NATURAL LANGUAGE PROCESSING

LEONARDO LESMO

Dipartimento di Informatica – Università degli Studi di Torino

MARIA TERESA PAZIENZA

Dipartimento di Informatica, Sistemi e Produzione – Università di Roma Tor Vergata

1 Introduction

The study of language is fascinating for a number of reasons. First of all, human languages are intimately connected to the knowledge of the world and to our living in the world. We are able to talk about the things we can touch, about things that happen, about our emotions, desires, sensations. Second, there are plenty of languages, each one with specific features, but all with a universal common basis. Third, language can be, and has been, studied from many perspectives, ranging from linguistics proper to psychology, from mathematics to philosophy, from computer science to biology. Finally, language has a lot of interesting and economically promising applications, so that its inherent scientific appeal is supported by the practical value, as we will show in the third section of this article. But language is very complex. The fact that it is one of the most prominent (perhaps "the" most prominent) distinguishing features of humans can be taken as showing that only the human brain has reached the level of development able to cope with such an intricate object. Consequently, it has been attempted to split the full task of understanding how language works into subtasks, by identifying "modules" that take care of different aspects of the problem.

Phonetics is a study of sounds from an acoustic point of view, independently of their function in the specific languages. Conversely, *phonology* is the study of how sounds are organized and used in different existing communication systems based on language. In other words, phonology analyses the sound patterns of a particular language in order to determine which phonetic sounds are significant and how these sounds are interpreted by the native speaker.

Independently of the way the basic units of language, i.e. words, are expressed (i.e. by voice or by writing), it is assumed that they have an internal structure. *Morphology* (the study of "form") aims at identifying the rules governing this structure. For instance, the first word of this paragraph can be assumed to be composed of four parts ("in", "depend", "ent" and "ly"), each of which contributes not only to the structure, but also to the meaning of the word. Morphology is important since it governs the encoding of information useful both for the syntactic and for the semantic component. With respect to syntax, the syntactic number (cat vs. cats) and the tense of verbs (count vs. counted) are examples of syntactic-semantic pieces of information often encoded by means of ßexional afÞxes (-s and -ed in the examples) attached to a word *root* ("cat" and "count"), while *derivational afÞxes* enable to "derive" new words from other words.

Syntax is one of the most deeply studied sub-fields of computational (and classical) linguistics. This is probably due to the feeling that words are subsidiary to the construction of the fundamental element that conveys linguistic meaning, i.e. the sentence. The first thing to do is to establish how sentences are made up, i.e. how words are assembled into sentences. This sounds rather similar to what we have seen for the morphemes composing the words. A relevant difference is that while the inventory of words (including all their variations due to the morphological processes) is finite and relatively stable, the number of sentences in a language is potentially infinite. So, we can have a list of all words in a language (a dictionary), but not a list of all sentences of a language. A major impulse to the study of syntax was given by Chomsky, who devised a method for describing an infinity of expressions by means of a finite set of rules (generative grammars).

Semantics is the study of meaning. People use language for exchanging information, and people must be able to reason on these pieces of information. But this means that they should be represented in a way that enable humans to draw inferences, choose lines of behaviour, react in some way to what they have come to know. However, language is ambiguous, so that most sentences may have different meanings. Consequently, there have been attempts to develop methods for representing meaning unambiguously. The major achievement in this area was the development of *formal logics*, that, at least in part, was developed in the

last century, starting from the works of Frege and Russell, with the goal of providing unambiguous representations enabling formally correct reasoning. The correspondence between a sentence and its meaning is largely based on the structures studied by the syntax: the syntactic structures guide the semantic interpretation process via the *principle of compositionality*: the meaning of a part of a sentence can be derived on the basis of the meaning of its subparts (where what is a part is defined by the syntax). Of course, this implies that the entire meaning of the sentence depends on some elementary building blocks, i.e. the words: the study of the meaning of words is called *Lexical Semantics*.

Pragmatics is the study of use. We use the language (as most other activities we carry on) to achieve our goals: we do things with words [2]. Sentences are not only abstract linguistic objects, but are acts: since speech is the basic way to interact with language, they have been called *speech acts*. What is needed is an explanation about why a given sentence, with a given meaning, has been chosen in a given context by the speaker to pursue her/his goals. In complex tests or dialogues, we must justify why a given sequence of sentences has been chosen, i.e. which relations exist among them: this is the goal of *rethorics*. Pragmatics has recently been related to the study of planning, which could provide interesting insights on its principles.

