A LINGUISTIC APPROACH TO SEMANTIC EXTRACTION FROM TEXT

ABEL BROWARNIK AND ODED MAIMON
DEPARTMENT OF INDUSTRIAL ENGINEERING, TEL AVIV UNIVERSITY.

Abstract. Ontology learning from text is the process of distilling knowledge - both implicit and explicit. Machines acquire knowledge either through human intervention or by means of an automatic, human-less learning approach, i.e. unsupervised ontology learning, using unsupervised, automatic understanding of text. Text understanding makes resort to Machine Learning or to a Linguistics-based approach. Both approaches require that a semantic representation of the text be obtained. This paper describes the context of Ontology learning, emphasizing the extraction of semantic content. We review the possible approaches and propose a heuristics based linguistic model for the automatic extraction of semantic content. The model examines the structure of the English sentence and corpus-based facts showing that sentence length is bound. This leads to the conclusion that it is possible to use *finite state* automata to heuristically detect clause boundaries within sentences. We show a clause-semantics retrieval example that could not be solved using other methods currently available. The semantics of the whole sentence can be obtained by combining the semantics of each individual constituent clause, based on the sentence structure found. A further paper will present the complete automaton for clause boundary detection, together with detailed results and a comparison to other available approaches.

Keywords: *Ontology learning, text understanding, Machine-learning, Linguistic-based approach.*

1. Introduction

An ontology is – in the context of this paper - a formal, explicit specification of a shared conceptualization. Formal refers to the fact that the ontology should be machine readable (therefore ruling out natural language). Explicit means that the type of concepts used, and the constraints on their use are explicitly defined. Shared reflects the notion that an ontology captures consensual knowledge, that is, it is not private to some individual, but accepted by a group.

Ontologies are building blocks for the Semantic Web. An ontology can be used for machine translation, word sense disambiguation, question answering systems, text understanding and additional modern information technology tasks. Most of the best available ontologies (such as Cyc or SUMO) were prepared with human intervention, turning it into a resource that is both costly and difficult to obtain and maintain.

The goal of ontology learning is the (at least semi) automatic extraction of background knowledge from a given corpus (or set) of documents, to form an ontology. Two main approaches can be considered towards the goal:

- Extracting terms from the text, generalizing terms into concepts, building taxonomies of concepts and finally extracting relationships between concepts.
- Splitting the corpus documents into sentences, then finding the clauses composing the sentences (a sentence consists of one or more clauses), assigning each clause its semantic value and composing the semantic value of a sentence as the combination of the semantic value of the composing clauses. The final step in this approach is to find generalizations of the semantic statements.

While the first approach is widely covered in the literature, the second represents a novel attempt to cope with the task of Ontology Learning. This second approach will be covered in

REVISTA ELECTRÓNICA DE LINGÜÍSTICA APLICADA (ISSN 1885-9089)
2011, Número 10, páginas 9-29
Recibido: 29/11/2010
Aceptación comunicada: 14/03/2011

depth in a separate paper. As a preliminary step, this paper will deal with the task of extracting the semantic representation of a sentence, a task that still requires the attention of the research community.

The paper is organized as follows. Section 2 defines ontologies. Sections 3 and 4 deal with the approaches to ontology learning. Section 5 tackles the issue of semantic representation. In Section 6 we propose a heuristic-linguistic model to extract the semantic contents of text. We show an example of a finite state automaton for the task and state the future work towards a full-fledged framework for the extraction of semantic contents of text. Further research is described in Section 7.

2. Ontology

It is now widely accepted that Natural Language processing (NLP) requires making reference to large amounts of real world knowledge. Hence, a knowledge representation scheme is needed. Without it no search or inference would be possible. Ontologies are the main artifact for knowledge representation purposes.

Gruber (1993) defines ontology as an explicit specification of a conceptualization. The term is borrowed from philosophy, where ontology is a systematic account of Existence. For knowledge-based systems, what exists is exactly that which can be represented. Studer *et al*. (1998) define it as a formal, explicit specification of a shared conceptualization. A conceptualization refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. Explicit means that the type of concepts used, and the constraints on their use are explicitly defined. Formal refers to the fact that the ontology should be machine readable, which excludes natural language. Shared reflects the notion that an ontology captures consensual knowledge, that is, it is not private to some individual, but accepted by a group.

Feldman and Sanger (2007: 43) define an ontology as "*a tuple O:=*($C, \leq c$) *consisting of a set …*[of] *concepts and a partial order…labeled a concept hierarchy or taxonomy*".

Turnitsa and Tolk (2006) underline in their paper (focused on ontologies for Modeling and Simulation) the complexity of the issue:

> "[…] we can fall into the trap of thinking that an ontology can be quite a simple thing. If one wishes to derive the ontology for a particular domain space, all they would have to do is to somehow specify their conceptualization of that domain space. It sounds simple, until the details of how to specifying the conceptualization is actually attempted" (Turnitsa and Tolk 2006: 2).

There are several ontology types. Buitelaar *et al*. (2005b) classify ontologies as follows:

- Ontologies referring to high level concepts such as time, space, and event are classified as top-level ontologies and are generally domain independent. It makes sense to speak about unified, top-level ontologies.
- Ontologies describing the vocabulary related to a generic domain by specializing the concepts introduced in the top-level ontology are domain ontologies.
- A description of the vocabulary related to a generic task or activity by specializing the top-level ontologies is a task ontology.
- Finally, there are application ontologies. These are the most specific ontologies. Concepts in application ontologies often correspond to roles played by domain entities while performing a certain activity.

Top level ontologies, also called upper or foundation ontologies, were most likely constructed, at least initially, with human intervention (see Shah *et al.* (2006) for Cyc,

Masolo *et al.* (2003) for DOLCE and Niles and Pease (2001) for SUMO). It is worth noting that there are only a few upper ontologies and their rate of change is allegedly low.

Human intervention becomes an issue for the other types of ontologies. Those ontologies are numerous, potentially larger, and probably evolve often. The benefits of automatically (or at least semi-automatically) learning ontologies are one of the motivations of ontology learning research. Using the massive amount of written information (unstructured text) available on the World Wide Web makes the ontology learning task possible.

3. Ontology Learning - the first approach

The first approach to Ontology Learning can be said to focus on a mixture of assisted - or augmented - pattern extraction strategy, together with a hierarchic approach with a further generalization to cope with non-hierarchical issues (mainly relations but possibly concepts as well).

Ontobasis (Reinberger and Daelemans 2004) is a project aimed at identifying and formalizing the methodological and linguistic principles for ontology engineering, including ontology mining and algorithms for their automated learning, and developing new theories and algorithms for alignment and merging of ontologies from different distributed sources. Ontobasis tries to learn ontologies from domain specific corpora, applying on it syntactic parsing, selecting specific syntactic structures on which clustering and pattern-matching is performed. The objective is to extract semantic relations between nominal expressions. Reinberger and Daelemans (2004) report about Ontobasis' choice of unsupervised extraction techniques, based on the fact that they do not require external domain knowledge (i.e. thesauri or tagged corpora).

Navigli and Velardi (2004) present a method and a tool, OntoLearn, aimed at the extraction of domain ontologies from web sites, and more generally from documents shared among the members of virtual organizations. OntoLearn first extracts a domain terminology from available documents. Then, complex domain terms are semantically interpreted and arranged in a hierarchical fashion. Finally, a general purpose ontology, i.e.WordNet, is trimmed and enriched with the detected domain concepts.

The DOGMA (Developing Ontology-Guided Mediation for Agents) (Reinberger and Spyns 2005) approach takes the same starting point as the Object Role Modelling (ORM) method (Halpin and Morgan 2001): basic facts holding for an application domain are described by binary elementary sentences. The reduction of verbal descriptions (be it from a text corpus or formulated by domain experts) to elementary sentences (or lexons in DOGMA parlance) is situated at the linguistic level (Spyns and De Bo 2004). A meta-lexon is created by the move from the linguistic to the conceptual level (i.e. after meaning disambiguation - see Spyns (2005) for more details).

