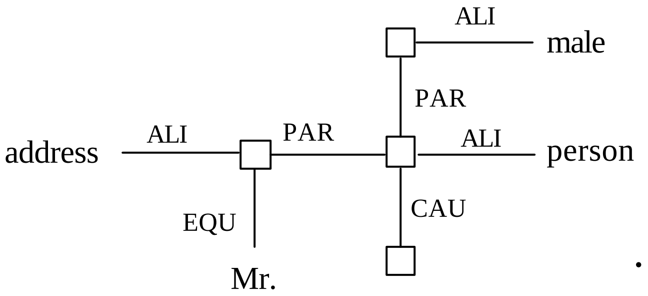
Chunk 611 : This chunk has only one word again.

Chunk 612 :

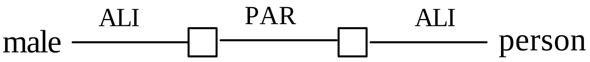


Note that the "act" is "be succeeded" and that this verb was already processed to fill the slot **EVENT**. Compare with the act "was appointed" in the first sentence.

Chunk 62 :



Note that "Mr." and "John" can be combined if we assume that the fact that both word graphs contain the subgraph

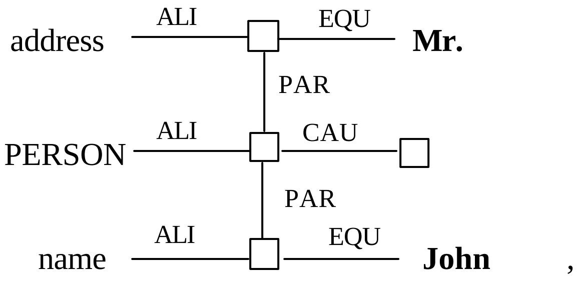


justifies this.

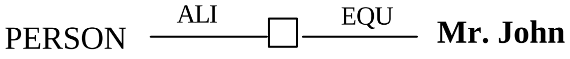
This is an example of similarity of two word graphs.

Chunk 7 : There is no possibility to combine the three word graphs.

Remarks: Due to the fact that various names did not have a word graph, the filling of slots is still not very well possible. Only Chunk 62 gives information when "Mr." and "John" are combined. Then the chunk graph for "by Mr. John" is



where we now wrote person in capitals as this is one of the slots. From the chunk graph we now read off “Mr. John” as filler of the slot PERSON, we may replace the graph by

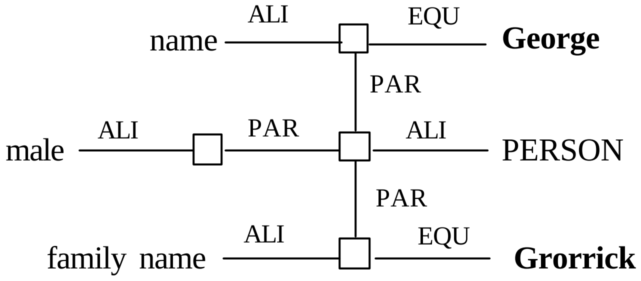


.

The third phase introduces reasoning by expansion of concepts. This holds both for the names of the slots and for the words occurring in the chunk graphs. As an example we consider the slot LOCATION and the word “continent” in Chunk 53. Any word graph for LOCATION may contain several instantiations or associations, without mentioning “continent”. Likewise the word graph for “continent” may not contain the concept “location”. However, this is rather unlikely. Describing a continent will involve mentioning its location.

To illustrate how important the expansion process is for obtaining our extraction goal, how much background knowledge is needed, we will now discuss the construction of chunk graphs in detail.

Chunk 1 : The word “Grorrick” was not encountered in the lexicon. Yet it has to be represented in relation with “George” as both words belong to the same chunk. What we need is *relevant* background information about “George”. It is a name in English, in fact it is a first name. Persons have both a first name and a family name. This is what makes it plausible that “Grorrick” is a family name. This information should be available to the computer. Note that it might be possible for the computer to expand the concept “name” to obtain this information. If not, the computer has no way to handle the word “Grorrick”. The chunk graph becomes:

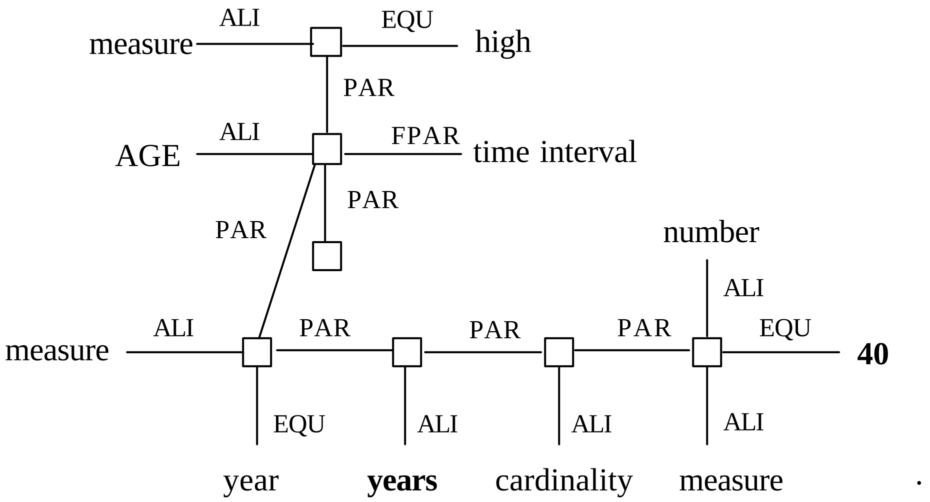


,

and we have found the filler for the slot PERSON in the first sentence.

Chunk 2 : The relevant background information in this case is that “old” says something about a time interval. “40” stands before years (plural) and therefore relevant background information is that “years” is a set. If expansion shows that “40” can be the value of the cardinality of a set we can combine “40” and “years”. Expansion of “age” in the word graph of “old” may yield that it is a time interval.

Now we can combine into:



This rather complicated chunk graph contains AGE. The filler of AGE may be chosen from this graph by noting that the words “40” and “years” occur in the text. Other words are due to the construction of the word graphs (like “high”) or due to the expansion process (like “cardinality”).

Chunk 3 : The subchunks 31 and 32 each pose a special problem.

In chunk 31 only “president” is mentioned. The word graph contains the concept “officer”.

The slots OLD POSITION and NEW POSITION contain POSITION and a list of possible positions might **not** include “president” but may include “officer”. On the other hand expansion of “president”, by expansion of “officer”, may lead to the conclusion that “president” is a position.

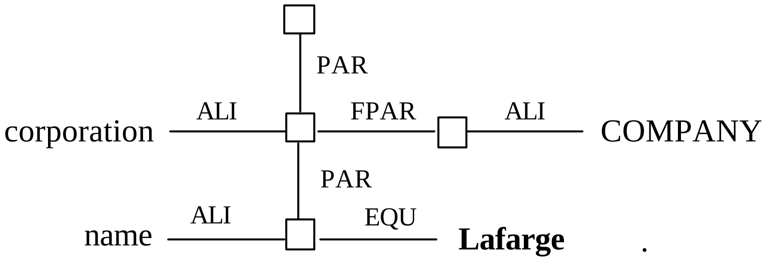
In both ways the link between POSITION and “president” can be established. What remains is the problem with OLD and NEW, as we already discussed just before we considered building chunk graphs. Solving that problem involves using the given text and not just expanding words of the chunk graph.

In chunk 32 the word “Hupplewhite” poses the problem. Being a word in the middle of a sentence beginning with the capital H suggests that “Hupplewhite” is a name. This also uses the given text. Therefore we should, in principle, not process this word in this third phase. However, we will discuss it here. The fact that the word follows the word “manufacturer” implies that it is the name of that “manufacturer”.

The chunk graph for 32, constructed sofar ties the name up with COMPANY, and therefore we have found another potential filler. However, also COMPANY only occurs in the slot names OLD COMPANY and NEW COMPANY, so that we have the same problem as for OLD POSITION and NEW POSITION again.

Chunk 41 : The two subchunks can be combined due to the fact that expansion of “officer” gives that it is a “position”. From the combined graph we read off that CEO is a filler of POSITION.

Chunk 42 : “Lafarge”, like “Hupplewhite”, must be a name and stands right before “corporation”, and as “corporation” is of type COMPANY we find another filler of OLD COMPANY or NEW COMPANY. The chunk graph looks like



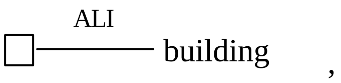
The two chunk graphs could be combined by remarking that, in chunk graph 42, the PAR-link, that represents “of”, has a token that should occur in chunk graph 41. The word order suggests that this is “CEO”. For the extraction of knowledge, in the form of slot fillers, this combining is not absolutely necessary. Note that the subchunks 41 and 42 already gave the answer.