2 Language and Artificial Intelligence

In this section, we aim at building a bridge between theory and applications. After linguistics and philosophy have explained us what is language, we need a set of computational tools that enable a computer to understand language. In order to implement such tools, there are two main requirements: knowledge and processes. Let us take syntax as an example. There are two computational tools that are devoted to exploit syntactic knowledge: the parser and the generator. The first of them must take as input a sentence (whose words have already been analysed from a morphological point of view), and decide which is its internal structure, i.e. how the composing words are related to each other. Viceversa, the generator should take a structure and produce a (linear) sequence of words. But this is not enough: a parser must use a grammar, that describes the language under analysis, and the grammar can be more or less complete. So, we can have the best processes for parsing and generation, but, with a limited grammar, we will be able to handle just a minimal subset of the full set of sentences belonging to the language: this is the *coverage* problem. This section addresses the way the knowledge (at different levels) can be expressed and the way the processes can be implemented.

2.1 Phonetics, Phonology, and Speech

The studies on the analysis of linguistic sounds are partially separated from the ones of other branches of natural language. This is due to the fact that they involve a background on acoustics that falls within the realm of engineering. Moreover, the interpretation of speech sounds poses significant problems associated with the continuous flow of sounds (the separation in words is not explicitly marked) and with the acoustic similarity of some sounds (e.g. the ones associated with the *phonemes* /p/ and /t/), thus producing a lot of ambiguity. The classical introduction to this topic is [25]. Even if we disregard the goal of having a full comprehension of the speech input, still the problem is so hard that it had to be split into tasks, in a way much similar to the one we have seen for NLP in general. The first step is to *transform* the continuous waveform (speech) into a digitised counterpart. This applies to any sound we want to encode in a digital form (e.g. music), but special solutions have been adopted for speech. Then, the "spectral" features are *extracted* from the digitised sound. These features concern the frequency distribution of the sound components. The third step involves the *classification* of a temporal segment (frame) of the input, on the basis of its spectral characteristics. The classification is made in terms of phones, a lower-level counterpart of the classical phonemes. This step is often performed by means of Neural Networks. In the fourth step, the sequence of recognized phones is *matched* against the known words. This is the speech analogous of dictionary lookup for text. In order to cope with the inherent ambiguity of phone classification, a "best match" must be obtained between the recognized phones and a set of pre-stored phone sequences associated with the known words. This is carried out by means of an efficient storage of the set of words (the so-called language model) via techniques as the Hidden Markov Models (HMM) and by means of optimised matching methods.

Note that speech recognition stops here, i.e. in the phase where should start the standard syntactic and semantic analysis. Often, the limitations in the recognition phase make impossible to use a standard grammar, so that simpler domain-dependent grammars are encoded as finite state automata. Around 2000, W3C has proposed a standard for speech interfaces, and an extension of XML devoted to the description of speech grammars (VXML: http://www.w3.org/TR/voice-intro/) where VXML stands for Voice XML. Note that in this short overview, we focused on speech recognition. Many efforts are also devoted to speech synthesis.

2.2 Morphology

Morphology analysis concerns the internal structure of the words. All the words are constituted by morphemes, where each morpheme represent one unit of meaning. The morphological analysis of a word is the process that takes as input the word and returns its morphological structure. The stem of a word is the morpheme that brings the "main" meaning of the word, the remanent morphemes in the

word, called affixes, modify and specify this main meaning. The complexity of the morphological structures can be very different among languages, however most morphological systems rely on the finite-state devices since they can encode the *lexicon*, i.e. the list of stems and affixes, the *morphotactics*, i.e. allowable morpheme sequences, the *orthographic rules*, i.e. the changes that occur in a word when two morphemes combine. In particular *two-level morphology* is based on finite-state transducers, that are an extension of finite-state automata that can generate output symbols. The main idea is that by using a finite-state transducer we transform a word (*the surface level*) into the sequence of morphemes (*the lexical level*) [19]. Moreover, finite-state devices can be also used to efficiently store large dictionaries that are widely used in spelling correction systems.