Alani *et al*. (2003) explain the importance of relations in text understanding. It exemplifies it by saying that most information extraction tools can recognize "Rembrandt" is a person and "15 July 1606" is a date, but not the fact that Rembrandt was born on 15 July 1606. It is by recognizing such relations that we are able to populate an ontology. The paper details Artequakt, a system that created an ontology for the artists and painting domain using IE tools and methods to populate the ontology from web documents, based on the ontology metadata and on Wordnet definitions. After dropping duplicates the system was provided with narrative construction tools. Upon querying the KB – through an ontology server – and retrieving relevant facts, the system generates a specific biography on demand. The authors argue that the same methodology can be used for other domains.

Byrne (2006) aims to produce a system for running queries that do not assume the user has expert knowledge of the data structure or the specialist domain terminology. This goal is to be attained, says the author, by performing two tasks:

- Extraction of two-place relations from free text: The purpose of this step is to translate key facts from the textual material into a standardized format by using a combination of rule-based and machine learning approaches.
- Automatic assembly of all relevant data into an ontology: An ontology is defined here as a graph of two-place relations where the edges represent predicates and the nodes the entities they apply to. All of the relevant information — from database fields, domain thesauri and the extracted textual relations — will be combined into such a graph.

Following Buitelaar *et al.* (2005) "[…]the process of defining and instantiating a knowledge base is referred to as knowledge markup or ontology population, whereas (semi-) automatic support in ontology development is usually referred to as ontology learning". Cimiano (2006) describes the tasks involved in ontology learning as forming a layer cake, with the following example:

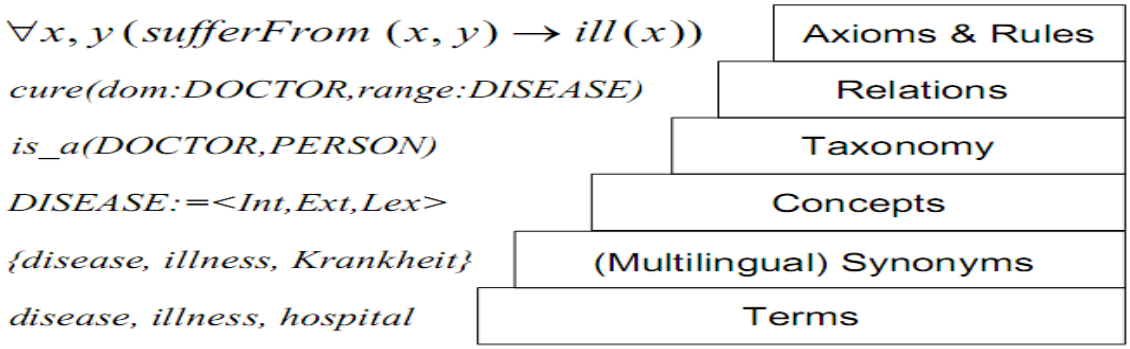


Figure 1: Cimiano's (2006) ontology "layer cake".

This approach assumes that terms (gathered through term extraction methods) are the basic building blocks for the task of ontology learning. There are many methods of term extraction (Bourigault (1999), Kozakov (2004), Wermter (2005)) and many tools are publicly available (Baroni (2004), Navigli and Velardi (2004), Sclano and Velardi (2007)).

The synonym layer is either based on ready-available sets such as Wordnet synsets (after sense disambiguation), on clustering techniques or other similar methods, or on web-based knowledge acquisition.

The concept layer is unclear, due to the fact that no consensual definition of what a concept is exists. The authors view is that it should include: an intensional definition of the concept, a set of concept instances, i.e. its extension, and a set of linguistic realizations, i.e. (multilingual) terms for this concept.

The concept hierarchy level (i.e. the taxonomic level) uses one of three paradigms to induce taxonomies from text:

- The application of lexico-syntactic patterns to detect hyponymy relations (Hearst 1992). This approach is known to have reasonable precision but very low recall.
- The exploitation of hierarchical clustering algorithms to automatically derive term hierarchies from text, based on the Harris' distributional hypothesis, that terms are similar in meaning to the extent in which they share syntactic contexts (Harris 1968)

- The third paradigm stems from the information retrieval community and relies on a document-based notion of term subsumption as proposed for example in Sanderson and Croft (1999). Salient words and phrases extracted from the documents are organized hierarchically using a type of co-occurrence known as subsumption.

The resulting structure is displayed as a series of hierarchical menus.

The relation level has been addressed primarily within the biomedical field. The goal of this work is to discover new relationships between known concepts (such as symptoms, drugs, and diseases) by analyzing large quantities of biomedical scientific articles.

Relation extraction through text mining for ontology development was introduced in work on association rules in Maedche and Staab (2000). Recent efforts in relation extraction from text have been carried on under the ACE (Automatic Content Extraction) program, where entities (i.e. individuals) are distinguished from their mentions, and normalization, the process of establishing links between mentions in a document and individual entities represented in a ontology, is part of the task for certain kind of mentions (i.e. temporal expressions).[1]

The rule level is only beginning to be explored (Lin 2001). The EU-funded project Pascal textual entailment challenge, a related task, (Ido *et al.* 2006) has driven attention to this problem.[2]

Sowa and Majumdar (2003) list a number of knowledge representation artifacts used in Artificial Intelligence. The list includes Lexical Semantics, (with WordNet as an example), Axiomatized Semantics (exemplified by several domain-dependent knowledge bases, as well as by Cyc), Statistical approaches (used for tasks such as deriving grammar rules from a corpus, or extracting relations from text), Knowledge Soup (Sowa 1990, 2000):

> "*What distinguishes knowledge soup from systems such as Cyc is the absence of a predefined, monolithic organization of the knowledge base. Instead, the knowledge can be developed and organized incrementally in an open-ended lattice of theories, each of which is consistent by itself, but possibly inconsistent with other theories in the same lattice*" (Sowa 1990: 459)

and Task-Oriented Semantic Interpretation, a simplification of the Knowledge Soup.

The proliferation of ontology learning methods and tools make evident the importance of the subject, and at the same time underlines the requirement for ontology evaluation methods and tools.

Gómez-Pérez and Manzano-Macho (2003) present a review of ontology learning methods and tools. Their review shows that the only semantic tool used is WordNet (if any). It should be noted that WordNet is not always considered a semantic tool.

As a closing remark, it is worth mentioning a list of shortcomings of the methods reviewed, as shown by Chen and Gaofeng (2005):

- the transition from the syntactic to the semantic layers does not fully take into account the semantic information of the sentences,
- the methods rely mainly on WordNet,
- the semantic information on the input corpus is not fully utilized,
- relation extraction is not based on semantics, and

there are limitations to the pattern methods used.

4. Ontology Learning - the second approach

[1] http://www.itl.nist.gov/iad/894.01/tests/ace/

[2] http://www.pascal-network.org/Challenges/RTE

The second approach focuses on the semantics of the text used to learn the ontology. Ideally, the process would include deep parsing, semantic labeling of the text and a process of knowledge accumulation. On top of it, there is a reasoning layer (with activities such as entailment, generalization, and more). This is one of the models of how humans learn what they learn (also from text), including all the background knowledge they possess. From a practical point of view, what seems ideal is – to date – rather unfeasible. To overcome these limitations, researchers apply practical approaches based on heuristics and partial methods.