Chunk 5 : Chunk 51 must be interpreted as a pronoun, because “one” is used and not “1”, so we have to choose word graph (2).

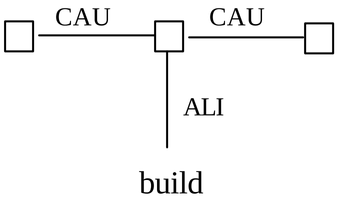
Chunk 52 poses the main problem, coming from the phrase “leading”, as an adjective may be combined with “construction” as a noun. However it is to be combined with “companies”. How can a computer interpret the three consecutive nouns “construction”, “material” and “companies”? The basic idea is to use expansion of the, small, word graphs given. Suppose we consider:

Construction:

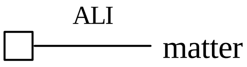
(1)



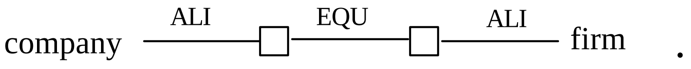
(2)



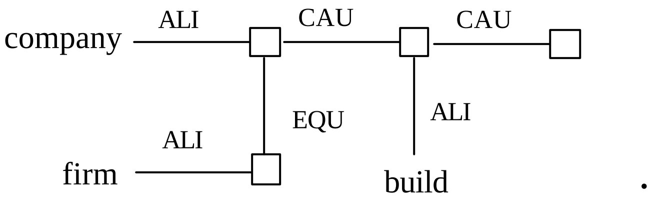
Material:



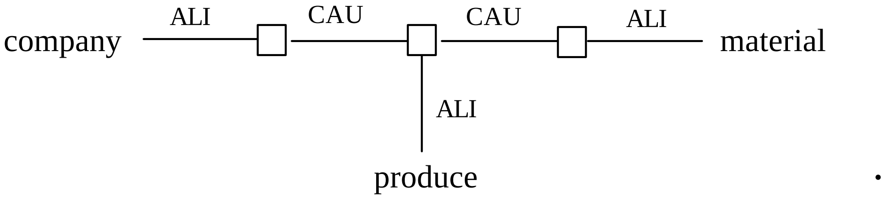
Company:



We have to find proper expansion. Let us start by saying that a “company” does something, i. e., there is a CAU-arc going out from its token. This suggests that for construction we use the second word graph and then we can already construct

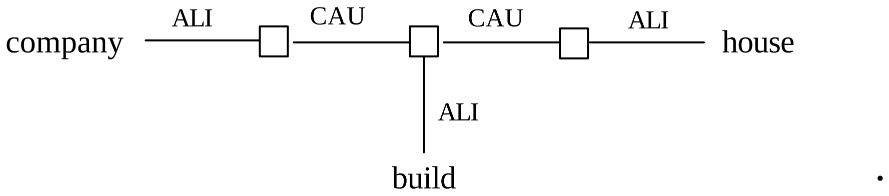


The word “material” or “matter”, because of its standing on the right of “construction”, must be expanded to link up with “building” as an instrument. However, it can also be linked with “companies” if we expand “companies” as entities producing something. This would lead to a graph like



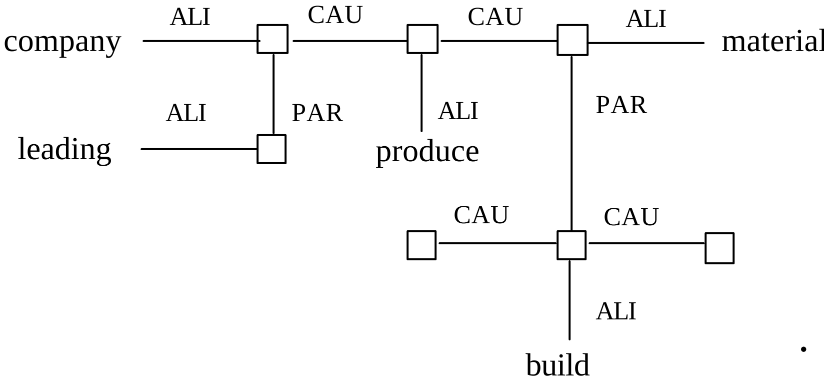
Note now that without the word “material” we would read “construction companies” and the first linking of graphs would be the only one. The sentence might have had the phrase “house construction companies”. That phrase indicates that the companies construct houses. The computer has to know how to deal with a sequence of nouns. We might instruct it in the sense that the last noun is the essential one. This would then mean that the adjective “leading” is to be attached to “companies”.