2.3 Syntax and Parsing

The current efforts on syntactic analysis are largely based on the work of Chomsky, which started in the mid-fifties [6] and evolved throughout the years to the theory of Minimalism. Syntactic knowledge is based on *phrase structures* [16] (also called *constituents*), which are recursively defined; consequently, a finite set of *phrase structure rules* is able to describe an infinite set of *sentences* belonging to the language. Chomsky's approach is *generative*, in the sense that the rules enable one to generate the set of sentences of the language. In AI, more attention has been paid to the inverse process, i.e. *parsing*, whereby one devises the internal (hidden) structure of a sentence in order that it be compliant with the rules. There are two main goals here: the first one is the definition of a grammar of a language that describes all and only the sentences belonging to it. The second one is the design of algorithms able to use the grammar to extract the structure (usually a tree) in an effective way. The two goals are in contrast: in order to have a precise description of the language, we need a powerful grammar, but the more powerful is the grammar, the more expensive is parsing from a computational point of view. It currently seems that it is possible to have grammars that correctly describe the natural languages and are parsable in polynomial time [29]. In order to get these performances, a number of *syntactic formalisms* (i.e. ways to write grammars) have been defined. They include, but are not limited to, Generalized Phrase Structure Grammars (HPSG) [13], Lexical Functional Grammars (LFG) [4], Combinatory Categorial Grammars [28], Tree Adjoining Grammars [17]. The first goal (complete coverage of a language) is far from being achieved. Historically, grammars are developed manually: some linguists choose a formalism, then sit at a table and write down the grammar, according to their intuition (as *native speakers*) on the set of correct sentences. This approach is very expensive and error-prone, so that, thanks to the AI results on machine learning, *grammar induction* has been proposed as a viable alternative: in this case, it is assumed that a set of sentences are given together with their syntactic structure. Then, a learning algorithm, usually based on statistical techniques, takes as input the pairs and produces the associated grammar [7]. This method is widely used, though it depends on the availability of a treebank, whose construction requires a huge human effort. However, treebanks have the advantage that they cover the language as it is actually written (or spoken) while the manual approach tends to produce grammars that are more *prescriptive*, i.e. that describe language as it *should* be written or spoken. It must be observed that many application-oriented systems use different approaches to parsing, which are less clear from a formal point of view, but produce a parse tree with less effort (often with some degree of errors). Among these methods, often referred to as *shallow*, *chunk parsing* must be mentioned [1]. Finally, it is worth noting that a representation other than phrase structure is widely studied nowadays, i.e. *dependency grammars* [23].

2.4 Semantics and interpretation

The syntactic structure of a sentence, can be used as the basis for the translation of the sentence into a meaning representation. Since the principle of compositionality (see 1) is at work here, we need knowledge of two types. First, we must know how to put together two or more "pieces" of meaning representations in order to obtain the composite meaning; second, we must know what the individual words mean, in order to be able to start up the process from the elementary units. The fundamental work on the semantics of language is the one of [24], who adopted a powerful logical formalism (intensional logics) to fulfil the task. Nonetheless, Montague's seminal work remained largely theoretical, while some more practical approaches emerged (as Discourse Representation Theory [18], which encompasses both the specific problem of single-sentence interpretation and the problem of the integration of a sequence of sentences into a single integrated representation). However, it must be observed that no practical system is currently able to carry out a full semantic interpretation. This depends on the fact that semantics is much more knowledge intensive than syntax, since it involves knowledge of the meaning of words, which, in turn, involves a representation of the world that surrounds us. Today, most practical activities in semantics are devoted to the development of large repositories of word and world knowledge: WordNet [11] is a widely used lexical KB based on the concept of synonym set and where meaning relations (as synonymy, hyponymy, antonymy, etc.) are represented explicitly. Plenty of information can be obtained from the official site of the Global WordNet Association (http://www.globalwordnet.org/). With respect to world knowledge, the research on semantics is being linked to the AI area of knowledge representation via the study of ontologies, which should constitute the basis not only for