The literature shows several attempts in this direction. The model proposed in Chen and Gaofeng (2005) makes extensive use of FrameNet, a frame semantics resource described in Fillmore (1976) and Fillmore *et al.* 2003). A semantic frame can be thought of as a concept with a script. It is used to describe an object, state or event. Chen and Gaofeng avoids the need to deep parse the sentences forming the text by using Framenet

> "*Given a tagged sentence as input, the syntactic parser produces a phrase-structure tree as output. The semantic role-labeling task can identify the semantic relationships, or semantic roles, filled by constituents of a sentence within a semantic frame based on the syntactic parsing tree. The whole process will generate the syntactic-semantic mapping structure*" (Chen and Gaofeng 2005: 377).

Chaumartin (2005) presents another attempt to tackle the semantic representation issues. Instead of using Framenet, the lexical-semantic transition uses VerbNet (Schuler 2006). Antelope is an implementation of Chaumartin's work.[3]

5. Semantic representation

Both methods (Chen and Gaofeng (2005) and Chaumartin (2005)) deal with text at the sentence level, without taking into account sub-sentence components. Chen and Gaofeng (2005) does not provide a tool to showcase the capabilities of his approach, except for an example on the paper:

(1) "They had to journey from Heathrow to Edinburgh by overnight coach. "

The example is assigned Framenet's frame *Travel* with all its elements (traveler, source and goal).

Chaumartin (2005) released a full-fledged toolbox to test the capabilities of his approach. The system comes with an example:

(2) "the general to whom President Lincoln gave all powers in Washington captured Lee's troops during the Battle of Gettysburg"

The result (i.e. the semantic representation) given by *Antelope* is:

[3] See http://www.proxem.com/Download/Research/ANTELOPE-revue_TAL.pdf

gave(**Subject**: President Lincoln, **DirectObject**: powers, **PrepObject**: *to* general, **SpaceComplement**: *in* Washington)

Predicate 'gave' may belong to frame_give#1 (score=3)
the **general**$_{(Recipient)}$ to whom **President Lincoln**$_{(Agent)}$ **gave**$_{(Verb)}$ all **powers**$_{(Theme)}$ in Washington captured Lee 's troops during the Battle of Gettysburg
Constraint on word 5 (VERB) - 8 sense(s): give#1 give#3 give#8 give#14 give#17 give#19 give#24 give#29
Constraint on word 1 (RECIPIENT) - 2 sense(s): general#1 general#2
gave(AGENT => President Lincoln [animate], THEME => powers [animate], RECIPIENT => general [animate], ASSET => ?, SOURCE => ?)
has_possession(*start(E)*, AGENT=President Lincoln, THEME=powers)
has_possession(*end(E)*, RECIPIENT=general, THEME=powers)
transfer(*during(E)*, THEME=powers)
cause(AGENT=President Lincoln, *E*)

---

captured(**Subject**: general, **DirectObject**: troops, **TimeComplement**: *during* Battle of Gettysburg)

Predicate 'captured' may belong to frame_steal#1 (score=2)
the **general**$_{(Agent)}$ to whom President Lincoln gave all powers in Washington **captured**$_{(Verb)}$ Lee 's **troops**$_{(Theme)}$ during the Battle of Gettysburg
Constraint on word 10 (VERB) - 1 sense(s): capture#5
Constraint on word 1 (AGENT) - 2 sense(s): general#1 general#2
captured(AGENT => general [animate], THEME => troops, SOURCE => ?, BENEFICIARY => ?)
manner(*during(E)*, Constant=illegal, AGENT=general)
has_possession(*start(E)*, ?SOURCE=?, THEME=troops)
has_possession(*end(E)*, AGENT=general, THEME=troops)
not(has_possession(*end(E)*, ?SOURCE=?, THEME=troops))
cause(AGENT=general, *E*)

Table 1: A semantic representation from Antelope.

The result is a clear semantic representation of the sentence in terms of VerbNet classes and all the resulting constraints. The representation includes all the semantic details necessary to assess the situation and allow for higher order activities such as question answering, reasoning and maybe automatic translation. Yet, for other sentences results are not satisfactory, as in:

(3) "Most of these therapeutic agents require intracellular uptake for their therapeutic effect because their site of action is within the cell"

or

(4) "Here is a word w which is the head word of a constituent in the sentence and is not recognized by the traditional method when finding its associated concept"

The examples above yield no result (i.e. no VerbNet class is recognized and therefore no semantic representation is extracted). One of the reasons that may lead the systems above to fail to discover the semantic contents of complex or compound sentences is probably the fact that such a sentence structure requires more than one frame or verb class to be found. The internals of Antelope may require, apparently, too many computations for such sentences. As a result, an acceptable coverage of multi-frame or multi-verb sentences may be beyond the reach of Antelope. In order to improve the ability to extract the semantic content from text (and obtain a semantic representation), it seems necessary to look deeper into the structure of the language.

In the rest of this paper we will argue that aligning clauses to VerbNet verb classes results in a wider coverage that allows for a more effective extraction of semantic representations. To this end, we will show an algorithm to split sentences into clauses. The

algorithm uses finite state automata (FSA) to split the sentences. While it is common to expect that natural language generally requires more complex computational devices than FSA, we will show that under certain, actually non-restrictive, conditions, English sentences are tractable with FSA.

6. The proposed model -A deeper grammatical analysis

In order to solve the shortcomings of the two above models - concerning the extraction of semantic representations from sentences - it is necessary to look deeper into the grammatical structure of sentences. Both Chen and Gaofeng (2005) and Chaumartin (2005) take sentences as the minimal grammatical construct, and check against it the lexical-semantic resource used by each method - Framenet for Chen and Gaofeng (2005), VerbNet for Chaumartin (2005). The difficulty to decompose a sentence into smaller meaning units (i.e. clauses) may be at the origin of such a decision. We will look deeper into the definitions of English grammar to find a more convenient way to extract a semantic representation.

The Longman Grammar of Spoken and Written English, or LGSWE (Biber *et al.* 1999) defines a spectrum of several grammatical units (or classes) forming a hierarchy:

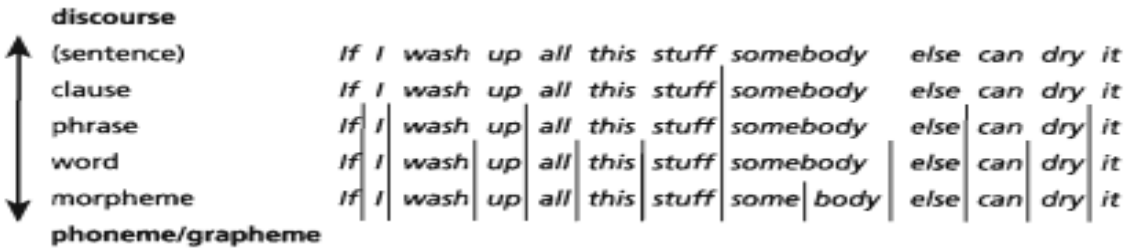


Figure 2: Hierarchy of grammatical units (from LGSWE, p. 50).

In general, a class is made out of one or more constituents of a lower level class. Discourse is made of one or more sentences, made, in turn, of one or more clauses. Clauses are made of one or more phrases. Phrases are constituted by one or more words that are themselves constituted by morphemes. It is possible to describe discourse in terms of morphemes or words, but this would be a very complex way to describe a language, a way that would make learning the language an extremely hard task. Using hierarchical structures, the task becomes more tractable.

We will focus on four types of grammatical units, namely words, phrases, clauses and sentences.

Words

- Lexical words – the main carrier of meaning in a text. Lexical words are numerous and form an open class. They can be nouns, verbs, adjectives and adverbs.
- Function words – indicate relationships between lexical words or larger units, or indicate how to interpret a lexical word or larger unit. Function words form a closed class. A function word may belong to one or more of the following subclasses:
    - Determiners – used to narrow down the reference of a noun
    - Pronouns – used instead of full noun phrases
    - Primary auxiliaries (be, have, do) – used to build complex verb phrases
    - Modal auxiliaries (can, could, may, might, must, shall, should, will, would) – used to build complex verb phrases. There are also semi-modals such as dare, need and others.