Then the relation with the forelast noun should be established. A “material company” asks for an interpretation of “material”. Is this a noun or an adjective? The lexicon only gave the noun interpretation. But then we can link by the expansion that companies produce, leading to the second graph. If “construction” is the forelast noun the first graph would result: the company constructs. The first noun in “house construction company” would be interpreted as the unspecified token, which would lead to the, correct, graph



The first noun in “construction material company” has to be linked to the graph constructed sofar for “material company”, and, again, linking with the forelast word, “material” is searched for. We could use both word graphs for construction semantically. “Building material” can be both material of which a building consists (*after* the building) and material used for building (*during* the building). So, basically, there is very subtle ambiguity here. In practice, we would prefer to use the second interpretation, so material used, as instrument, during the building process.

As a result, we have



We have given this discussion, because of the interesting problem of constructing a chunk graph here. For the goal of information extraction it does not give any answer in the form of a slot filler.

Chunk 53 yields a slot filler as the expansion of “continent” may lead to the information that it is a LOCATION. Let us recall that for the **EVENT** "appointment" slot names were specified, of which LOCATION was one. Finding a filler for this slot is enough to conclude that we have found the required filler as chunk 53 belongs to the first sentence. More detailed expansion can lead to the information that it is the location of "companies" and, via "one", the location of "Lafarge corporation". But such a detailed analysis is not necessary. This is an example of the usefulness of *partial* structural parsing.

Chunk 6 : We now have to investigate the second template and find fillers for the slots.

The sentence is rather short indeed. We already used the word "succeeded" to fill the slot **EVENT** with "succession". Next to that "Mr. John" was localized as a PERSON. So from the sentence part "He will be succeeded by Mr. John" we can only process chunk 611, which is the pronoun "He". But this pronoun refers to a person, George Grorrick, mentioned in the first sentence. The implications of this cannot be found by expansion within Chunk 6.

Chunk 7 : Chunk 7 only contains data referring to TIME, but this was not chosen to be a slot name. So we can refrain from processing this chunk.

The second sentence sofar has only led to fillers for the slots **EVENT** and POSITION.

We have not been able to fill all the slots of the two templates corresponding to the two sentences. We definitely need some extra reasoning.

In the fourth phase we do not expand word graphs with lexical information, but, as remarked before, now the context information is used to decide upon fillers. We will not do this in detail, but will only mention what can be decided in this phase for this example.

For the first template, "appointment", in principle enough information was found to fill the slots OLD POSITION, NEW POSITION, OLD COMPANY and NEW COMPANY. However, it was still to be decided which name should fill which slot. We have given a reasoning at the end of Section 4.7 to find

OLD POSITION	president
NEW POSITION	CEO
OLD COMPANY	Hupplewhite
NEW COMPANY	Lafarge

.

This completes the first template. For the second template OLD POSITION and OLD COMPANY cannot be determined. The pronoun “He” plays the vital role in determining the fillers for the slots NEW POSITION and NEW COMPANY of “Mr. John”. They are found by the fact that the word “succeed” is interpreted as “get the position of”, where the free token in the pronoun “He” is identified with the only person mentioned in the first sentence, who is George Grorrick. Therefore we get NEW POSITION: president and NEW COMPANY: Hupplewhite. The slot LOCATION is still to be filled. Due to the pronoun “He”, referring to George Grorrick, we can only conclude that the succession took place at the manufacturer Hupplewhite. However, this company might be a company in South America. For the “appointment” a location is mentioned, but the LOCATION slot of the “succession” has to remain open.

Concluding, we see that there are four phases, that each can provide fillers for the chosen slots.

- The first phase, just the construction of word graphs, hardly gave any filler.
- The second phase, the construction of chunk graphs, gave some possibility to attach names to slots.

However, the important phases are:

- The third phase, in which expansion of word graphs gave the opportunity to link potential fillers to slots.
- The fourth phase, in which context information was used, in principle formed by both sentences, turned out to be of vital importance to decide on the proper choice of fillers.