semantic interpretation, but also for software interoperability (Semantic Web) [3]. An ontology (note that, differently from its use in philosophy, the word is used in AI as countable) is an explicit conceptualisation of the world [14]. Of course, since the task is huge, we usually talk about the conceptualisation of given specific domains (i.e. parts of the world), though efforts are on the way to propose top-levels (or upper-levels) able to provide the overall framework into which all domain-specific ontologies can be integrated; see the Suggested Upper Merged Ontology, at the SUO IEEE site http://suo.ieee.org/, or the Dolce (Descriptive Ontology for Linguistic and Cognitive Engineering) proposal [12]. The scenario offered by the Semantic Web is characterized by a huge amount of documents and users willing to access them. Both the multilingual aspects which characterize the (Semantic) Web and the demand for more easy-to-share forms of knowledge representation, being equally accessible by humans and machines, push for a linguistically motivated approach to ontology development. Ontologies should thus express knowledge by associating formal content with explicative linguistic expressions, possibly in different languages. By adopting such an approach, the intended meaning of concepts and roles could become more clearly expressed for humans, while content mediation between autonomous agents still requires further analysis at methodological level.

2.5 Pragmatics

The general issues of pragmatics have been faced just in part from the computational point of view. Some efforts are currently under way to build treebanks annotated with rethorical relations, usually according to Rhetorical Structure Theory [20], with temporal information, with affective data, and much more. However, most activities concerned with the field of pragmatics are related to the implementation of practical dialogue systems, since they are of paramount importance in driving the interaction (especially in case of speech systems) with the user. The well known Eliza system [31] is a first example of implemented dialogue system. The Dialogue Management Systems can be classified according to the type of interaction: System-driven or Mixed-initiative. In the first case, the system asks question in order to clearly understand the user's interests, and the user must answer them. In the second case, the user has some freedom in asking questions independently of the system expectations. Most currently available Dialogue Management Systems are based on a fixed sequence of steps, usually modelled via finite state automata, or via production rules, but some more flexible approaches do exist [22]. A lot of practical and demonstrable systems can be found in the site compiled and maintained by Michael McTear (http://www.infj.ulst.ac.uk/ cbdg23/dialsite.html)

2.6 A Note on the architecture of NLP systems

The presentation in this and in the previous section has assumed that the processing of language can be actually carried out as split in well-defined sub-modules which operate in sequence. The idea that parsing starts after the morphological analysis of the input words has finished, or that, before carrying out the semantic interpretation, we must wait until we have at disposal the full parse tree of the input sentence is valid from the point of view of system developers, but cannot be defended on theoretical grounds. The idea is that the modules must be able to provide some feedback, along the way, to help the *previous* modules to perform their job. It is clear that semantic information can be useful to *prune* in advance some syntactic trees, since they (though syntactically correct) have no meaning consistent with our world knowledge. This could reduce the amount of ambiguity associated with syntactic processing, and can contribute to speed up the whole analysis. On the other hand, this *interleaving* of module operations poses significant problems in software engineering, so that it has been only partially exploited. Among others, [8] defines a software infrastructure for NLP as a set of common models for the representation, storage and exchange of data in and between processing modules in NLP systems. Such a framework could either support a suitable linguistic description or make available at a computational level relevant portions of the linguistic abstraction required by a number of applications. Nonetheless, a significant amount of research is being carried out on the way people actually process sentences (*psycholinguistics*) in order to get from them useful insights about the possible organization of NLP systems. In the engineering phase of a NLP system, it is important to define the nature and format of lexical information. A lexicon may be simply defined as the component of a NLP system that expresses linguistic information about words; a more comprehensive one describes the lexicon as "an association between surface forms and linguistic information", thus covering linguistic principles and application oriented purposes. A few linguistic theories (as HPSG [26]) postulate lexical descriptions where surface forms are related to a complex structure of linguistic principles (from morphosyntactic features to semantic constraints), aimed to support a variety of very complex inferences. Lexicon is considered as a comprehensive knowledge repository where representation and content support several deductive processes (inheritance, forward/ backward reasoning, constraint propagation) and linguistic processes. Among them, we must mention POS (Part Of Speech) taggers [5], whose task is to provide the parser with just one hypothesis about the category (noun, verb,) of each input word. For instance, in "I love her", "love" is *tagged* as a verb, while in "I appreciate your love", it is tagged as a noun. Also robust approaches to parsing are dependent on (partial) lexical information: a subcategorization lexicon may be used to deal with PP-attachment. Named Entity (NE) recogni-