- Prepositions – used to introduce prepositional phrases. Examples of prepositions: about, after, around, as, at, by, down, for, from, in, into, like, of, off, on, round, since, than, to, towards, with, without, such as, as far as, and more.
- Adverbial particles – used mainly to give a meaning of motion or result. Examples: about, across, along, around, away, back, by, down, forth, home, in, off, out, over, past, through, under, up, etc.
- Coordinators – used to build coordinate structures – both phrases and clauses. The main coordinators are *and, but* and *or*. Other coordinators for special cases are *nor, either* and *neither*.
- Subordinators – used to introduce dependent clauses. There are three main subclasses of subordinators:
- Used to introduce adverbial clauses: after, as, because, if, since, although, whether, while and more.
- Used to introduce degree clauses: *as, than, that*.
- Used to introduce complement clauses: *if, that, whether*.
- Wh-words – used to introduce clauses. Except for *how* and *that* wh-words begin with *wh*. Wh-words are used as interrogative clause markers (as in "***What** do they* want") or as relativizers (as in "*the car **which** she had abandoned"*).
- Existential *there* – often described as an anticipatory subject (as in "*there is no other option*").
- The negator *not* – used mainly to negate clauses
- The infinitive marker *to* – used mainly as a complementizer preceding infinitive forms of verbs.
- Numerals – generally either ordinals (the answer to *which*) or cardinals (the answer to *how many*).
- Inserts – a type of word inserted freely on text, carrying often emotional and interactional meaning, especially frequent in spoken text.

The following table shows the differences between lexical and function words.

features	lexical words	function words
frequency	low	high
head of phrase	yes	no
length	long	short
lexical meaning	yes	no
morphology	variable	invariable
openness	open	closed
number	large	small
stress	strong	week

Figure 3: Typical differences between lexical and function words (from Biber et al 1999: 55).

An additional problem faced by the task of extracting semantic representations from text is the fact that words can belong to more than one class. A single word such as *like* can be a lexical word (noun, verb, adjective or adverb) or a function word (a preposition or a subordinator).

Phrases

Phrases are constituted by one or more words. A phrase may embed other phrases at different levels. Such an embedding may result in more than one meaning for the whole construct. Following the example on Biber *et al.* (1999: 94) the phrase "Mr Adamec threatened to quit last night" can be interpreted in two ways:

a. [Mr Adamec] [threatened] [to quit] [last night]
b. [Mr Adamec] [threatened] [to quit [last night]]

Phrases are indicated by brackets. The first interpretation means that Mr Adamec expressed last night his threat to quit sometime in the future, while the second interpretation means that Mr Adamec threatened to quit last night. The different levels of embedding are exemplified by the phrase [last night], which is embedded more deeply in b than in a.

There are several types of phrases, the major types being noun phrases, verb phrases, adjective phrases, adverb phrases and prepositional phrases.

- Noun phrases – a phrase that consists of a noun alone or accompanied by a determiner, and possibly followed by complements to complete the meaning of the phrase. A complex noun phrase can even be discontinuous, as in the following sentence (the noun phrase fractions appear in bold):

(5) "In this chapter a description will be given of the food assistance programs that address the needs of the family"

- Verb phrases – a phrase with a lexical verb or a primary verb as head or main verb, either alone or accompanied by auxiliaries. Verb phrases in our context do not include accompanying elements such as objects and predicatives. Verb phrases are often discontinuous, as in the following :

(6) "The current year *has* definitely *started* well"

- Adjective phrases – a phrase containing an adjective as head, probably accompanied by modifiers. Adjective phrases can be discontinuous as in the following:

(7) "You couldn't have a *better* name *than that*"

- Adverb phrases – a phrase headed by an adverb, possibly with modifiers.
- Prepositional phrases – it consists of a preposition and a complement. It may be viewed as a noun phrase extended by a link showing relationship to surrounding structures.

Clauses

LGSWE (Biber *et al.* 1999) define a clause as follows:

> "A clause is a unit structured around a verb phrase. The lexical verb in the verb phrase characteristically denotes an action (drive, run, shout, etc.) or a state (know, seem, resemble, etc.). […] The verb phrase is accompanied by one or more elements which denote the participants involved in the action, state, etc. (agent, affected, recipient, etc.), the attendant circumstances (time, place, manner, etc.), the attitude of the speaker1 writer to the message, the relationship of the clause to the surrounding structures, etc. Together with the verb phrase, these are the clause elements. The clause elements are realized by phrases or by embedded clauses".

A clause may be divided into two main parts: a subject and a predicate. The subject is generally a nominal part, while the predicate is mainly a verbal nucleus.

One of the main characteristics of LGSWE (Biber *et al.* 1999) is the study of texts according to their source. The book refers to four main *registers*, CONV (for conversation), FICT (for fiction), NEWS and ACAD (for academic text). The behavior differs between the different registers. For instance, the following table describes the distribution of sentences following the number of clauses included, from two samples on text on LGSWE (Biber *et al* 1999: 102-121):

form	CONV	NEWS
non-clausal word forms	21	0
single-clause units	15	1
two-clause units	3	2
three-clause units	2	2
four-clause units	0	2

Table 2: Distribution of clausal units and non-clausal material in two text samples (from Biber et al. 1999: 121)

This and other types of differences will be used in following sections of this paper. The CONV register will be skipped. While this paper will refer mainly to ACAD, the methods apply also to NEWS and FICT.

Preisler (1992) focuses on three of the grammatical structures we defined so far: clauses, phrases and words. A clause in this context contains the constituents Subject, Verbal, Complement and Adverbial, all or some of them. Each of the constituents is in fact a phrase (there are several phrase types). A phrase is constituted by a head (H) and a modifiers, either a premodifier (PRM) or a postmodifier (POM), or any combination of such constituents. In turn, words are also made of a root ® and affixes, either prefixes (Pf), suffixes (Sf) or a combination. Preisler summarizes the constructs in a rank scale as follows:

Rank	Structure	Constituents	Realization
1	Clause	S V C A	Phrases
2	Phrase	PRM H POM	Words
3	Word	Pf R Sf	Morphemes

Table 3: Rank scale of grammatical analysis (from Preisler 1992: 21)

Yet, there is a variation, for example, where a phrase constituent can be another phrase (as in "a ridiculously low price"), instead of a word, as we would expect, or even a clause that may be a constituent of a phrase (as in "The Lady sitting on the couch"). Such a behavior is called rank shifting (see Preisler (1992:22). Rank shifting adds complexity to the task of extracting semantic representation, because the automatic detection of clause boundaries that is essential to the decomposition of sentences becomes a very complex task. In this context, rankshifted clauses are called subclauses, while non-rankshifted clauses are called main clauses.

Clauses appearing together in a larger unit (generally sentences, but possibly phrases in the event of a rank-shifted clause) are linked by structural links, the principal types being coordinators, subordinators and wh-words. These were already mentioned in the function word section above. C*oordinators* create coordinated clauses such as:

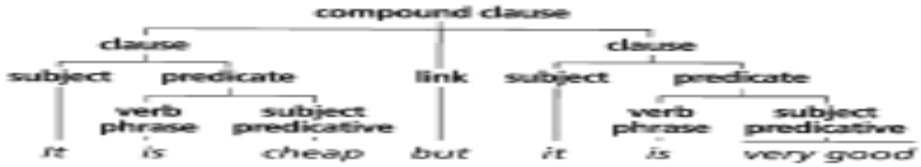


Figure 4: Coordinated clause (from Biber et al. 1999: 135).