All four phases should have their place in any automatic information extraction procedure, on the basis of KGExtract.

4.9 Discussion

Let us consider the 10 functionalities mentioned by Hobbs.

1. A Text Zoner. Clearly our parsing by chunks turns a text into a set of segments in much more detail.
2. A Preprocessor. Also this is covered by parsing by chunks.

3. A Filter. We consider the whole text without filtering. Filters, based on the types of slots could easily be added.
4. A Preparser. The chunks are the small-scale structures that people are looking for.
5. A Parser. One of the new features of our approach is that the traditional parse trees get a much less important role to play.
6. A Fragment Combiner. The formulation of Hobbs stresses the traditional representation forms of parse-tree or logical-form fragment. Both are replaced by knowledge graphs.
7. A Semantic Interpreter. The traditional approach is to start with syntactic aspects. As we have discussed in [Zhang, 2002], the essence of the knowledge graph approach is formed by the semantic aspects.
8. A Lexical Disambiguator. In our analysis of the example disambiguation took place by taking into account the other parts of the sentence. Consider the discussion about Chunk 32, Hupplewhite could be the name of a PERSON, a COMPANY or a LOCATION. As seen before there is the word COMPANY, the interpretation as name of a COMPANY is most likely. Disambiguation is context dependent.
9. A Coreference Resolver. This too is a typical AI-problem, that was solved by taking into account the context. See the discussion about Chunk 611.
10. A Template Generator. We get the filled in templates as knowledge graph structures.

The hardest problems seem to be those encountered in 8. Disambiguation and 9. Coreference Resolving. Although our main goal in this paper is to show the usefulness of the idea of partial structural parsing in the field of Information Extraction, the problems we hit upon deserve some further discussion.

We already saw in [Hoede & Zhang, 2001b] that background knowledge is decisive for obtaining a sentence graph with structural parsing. Let us end with the thesis that intelligence, and therefore also artificial intelligence, heavily depends on the use of background knowledge.

A word graph is considered to be without limits essentially. A concept and its nearest neighbors form a subgraph of the mind graph that can be called foreground knowledge. The subgraph of the mind graph arising after deletion of the concept token can be called background knowledge of that concept. In Section 4.7 we pointed out that expansion of concepts plays an important role in thinking. Given a concept the number of associations with that concept will in first instance be limited. A person does not have his whole mind graph at his disposal immediately. However, by considering the concepts in the associations, i.e. in the word graph of the concept, and replacing these concepts by their meaning, i.e. their word graphs, the word graph of the original concept can be “expanded” and a larger word graph is obtained. In principle this can go on indefinitely until the whole mind graph is obtained, i.e. a knowledge graph corresponding to all knowledge available to that mind.

For a computer approach, that is simulating this process, we have at our disposal the word graph lexicon. The smaller this lexicon, the fewer the associations the computer has and the less expansion can take place. Like for human beings, the computer’s abilities to think, i.e. link somethings, are highly dependent on its information. The more information is contained in the lexicon of word graphs, i.e. the larger these are, the higher the probability that by expansion relevant linking of concepts takes place. There is, however, a second source of information, namely the context in which the concept is considered.

If, like in our example, two sentences are given, for extracting information from the second sentence the computer has the information contained in the first sentence at its disposal too. Next to its internal information, contained in the lexicon, there is the external information contained in the *context*. In a way the context also expands the knowledge of the computer. This becomes even clearer when we consider a dialogue. The description of a dialogue by means of knowledge graphs can be as follows. Speaker A says something and a sentence graph is made for this. The answer of speaker B is likewise transformed into a sentence graph, that is joined with the first graph. Every time new information is exchanged the graph representing what has been said sofar, in each of the minds of the speakers A and B, is expanded. This expansion is also due to context, now coming from the dialogue partner and not from the foregoing text.

So there are two forms of expansion available to the computer. One is due to combination of word graphs from its lexicon, the other is due to context processing. The development of an automated information extraction procedure, based on this idea of expansion, is challenging.

Reference

[Aberdeen *et al.*, 1995] J. Aberdeen *et al.*, MITRE: *Description of the Alembic system used for MUC-6*. In: Proc. Sixth Message Understanding Conf., Columbia, MD, Morgan Kaufmann, 1995.