tion grammars also rely on a variety of lexical knowledge: Gazetteer entries and trigger words are typical lexicalized information. Lexicons used for these tasks are broadly general, although NE catalogues or proper noun formation rules are not totally domain-independent. Source lexical information is also adopted for word semantic disambiguation and classification. To cover linguistic phenomena (that are dynamic in nature) both at a wide extent and in specialized languages (with specific jargon, style and phenomena), large scale corpora have been analyzed and cover the study of general phenomena in a language (by coupling several sublanguages in single samples) or narrower (i.e. domain specific) aspects (by selecting sampling of given sublanguages). Pervasive language ambiguity has been approached by inducing preference models (or rules) either from untagged or from previously disambiguated samples. Probabilistic approaches have often been proposed according to the availability of large scale controlled or raw data as well as of complex training data set (among others large collection of lexical resources, as Penn TreeBank [21]). Finally, as we will see in the next section, most application-oriented systems do not perform a full interpretation of the input text, so that they do not include all modules and their architecture is designed according to the task at hand.

3 NLP for application development

In the wide and heterogeneous context of (sometimes ungrammatical) human language communication, the interest in robust systems to automatically process unstructured textual data is continuously growing in order to improve both text readability and understanding. The web is currently the scenario in which the development of intelligent systems for tasks (among others) like Information Retrieval (IR) , Information Extraction (IE), Question Answering (Q/A), Textual Entailment (TE) etc. is achieving valuable results. All of them require linguistic knowledge stratified across several levels of processing. Such a knowledge spans syntactic-semantic patterns for locating facts or events in texts, domain-specific word or concept classes for semantic generalization, specialized lexicon of terms, etc.

While Information retrieval is the task of selecting relevant documents from a text corpus or collection in response to a user's information need [27], Information Extraction can be seen as the process by which a system, by processing textual data sets in a linguistically motivated fashion, is able to derive a structured representation of (part of) their content [15]. This constitutes its distinguishing features with respect to IR (where both source and output information has the same format) and data mining (where source information is characterized by a more precise set of structures); such a feature motivates the pervasive influence that linguistic models and methods assumed for IE as well as the fact that along the time IE systems get a central role in highlighting NLP successful approaches.

There are a few tasks implemented in all IE systems:

NE: named entitiy recognition (searching and classification of names, locations,)

CO: coreference resolution (highlights in texts identity relations between entities)

TE: template element construction (descriptive information added to results of NE by using CO.

TR: template relation construction (relations among TE entities are identified)

ST: scenario Template production (TE and TR results are used to fill specific event scenarios)

Is there a difference for Information Extraction from web documents and 'traditional' ones? The main difference is in the presence of documents' structure: in fact, traditional information extraction only applies on textual data, while in the web the document structure can be used for several purposes, from inferring the content (or document) type to extracting relevant information for template filling (consider for example the case of the paper's author name: it is located in a specific position into the document).

Question Answering deals with the problem of identifying the answer to a question by accessing large collections of documents. Q/A international competitions focused mainly on English language. Recently, multilingual Q/A is emerging related to the wide interest in developing research covering further languages. Q/A real systems are dedicated systems with knowledge in a specific application domain and able to identify, in documents, fragments of texts containing the content required by the questions (all is provided in natural language). The answer to a question must be provided in real-time. Possible questions may assume different lexicalizations: the focus is identified by initial lexical items that could be one among: how, who, why, what, where, when. For each Q/A system we can identify:

Question type: idiomatic categorization of questions for purposes of distinguishing between different processing strategies and/or answer formats.

Answer type: class of objects involved by the question; generally such objects are related to the named entities (NE) identified into the question. *Question focus*: property/entity to which the question is interested ("Where is located Coliseum?").

Question topic: object/event the question is about (in the question "What is the height of Monte Bianco?", the focus is the height, the topic is the Monte Bianco mountain).

Candidate passage: any length text fragment (short as a sentence or the entire document) retrieved by the search engine in response to a question.

Candidate answer: it identifies in the context of the question, a short fragment of text that could appear also in the answer.

Q/A systems may be classified accordingly to sharpness of their behaviour:

- *Slot-Þlling*: only very simple questions may be replied (in an IE fashion)

- *Limited-domain*: questions may be more complex, while the knowledge is limited to a specific area of competences.

- *Open-domain*: expected to reply to complex and structured questions, they integrate IR, IE and NLP techniques.