On the other hand, subordinators and wh-words create embedded clauses such as the following three examples:

Figure 5: Main clause with embedded adverbial clause (from Biber et al. 1999: 135).

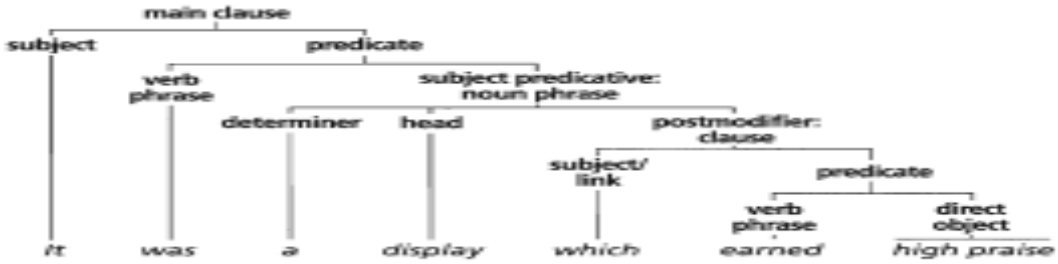


Figure 6: Main clause with embedded relative clause (from Biber et al. 1999: 136).

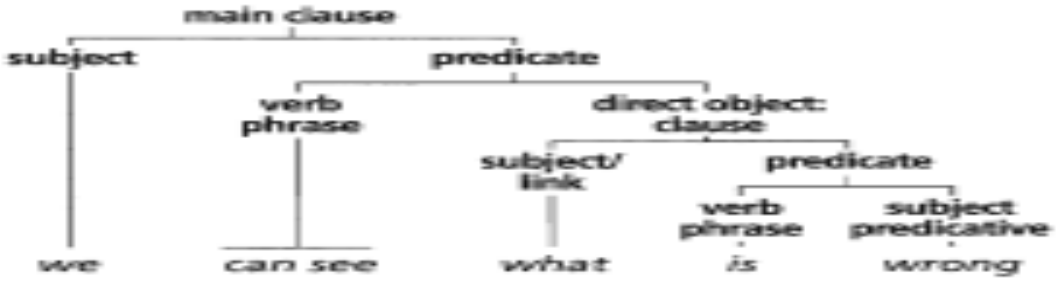


Figure 7: Main clause with embedded nominal wh-clause (from Biber et al. 1999: 136).

Sentences

Preisler (1992) defines a sentence as "one or more main clauses, corresponding to units which in written language are bounded by the punctuation mark".

SIL defines a sentence as "a grammatical unit that is composed of one or more clauses".[4]

Jones *et al.* (1922) state that "a sentence is a word or a group of words expressing a complete thought". This definition is, at least, ambiguous. How does one characterize a complete thought? Moreover, a complete thought may require much more than a single sentence.

Sentences can be classified as follows:[5]

[4] (http://www.sil.org/linguistics/GlossaryOfLinguisticTerms/WhatIsASentence.htm).
[5] http://en.wikipedia.org/wiki/Sentence_(linguistics)

By structure

- A simple sentence consists of a single independent clause with no dependent clauses.
- A compound sentence consists of multiple independent clauses with no dependent clauses. These clauses are joined together using conjunctions, punctuation, or both.
- A complex sentence consists of at least one independent clause and one dependent clause.
- A complex-compound sentence (or compound-complex sentence) consists of multiple independent clauses, at least one of which has at least one dependent clause.

By purpose

- A declarative sentence or declaration, the most common type, commonly makes a statement: "I am going home".
- An interrogative sentence or question is commonly used to request information — When are you going to work?" — but sometimes not
- An exclamative sentence or exclamation is generally a more emphatic form of statement expressing emotion: "What a wonderful day this is!"
- An imperative sentence or command tells someone to do something: "Go to work at 7:30 in the morning".

6.1. Grammatically driven extraction of semantic representation

The grammatical analysis presented in the previous section shows that it is possible to see a sentence as a composition of one or more clauses either coordinated or subordinated. The coordinating or subordinating links, implemented generally by means of function words, determine the meaning of the combination of clauses (when the sentence consists of more than one clause).

The two semantic representation extraction approaches reviewed above (Chen and Gaofeng 2005, Chaumartin 2005) look for frames or verbal classes in FrameNet and VerbNet respectively. Their search compares the whole sentence to the respective lexical-semantic resource, with mixed results, as reported.

In fact, the alignment of a multi-clause sentence to a verbal class instance (in VerbNet) or a semantic frame (in FrameNet) misses the fact that both verbal class instances and semantic frames are – in general – rather simple constructs. Both implement items that aim at capturing an individual fact or situation. As an example, see the verbal class own-100, with a single possible frame, NP V NP (i.e. "I own twelve oxen"). Hence, it seems appropriate to align clauses (and not whole sentences) to either a Framenet frame or a VerbNet verbal class (as noted before, both Framenet and VerbNet share some sort of equivalence).

As an example, the following sentence (already used in the context of a test of Antelope above) consists of a main clause followed by an adverbial (subordinated) clause introduced by the function word because (clauses are enclosed in brackets).

> [Most of these therapeutic agents require intracellular uptake for their therapeutic effect] because [their site of action is within the cell]

The main clause fits into the "require-103" VerbNet class, specifically into the frame "NP V for NP S_INF" of the above VerbNet class. The subordinate clause includes the verb "be". "To be" is not included on its own in a VerbNet class, but it can be dealt with without major

complications. The function word *because* makes clear that the content of the main clause has a causal relationship to the subordinate clause (the main clause occurs *because* of the subordinate clause).

Automatic extraction of semantic representations

The automatic extraction of semantic representation of sentences requires that an algorithm be designed for the task. Such an algorithm should cope with the problems associated with the task:

- Splitting English sentences into clauses (i.e. finding clause boundaries) is not trivial (although English speakers do it rather easily)
- More than one verbal class or frame may be suitable for a given clause. This is one of the types of ambiguity in language.

The rest of the section deals with finding clause boundaries. For the second problem we currently adopt the heuristics proposed by Chaumartin (2005). Chaumartin (2005) proposes that when more than one option exists for the assignment of verbal class to a sentence (or clause), the assignment with the higher number of bound thematic roles, as described in Schuler (2006), should be picked as the best fit. As an example, Chaumartin (2005) gives the sentence "The cultivator eliminated the beetles with pesticide". This sentence fits the following interpretations:

"murder-42.1, frame 1": the cultivator$_{(Agent)}$ eliminated$_{(Verb)}$ the beetles$_{(Patient)}$ with pesticide
"murder-42.1, frame 2": the cultivator$_{(Agent)}$ eliminated$_{(Verb)}$ the beetles$_{(Patient)}$ with pesticide$_{(Instrument)}$
"remove-10.1, frame 1": the cultivator$_{(Agent)}$ eliminated$_{(Verb)}$ the beetles$_{(Theme)}$ with pesticide

Chaumartin (2005) proposes to pick a verbal class following the rule that the interpretation with the higher number of thematic roles assigned is more likely to be the best suited interpretation. In the example above, the rule would predict that the preferred verbal class is "murder-42.1, frame 2", as would a native English speaker. A paper including a benchmark for this heuristic based disambiguation method will be made available in the near future.

6.2. Splitting sentences into clauses

The task of splitting sentences into clauses takes as input a single sentence. The output consists of a set of clauses and the links between the clauses found (when the sentence consists of more than one clause). Finding the boundaries and identifying the links between clauses requires computational devices such as *finite state automata* (FSA) or pushdown automata (PDA).