[Appelt *et al.*, 1995] D. Appelt *et al., SRI International FASTUS system: MUC-6 test results and analysis.* In: Proc. Sixth Message Understanding Conf., Columbia, MD, Morgan Kaufmann, 1995.

[Church *et al.*, 1996] K. Church, S. Young and G. Bloothcroft, *Corpus-based Methods in Language and Speech.* Dordrecht, Kluwer Academic, 1996.

[Cowie *et al.*, 1993] J. Cowie *et al.*, *The Diderot information extraction system.* In: Proc. of the first Conf. of the Pacific Association for Computational Linguistics, Vancouver, 1993.

[Cowie & Lehnert, 1996] J. Cowie and W. Lehnert, *Information Extraction*. In: Special NLP Issue of the Comm (Ed. Y. Wilks), ACM, 1996.

[DeJong, 1979] G. F. deJong, Prediction and substantiation: A new approach to natural language processing. *Cognitive Science*, 3: 251-273, 1979.

[Fisher *et al.*, 1995] D. Fisher, S. Soderland, J. McCarthy, F. Feng and W. Lehnert, *Description of the UMass Systems as Used for MUC-6.* In: Proc. of the 6th Message Understanding Conf., Columbia, MD, 1996.

[Gaizauskas *et al.*, 1995] T. Gaizauskas *et al.*, *Description of the LaSIE system as used for MUC-6*. In: Proc. Sixth Message Understanding Conf., Columbia, MD, Morgan Kaufmann, 1995.

[Grishman & Sundheim, 1996] R. Grishman and B. Sundheim, *Message Understanding Conference – 6: A brief history*. In: Proc. 16th International Conf. On Computational Linguistics, Copenhagen, 1996.

[Hobbs, 1993] J. R. Hobbs, *The generic information extraction system*. In: Proc. of the Fifth Message Message Understanding Conf., Morgan Kaufmann, 87-91, 1993.

[Hoede & Zhang, 2001b] C. Hoede and L. Zhang, *Structural Parsing.* In: Conceptual Structures: Extracting and Representing Semantics, Aux. Proc. of the 9th International Conf. on Conceptual Structures (Ed. G.W.Mineau), CA, USA, 75-88, 2001.

[Humphreys, 2000] Humphreys, *Two Applications of information extraction to biological science journal articles: enzyme interactions and protein structures.* In: Proc. of the Pacific Symposium on Biocomputing (PSB-2000), Hawaii, 505-516, 2000.

[Krupka, 1995] G. Krupka, *SRA: description of the SRA system as used for MUC-6*. In: Proc. Sixth Message Understanding Conf., Columbia, MD, Morgan Kaufmann, 1995.

[Mellish *et al.*, 1992] C. Mellish *et al.*, *The TIC message analyser.* Technical Report CSPR 225, University of Sussex, Sussex, England, 1992.

[MUC-3, 1991] *Proc. of the Third Message Understanding Conf.*, Columbia, MD, Morgan Kaufmann, May 1991.

[MUC-4, 1992] *Proc. of the Fourth Message Understanding Conf.*, Columbia, MD, Morgan Kaufmann, 1992.

[MUC-5, 1993] *Proc. of the Fifth Message Understanding Conf.*, Morgan Kaufmann, August 1993.

[MUC-6, 1995] *Proc. of the Sixth Message Understanding Conf.*, Columbia, MD, Morgan Kaufmann, 1995.

[MUC-7, 1998] *Proc. of the Seventh Message Understanding Conf.*, Washington, D.C., Morgan Kaufmann, 1998.

[Sager *et al.*, 1987] N. Sager, C.Friedman and M.Lyman, *Medical language processing: computer management of narrative data.* Addison Wesley, 1987.

[Schank & Abelson, 1977] R. C. Schank and R. P. Abelson, *Scripts, Plans, Goals and Understanding*. Lawrence Erlbaum Associates, Hillsdales, NJ, 1977.

[Yangarber & Grishman, 1998] R.Yangarber and R. Grishman, *NYU: Description of the Proteus/PET System as used for MUC-7 ST.* In: Proc. of the Seventh Message Understanding Conf., Washington, D.C., 1998.

[Zhang, 2002] L. Zhang, *Knowledge Graph Thoery and Structural Parsing*, Ph.D. thesis, University of Twente, Enschede, The Netherlands, ISBN 9036518350, 2002.