Among other resources, Q/A systems involve machine learning algorithms, gazetteers, NE taggers, Part of Speech taggers, Parsers, WordNet, Stopword list, domain terminologies, etc.

Textual entailment has been recently defined as a common solution for modelling language variability in different NLP tasks [9]. Textual Entailment is formally defined as a relationship between a coherent text T and a language expression, the hypothesis H. T is said to entail H (T $\rightarrow$ H) if the meaning of H can be inferred from the meaning of T. An *entailment function* e(T,H) thus maps an *entailment pair* T-H to a true value (i.e., *true* if the relationship holds, *false* otherwise). Alternatively, e(T,H) can be also intended as a probabilistic function mapping the pair T-H to a real value between 0 and 1, expressing the confidence with which a human judge or an automatic system estimates the relationship to hold. For example "*Yahoo acquired Overture*" entails "*Yahoo owns Overture*". Since the task involves natural language expressions, textual entailment is more difficult than logic entailment. In textual entailment, the only restriction on T and H is that they must be meaningful and coherent linguistic expressions: simple text fragments, such as noun phrase or single words, or complete sentences. In the first case, entailment can be verified simply looking at synonymy or subsumption relation among words. For example the entailment *cat* $\rightarrow$ *animal* holds, since the meaning of the hypothesis (*an animal exists*) can be inferred from the meaning of the text (*a cat exists*). In the latter case, deeper linguistic analysis are required, as the sentential expression expresses complex facts about the world: here is where Textual Entailment gets really interesting and complicated. TE may appear as *paraphrasing* and *strict entailment*.

- *Paraphrase*: the hypothesis h carries a fact that is also in the target text t but is expressed with different words. For example "Yahoo acquired Overture" is a paraphrase of "Yahoo bought Overture".

- *Strict Entailment*: target sentences carry different fact, but one can be inferred from the other. For example, we have strict entailment between "Yahoo acquired Overture" $\rightarrow$ "Yahoo owns Overture". In fact, the relation does not depend on the possible paraphrasing between the two expressions but on an entailment of the two facts governed by acquire and own. Whatever the form of textual entailment is, the real research challenge consists in finding a relevant number of textual entailment prototype relations such as: "X acquired Y" entails "X owns Y" , "X acquired Y" entails "X bought Y". Such patterns can then be used to recognise entailment relations in texts.

Several applications like Question Answering (QA) and Information Extraction (IE) strongly rely on the identification in texts of fragments answering specific user information needs. For example, given the question: "*Who bought Overture?*", a QA system should be able to extract and return to the user forms like "*Yahoo bought Overture*", "*Yahoo owns Overture*", "*Overture acquisition by Yahoo*", all of them conveying equivalent or inferable meaning. In order to disentangle to problem of Textual Entailment, different type of knowledge are needed. In fact, the entailment relation can be linguistically expressed at different levels: surface, lexical, syntactic, semantic or even pragmatic. In this view any type of linguistic, ontological or commonsense resource could be useful, such as WordNet, thesauri, domain ontologies, lexical-semantic databases, commonsense repository, etc.

4 Conclusions

As we said, language is fascinating; this short survey stresses two more points: language processing is extremely complex and has a large number of possible applications. The research on this topic has moved in two different directions: study of the theoretical principles governing language; development of systems that can carry out useful tasks. The overall feeling is that the mutual influence of the two fields has been only partial: the applications have exploited the theory as a general background, but without adopting the technical tools developed in the different areas. Rather, they have developed their own tools, according to their needs. On the contrary, the theory has taken advantage of the practical results mainly in terms of the money that is moving around NLP, thanks to the practical results. This situation can hardly be modified unless there will be a convergence between methods for deep analysis and methods for shallow analysis. Before this can happen, we must wait the development of really large repositories of interchangeable semantic knowledge. Only at that time true Natural Language *Understanding* can be achieved, which was one of the original goals of AI as set forth at the Dartmouth conference. We have travelled a long way toward this goal; each step has shown us how difficult is the climbing and how far is the top. But we are a bit closer now than we were a few years ago.

REFERENCES

[1] S.P. Abney. Parsing by Chunks. In R.C. Berwick, S.P. Abney and C. Tenny (eds.), *Principle-Based Parsing: Computation and Psycholinguistics*. Kluwer, Dordrecht, 1991.