6.2.1. Finite state automata

A finite state automaton is a simple mathematical model of computation used for text processing, compilers and other tasks. FSA represent computers with a very limited amount of memory. A finite state automaton M is a 5-tuple $(Q, q_0, A, \Sigma, \delta)$, where

Q is a finite set of states

q_0 is the start state

$A \subseteq Q$ is the set of accepting (or final) states

Σ is a finite input alphabet

$\delta : Q \times \Sigma \rightarrow Q$ is the transition function of M

A finite automaton starts in state q_0 and processes the input symbols sequentially. It moves from state to state (i.e. makes a transition) according to the input symbols. If the current state q is an accepting state (i.e. $\in A$) then the input sequence read so far is accepted. A sequence that is not accepted is rejected. Figure 7 shows a two-state automaton.

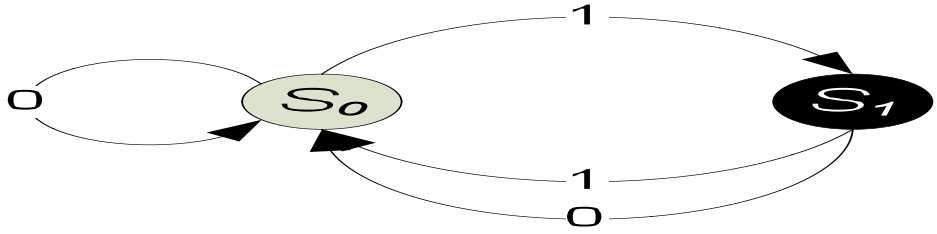


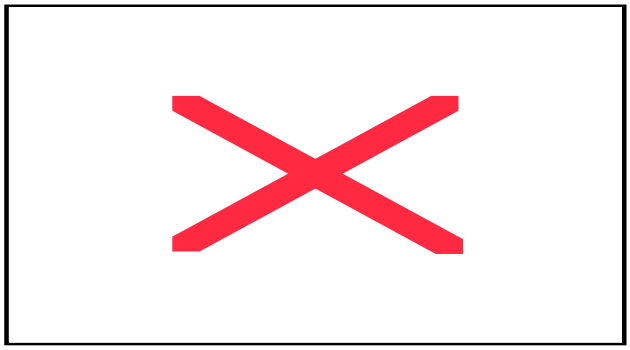
Figure 8: A simple two state automaton.

The automaton accepts {0,1} as a vocabulary, has a state set $\{S_0,S_1\}$, It starts in S_0. The single final state (in black) is S_1. The input sequence 01010101 leaves the automaton in an accepting state and is therefore an accepted sequence. The sequence 01010 on the other hand is rejected. The automaton above accepts any string with an odd number of "1" input symbols. It is said to recognize such sequences. The sequences are called regular if a finite automaton recognizes it. The automaton rejects (i.e. does not recognize) sequences with an even number of "1".

An FSA can be deterministic (as in the above example) on nondeterministic. An example of nondeterministic automata is a Markov chain, a FSA with probabilities assigned to the arcs. The limitations of FSA become evident when a set of input sequences (also called a language) require unlimited memory. Sequences of the form $\{0^n1^n, n \geq 0\}$ require that the FSA remember the number of 0 read, to decide whether the sequence is recognized or not. Such a language is called nonregular. FSA are not able to cope with nonregular languages.

6.2.2. Pushdown automata

Some of the languages that exceed the limitations of FSA (i.e. due to the presence of recursive structures) require a more complex computational model. Context free grammars (CFG) implement such a model. CFG are recognized by pushdown automata (Sipser 2006, Gough 1988). A pushdown automaton M is a 6-tuple $(Q, q_0, A, \Sigma, \Gamma, \delta)$ where



[6]

6.3. Using FSA to split sentences

Designing FSA is simpler than designing PDA. Yet, the possibility of using FSA depends on whether the input (i.e. the sentences) can be guaranteed to consist of a bound number of clauses. This, in turn, implies that the number of words in the sentence is bound. To show that this is the case for written text, an analysis of available corpuses follows.

From a theoretical point of view, a sentence can be extremely long. There is no theoretical limit on the number of words in a sentence. Wikipedia quotes at least four contenders to the title of the longest printed sentence, with up to 469,375 words[7] (!). The modest between the four contenders is a 1,287 word long sentence from Faulkner's "*Absalom, Absalom!*".

In practice, there are several hints that sentence length is generally bound. Sigurd *et al*. (2004) found that the Brown Corpus shows a sentence length with a predicted behavior (as function of *L*, the number of words) of $ë f_{exp} = 1.1 * L^{1} * 0.90L$.

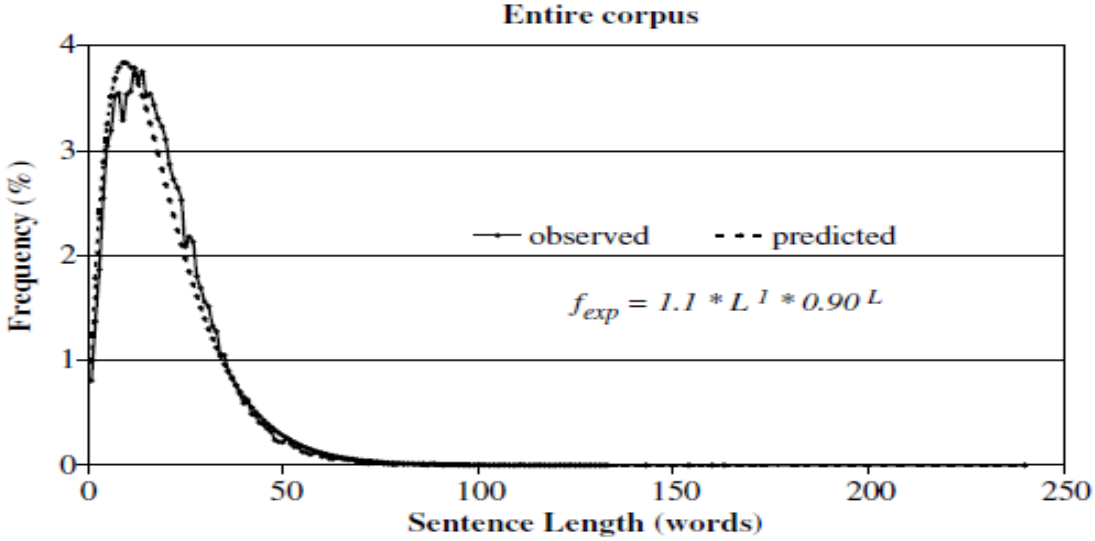


Figure 9: Sentence length distribution (from Sigurd et al. 2004: 14).

The graph shows that the overwhelming majority of sentences has 50 words or less. The number of sentences with 51 to 100 words is very low.

An analysis found at http://hearle.nahoo.net/Academic/Maths/Sentence.html also shows that sentence length is bound.

[6] See Sipser (2006: 115) (Theorem 2.20) for a proof of the theorem stating that "A language is context free if and only if some pushdown automaton recognizes it".

[7] http://en.wikipedia.org/wiki/Longest_English_sentence.

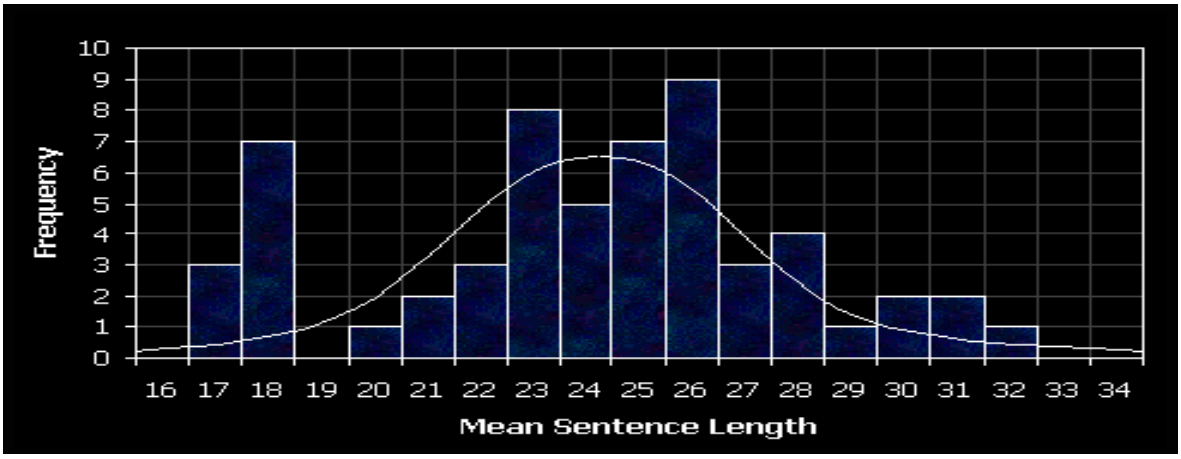


Figure 10: Mean sentence length.

Another study by Elia (2009) shows that the average length of a sentence found in resources such as Britannica and Wikipedia is less than 23. In comparison, sentences in a corpus of academic prose are 24 words long in average. An additional, interesting finding by Elia is that the minimum sentence length is 14.4 words in Britannica and 14.7 in Wikipedia. The maximum sentence length is 32.8 words per sentence in Britannica and 35.8 in Wikipedia. There is no data on academic prose, probably because academic prose is hard to measure (except for a given corpus), while Britannica and Wikipedia are a closed set at any given point in time.

Moreover, Biber *et al.* (1999) suggest, based on the analysis of its annotated corpus, that it would be unusual to find a sentence with more than three levels of embedding in written, academic text (other registers, i.e. CONV or NEWS show lower levels of embedding). The length of clauses is also limited, and therefore there is a clear bound in sentence length.

6.3.1. An example – a clause boundary detection automaton

Figure 11 below shows a simple automaton to detect clause boundaries within written sentences.

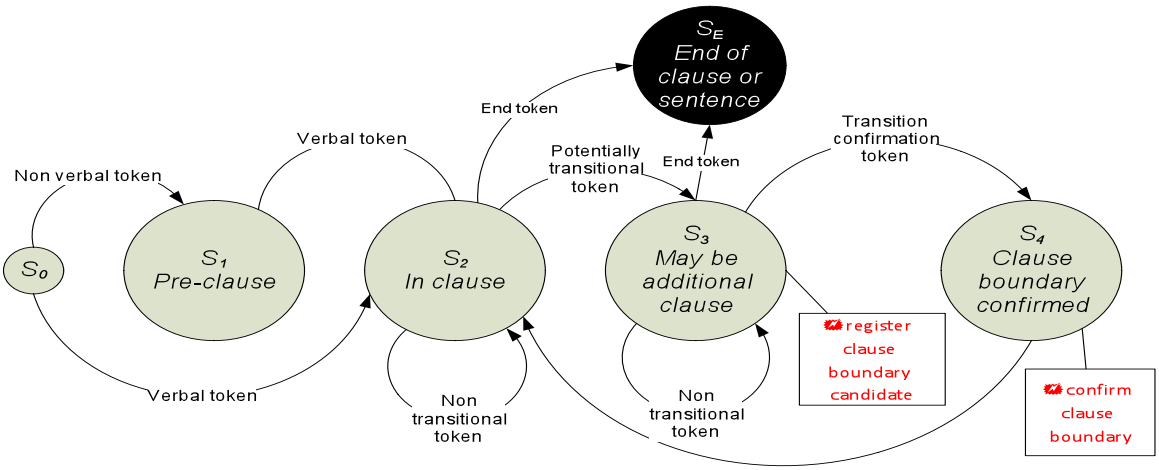


Figure 11: A simple clause boundary detector.

The automaton design stems from structural features observed in English sentences. These sentences consist of tokens (i.e. words or language symbols such as commas, semicolons, etc.). Tokens can be:

- Non transitional tokens (regular words, normally words that are not function words).
- Potentially transitional tokens (comma, semicolon, words such as *and*, *that*, *because*, and potentially any preposition).
- Transition confirmation tokens (i.e. a token that under certain conditions makes clear that a clause transition occurred).
- End tokens, such as a period that is not part of an abbreviation, or the end of the input.

The automaton in the example above (a simple finite state machine) recognizes English sentences (but not all English sentences). The single accepting state of the automaton above appears in black. Any sentence not accepted, as stated before, is rejected (i.e. is not recognized) by the automaton.

The sentence already seen in the context of Antelope and in the formulation of our model can be used as an example of how the automaton works. The sentence is:

(9) "Most of these therapeutic agents require intracellular uptake for their therapeutic effect because their site of action is within the cell"

The right decomposition into clauses is as follows:

[Most of these therapeutic agents require intracellular uptake for their therapeutic effect] because [their site of action is within the cell]

For this sentence in the automaton M $(Q, q_0, A, \Sigma, \delta)$ we have $Q = \{S_0, S_1, S_2, S_3, S_4, S_E\}$, $q_0 = S_0$, $A = S_E$, and Σ = {"Most", "of", "these", "therapeutic", "agents", "require", "intracellular", "uptake", "for", "their", "therapeutic", "effect", "because", "site", action", "is", "within", "the", "cell}.

The token "Most", the beginning of the input, starts at S_0. Being a non verbal token the automaton transitions to S_1. It stays at S_1 with tokens "of", "these", "therapeutic" and "agents", until the token "require" is read. This token ("require") sends the machine into state S_2. The tokens "intracellular", "uptake", "for", "their", "therapeutic" and "effect" are non-transitional tokens. Hence the machine remains in state S_2 until the token "because", a potentially transitional token is read. With it, the automaton transitions to state S_3. At this time the boundary (just before "because") candidate is marked. The following tokens ("their", "site", of" and "action") are non transitional, which leaves the machine in state S_3. The token "is" is a transition confirmation token. This is due to the fact that a clause does not contain more than one lexical verb, so the fact that a new clause seems to start and the presence of a new verb confirms the presence of a new, additional clause. The automaton goes into state S_4 and issues a confirmation on the clause boundary candidate found (before the token "because"). Immediately after the confirmation, with no further action, the automaton reverts to state S_2. The next tokens ("within", "the" and "cell") are non transitional. The automaton then reaches the end of the input, which sends it into S_E. Hence, the automaton found that the sentence consists of two clauses, with the pivotal element being the token "because".

This simple version of the automaton does not allow for the identification of the clauses composing the sentence as coordinate or subordinate. Moreover, it recognizes only a small subset of all the English sentences. In order to enlarge the coverage it is necessary to design the automaton with further details of the English sentence structure, as described in previous sections.

7. The roadmap to a full-fledged semantic content extractor

The final objective of our research is to learn ontologies from text. A necessary step towards the goal is the design of a semantic content extractor based on the grammatical analysis of the English language shown in this paper. To this end, an automaton based on the above grammatical analysis is to be designed. The automaton expected is more sophisticated than the automaton presented in the previous section but remains a finite state automaton, as the English sentences were found to be limited in size and number of clauses.

Upon designing the complete automaton, we will carry a thorough test of its coverage of written English. If the results show a wide coverage of English sentences (as expected), our heuristics-based semantic content extraction approach will be used as the infrastructure towards a linguistics-based framework for ontology learning.

References

Alani, H. *et al*. 2003. Automatic Ontology-Based Knowledge Extraction from Web Documents. *IEEE Intelligent Systems* 18/1: 14-21. [Available At http://eprints.ecs.soton.ac.uk/7396/]

Biber, D. *et al*. 1999. *Longman Grammar of Spoken and Written English*. Harlow: Longman Publications Group.