[2] J.L. Austin. *How to Do Things with Words*. Oxford University Press, 1962.

[3] T. Berners-Lee et al. The Semantic Web. *Scientific American 284*, 2001.

[4] J. Bresnan (ed.). *The Mental Representation of Grammatical Relations*. MIT Press, Cambridge Mass, 1982.

[5] E. Brill. Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging. *Computational Linguistics 21*, 543-566, 1995.

[6] N. Chomsky. *Syntactic Structures*. Mouton, The Hague, 1957.

[7] M.J. Collins. Three generative, lexicalised models for statistical parsing. *ACL97*, 16-23, 1997.

[8] H. Cunningham et al. Software infrastructure for natural language processing, *Proc. 5th Conf. on Applied Natural Language Processing*, Morgan-Kaufmann, 1997.

[9] I. Dagan et al. Probabilistic textual entailment: generic applied modelling of language variability, *Proc. Workshop on Learning Methods for Text Understanding and Mining*, Grenoble, 2004.

[10] M. Dean and G. Schreiber (eds.). OWL Web Ontology Language Guide. *W3C Recommendation*, 10 February 2004.

[11] C. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT press, Cambridge Mass, 1998.

[12] A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, L. Schneider. Sweetening Ontologies with DOLCE. In A. Gomez-Perez, V.R. Benjamins (eds.) *Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, Springer Verlag, 166-181 (2002).

[13] G. Gazdar, E. Klein, G. Pullum, and I. Sag. *Generalized Phrase Structure Grammar*, Blackwell, 1985.

[14] T. Gruber. Toward principles for the design of ontologies used for knowledge sharing, in N. Guarino & R. Poli (eds.) *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Kluwer, 1993.

[15] J. Hobbs. The Generic Information Extraction System. *Proc. 5th Message Understanding Conference (MUC-5), J. Artificial Intelligence*, 87-91, Morgan Kaufmann, 1993.

[16] R.S. Jackendoff. *X Syntax: A Study of Phrase Structure*. MIT press, Cambridge Mass, 1977.

[17] A.K. Joshi, Leon S. Levy, and Masako Takahashi. Tree adjunct grammars. *Journal Computer Systems Science 10*, 1975.

[18] H. Kamp, U. Reyle. *From Discourse to Logic*. Kluwer Academic, Dordrecht, 1993.

[19] L. Karttunen. KIMMO: A general morphological processor. In *Texas General Linguistics*, University of Helsinki, 1983.

[20] W.C. Mann and S.A. Thompson. Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. *Text 8*, 243-281, 1988.

[21] M. Marcus et al. Building a large annotated corpus of English: the Penn Treebank, *Computational Linguistics 19*, 1993.

[22] M.F. McTear. Spoken Dialogue Technology: Enabling the Conversational Interface. *ACM Computing Surveys 34*, 90-169, 2002.

[23] I.A. Melcuk. *Dependency Syntax: Theory and Practice*. State University of New York Press, Albany, 1988.

[24] R. Montague. The Proper Treatment of Quantification in Ordinary English. In R. Thomason (ed.) *Formal Philosophy: Selected Papers of Richard Montague*, 247-270, Yale Univ. Press, New Haven, CT, 1973.

[25] L.R. Rabiner, B.-H. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, 1993.

[26] I.A. Sag et al. *Syntactic Theory, a formal introduction*. CSLI Publication, Stanford, 2003.

[27] A. Smeaton. Information retrieval: still butting heads with Natural Language Processing, in *Information Extraction, a multidisciplinary approach to an emerging information technology*, Springer Verlag, 1997.

[28] M. Steedman. *The Syntactic Process*. MIT Press, Cambridge Mass, 2000.

[29] K. Vijay-Shanker, A.K. Joshi, and D. Weir. The convergence of mildly context-sensitive grammatical formalisms. In P. Sells, S. Shieber, and T. Wasow (eds.), *Foundational Issues in Natural Language Processing*. MIT Press, Cambridge MA, 1990.

[30] P. Vossen. *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*, Kluwer, Dordrecht, 1998.

[31] J. Weizenbaum. ELIZA: a Computer Program for the Study of Natural Language Communication between Man and Machine. *Communications of the ACM 9*, 36-45, 1966.