Buitelaar, P., P. Cimiano, and B. Magnini. 2005. Ontology Learning from Texts: An Overview. In P. Buitelaar, P. Cimiano and B. Magnini (eds.), *Ontology Learning from Text: Methods, Evaluation and Applications*. 3-12. Amsterdam: IOS Press.

Byrne, K. 2006. *Relation Extraction for Ontology Construction, Institute for Communicating and Collaborative Systems*. PhD Dissertation. School of Informatics, University of Edinburgh. [Available at http://homepages.inf.ed.ac.uk/s0233752/docs/phdproposal.pdf].

Chaumartin, F.R. 2005. *Conception et réalisation d'une interface syntaxe / sémantique utilisant des ressources de large couverture en langue anglaise*. Master de Recherche en Linguistique et Informatique. Université Paris 7. [Available at http://www.proxem.com/download/SyntaxeSemantique_2005_09_14_M2R.pdf].

Chen, E. and W. Gaofeng. 2005. An Ontology Learning Method Enhanced by Frame Semantics. *Proceedings of the ISM*: 374-382. [Available at http://doi.ieeecomputersociety.org/10.1109/ISM.2005.32].

Cimiano, P. 2006. *Ontology Learning and Population from Text.Algorithms, Evaluation and Applications*. New York: Springer.

Elia, A. 2009. Quantitative Data and Graphics on Lexical Specificity and Index of Readability: The Case of Wikipedia. *Revista Electronica de Linguistica Aplicada* 8: 248-271.

Feldman R. and J. Sanger. 2007. *Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. New York: Cambridge University Press

Fillmore, C.J. 1976. Frame semantics and the nature of language. *Annals of the New York Academy of Sciences* 280: 20-32.

Fillmore, C.J., C. Johnson and M. Petruck. 2003. Background to FrameNet. *FrameNet and Frame Semantics*. *Special issue of International Journal of Lexicography* 16/3: 235-250.

Gómez-Pérez, A. and D. Manzano-Mach, eds. 2003. *Deliverable 1.5: A Survey of Ontology Learning Methods and Tools.* [Available at http://www.sti-innsbruck.at/fileadmin/documents/deliverables/Ontoweb/D1.5.pdf]

Gough, K.J. 1988. *Syntax Analysis and Software Tools*. New York: Addison Wesley Publishing Company.

Gruber, T.R. 1993. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition* 5/2: 199-220. [Available at http://ksl-web.stanford.edu/KSL_Abstracts/KSL-92-71.html]

Halpin, T. and T. Morgan. 2001. *Information Modeling and Relational Databases.* Burlington, Mass: Morgan-Kauffman.

Harris, Z. 1968. *Mathematical Structures of Language*. New York: John Wiley & Sons.

Ido D., O. Glickman and B. Magnini. 2006. The PASCAL Recognising Textual Entailment Challenge. *Lecture Notes in Computer Science*, 3944: 177-190. [Available at http://www.cs.biu.ac.il/~glikmao/rte05/dagan_et_al.pdf].

Jones, G.M., L.E. Horning and J.D. Morrow. 1922. *A High School English Grammar*. Toronto and London: J.M. Dent & Sons.

Maedche, A. and S. Staab. 2000. Discovering conceptual relations from text. In W. Horn, (ed.), *Proceedings of the 14th European Conference on Artificial Intelligence*. 321-325. [Available at http://www.aifb.uni-karlsruhe.de/WBS/Publ/2000/ecai_amaetal_2000.pdf].

Masolo, C. *et al*. 2003. *WonderWeb Deliverable D18: Ontology Library* [Available at http://wonderweb.semanticweb.org/deliverables/documents/D18.pdf].

Navigli, R. and P. Velardi. Learning Domain Ontologies from Document Warehouses and Dedicated Websites. *Computational Linguistics* 30/2: 151-179. [Available at http://lcl2.di.uniroma1.it/termextractor/help/CL_2004_Navigli_Velardi.pdf].

Niles, I. and A. Pease. 2001. Towards a standard upper ontology. *Proceedings of the international Conference on Formal ontology in information Systems*: 2-9.

Ontology Learning from Text at ECML/PKDD (European Conference on Machine Learning), October 3rd 2005, Porto, Portugal – with: Philipp Cimiano: AIFB, University of Karlsruhe, Germany; Marko Grobelnik: Jozef Stefan Inst., Ljubljana, Slovenia; Michael Sintek: DFKI GmbH, Germany. http://www.aifb.uni-karlsruhe.de/WBS/pci/OL_Tutorial_ECML_PKDD_05/ECML-OntologyLearningTutorial-20050923.pdf.

Preisler, B. *1997. A Handbook of English Grammar on Functional Principles.* Aarhus: Aarhus University Press.

Reinberger, M.L and W. Daelemans. 2004. Unsupervised Text Mining for Ontology Extraction: An Evaluation of Statistical Measures. *Proceedings of LREC04*: 491-494.

Reinberger, M.L., and P. Spyns. 2005. Unsupervised text mining for the learning of dogma-inspired ontologies. In P. Buitelaar, P. Cimiano and B. Magnini (eds.), *Ontology Learning from Text: Methods, Evaluation and Applications*. Amsterdam: IOS Press.

Sanderson M. and B. Croft. 1999. Deriving concept hierarchies from text. *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*: 206-213.

Schuler, K.K. 2005. *VerbNet - a broad-coverage, comprehensive verb lexicon*. PhD dissertation. University of Pennsylvania. [Available at http://verbs.colorado.edu/~kipper/Papers/dissertation.pdf].

Sclano, F., and P. Velardi. 2007. TermExtractor: a Web Application to Learn the Shared Terminology of Emergent Web Communities. *Proceedings of the 3rd International Conference on Interoperability for Enterprise Software and Applications*. (Forthcoming) [Available at http://lcl2.di.uniroma1.it/termextractor/help/I-ESA_2007_Sclano_Velardi.pdf]

Shah, P. *et al*. 2006. Automatic Population of Cyc: Extracting Information about Named-entities from the Web. *Proceedings FLAIRS*: 153-158. [Available at http://www.cyc.com/doc/white_papers/FLAIRS06-AutomatedPopulationOfCyc.pdf].

Sigurd, B., M. Eeg-Olofsson and J. Van Weijer. 2004. Word length, sentence length and frequency – Zipf revisited. *Studia Linguistica* 58: 37-52.

Sipser, M. 2006. *Introduction to the Theory of Computation*. London: International Thomson Publishing.

Sowa, J. F. 1990. Crystallizing theories out of knowledge soup. In Z. W. Ras and M. Zemankova (eds.), *Intelligent Systems: State of the Art and Future Directions*. 456-487. London: Ellis Horwood Ltd.

Sowa, J.F. 2000. *Knowledge Representation: Logical, Philosophical, and Computational Foundations.* Pacific Grove, CA: Brooks/Cole Publishing Co.

Sowa, J.F. and A.K. Majumdar. 2003. Task-Oriented Semantic Interpretation. [Available at http://www.jfsowa.com/pubs/tosi.htm].

Spyns, P. 2005. Adapting the Object Role Modelling method for Ontology Modelling. In M.S. Hacid, Z. Ras and S. Tsumoto (eds.), *Proceedings of the 15th International Symposium on Methodologies for Intelligent Systems (ISMIS 2005*): 276-284

Spyns, P. and J. De Bo. 2004. Ontologies: a revamped cross-disciplinary buzzword or a truly promising interdisciplinary research topic? *Linguistica Antverpiensia*, NS 3: 279-292.

Studer, R., V.R. Benjamins, and D. Fensel. 1998. Knowledge engineering: Principles and methods. *Data and Knowledge Engineering* 25: 161-197.

Turnitsa, C.D. A. Tolk. 2006. *Ontology Applied - Techniques employing Ontological Representation for M&S, Fall Simulation Interoperability Workshop, Simulation Interoperability Standards Organization*. IEEE CS Press.