Computational Linguistics for Theory and Practice

Piek Vossen
Sail-labs, Antwerp
Piek.Vossen@sail-labs.be

1. Introduction

In this paper I will discuss the role of computational linguistics to develop language technology for user-applications. Early attempts to implement fundamental linguistic models in computers did not result in software that was useful for people. Various reasons can be given, among which that they tried to solve too many problems, that there was a structural lack of resources and too strong a focus on representational issues, without a clear specification of relevant application requirements. Nevertheless, we see that language technology and applications still have been developed but with minimal linguistic machinery. Statistical approaches seem to dominate the market, although, their capacity to process information is limited by definition. The expectation is that, eventually, it will be necessary to use language technology and deliver more precision and quality in information processing. However, this technology has to overcome some fundamental problems. I will therefore argue for more fundamental research within the area of Computational Linguistics, but not, as was done before, to implement some linguistic model or theory. Instead, the research should set priorities and focus on solving the most urgent problems and bottlenecks that are holding down current developments for Natural Language Applications

2. Computational models of linguistic theories

In the mid-eighties, Computational Linguistics at the University of Amsterdam mainly involved implementing a linguistic model, such as a grammar or a set of rules, in a computer program for the purpose of that linguistic model. A successful implementation was seen as evidence in favor of that linguistic theory. A grammar implementation could then be tested on sentences to proof descriptive adequacy (Chomsky 1965): the ability to describe all possible sentences.

By building a computational model, the linguist will realize implicit assumptions of the model that slipped the attention but must be made explicit for a computer. The linguist would learn from building the model. At the University of Amsterdam, Dik was leading a group of linguists and computer scientists that aimed at building a so-called CMNLU: a Computer Model of a Natural Language User (Dik 1989a). By discussing this model and partially implementing components, Dik's theory of Functional Grammar model would be improved (Dik 1989b).

On the other hand, there were discussions between linguists and computer scientists on the implications of these implementations. Computer scientists implemented more general parser formalisms or parser generators but the linguists complained about the limitations of these systems. They were forced to work in a way that they did not wish or like. In the end, Dik refused to use a Tomita-based parser-generator, and eventually implemented his theory of Functional Grammar in Prolog all by himself (Dik 1992). This implementation was heavily critisized by computer scientists as not being very efficient and well-designed, and the implementation also slightly differed from Dik's theory of Functional Grammar. Backtracking in Prolog forced him to some choices, such as parsing by generating.

Such experiments and implementations never resulted in complete working systems. They all lack descriptive adequacy because of the limited size of their lexicons and the incompleteness of the grammars. Briscoe and Boguraev (1989) pointed out that the existing systems could never be useful and could never be tested because the average size of their lexicons was 25 entries. They dubbed this the "lexical bottleneck" and initiated an EC-project Acquilex that was supposed to develop methodologies to build generic large-scale and rich lexical resources. These lexicons were extracted from Machine-Readable-Dictionaries or corpora. Again, the early lexicon projects had a strong focus on the formalism for representing data, relying on unification and Typed Feature Structures (Carpenter 1990) and HPSG (Pollard and Sag 1987). However, in this case the formalisms were not enforced by a linguistic theory but based on computational linguistic methods. For a discussion on lexical representations see Briscoe et al. (1993)

Acquilex was a pilot-research project that still resulted in small lexicons: 1000 up to 2000 word senses per language. Despite their richness, these lexicons did not solve the bottleneck. More importantly, these projects did not start from a set of specific application requirements and their formalisms were not integrated in for example a Machine Translation system. The implementation of complex Typed Feature Structure representations in Prolog or Lisp would also make it impossible to exploit these resources in the large-scale multi-user environments that are currently required. To retrieve a lexical specification for a word sense, its definition of features and values first had to be unified with the assigned types (including non-default and default inheritance). This could take minutes in a larger lexical knowledge base.

Obviously, these systems are experimental platforms to discuss, model and describe various lexical and linguistic phenomena. In a way, I sometimes see Acquilex as a problem-discovery project. Many of these phenomena and problems remain unsolved, even today after 10 years. This is not very encouraging for developing realistic applications. It does not make sense to hugely invest in the development of general-purpose lexical knowledge bases with complex representations if the purpose and use is not proven.

To summarize, we can say that the fundamental approaches in the eighties failed to produce working systems for various reasons:

- too much focus on representational issues and formalisms;
- lack of descriptive adequacy, both because of the size of the lexicons and the scope of the grammars;
- inadequacy of the theories to provide a descriptive framework for many phenomena;
- being experimental implementations (using Prolog or Lisp) that cannot easily be integrated in realistic environments;
- lacking a requirement-specification of what is really needed for language technology applications;

3. Applications with or without Computational Linguistics

Despite the failure of the fundamental Computational Linguistic approach to produce real applications, the current situation is still more hopeful than one would expect. There are nowadays various successful language technology products with sufficient quality, to name a few: spelling checkers, summarizers, automatic indexers and search engines. However, these applications typically use a minimal amount of linguistic rules and resources, and especially rely on statistics. Instead of a high-quality and deep analysis of text, they rely on shallow parsing, small stop-lists, and occasionally a simple lexicon.

The realization that partial solutions can be useful, that imperfect systems can solve real problems and that they can be the basis for commercial systems, has been very important for the commercial development of the field. In this respect, the notions of precision and recall that were introduced in Information Retrieval have had a tremendous effect. This made it possible to measure effectiveness of systems, and more important, to measure progress. The language technology systems in the MUC and TREC competitions combined minimal development costs and maximal gain. They hardly lead to new insights in Linguistics or in Computational Linguistics but they worked and could be implemented on a realistic scale. More striking, they beat all linguistic approaches (Voorhees 1999).

Suddenly, the linguistic foundation or theory has become totally unimportant, not even the computational formalism. As long as systems work, work fast, for huge quantities of data, can be easily adapted, and are robust. Within information retrieval, a purely statistical approach is even a trend. The main reasons for this are that statistical techniques:

- do not require development of language resources;
- do not require customization or verticalization to specific domains;
- use mathematical methods that can easily be implemented in computer programs;
- are robust, giving output even when information is not fully processed;
- are relatively language-independent (at least for analytic languages such as English);

However, the idea that these statistical techniques are also better than NLP-based techniques is a misconception. By definition, pure statistical techniques cannot be better than NLP-based techniques, for one reason because the latter may incorporate statistical techniques as well and improve the result. Nevertheless, there are other reasons why, on the long run, statistical techniques are only of limited value.

First of all, statistical techniques match (sub)strings across documents without any structural analysis and therefore cannot deal with:

- variation in expression of the same content by different strings within and across languages;
- make use of the compositional meaning of the linguistic structure;

There is only one exceptional type of language application, general Internet search-engines, for which the statistical techniques do seem to give satisfactory results in a mono-lingual setting, exactly because these two deficits are often not or less relevant.

A general search-engine on the Internet, first of all, does not deliver a specific answer to a question but only a list of documents on a certain topic, usually specified by a key-word query. If the structural relation between the query items is not important, as in the boolean query "medicine" & "poison", then it does not matter that the statistical techniques cannot recognize the structure in which the key words occur in text. This is different for compositional queries such as "poisonous medicine". Statistical techniques will certainly return documents on "medicines that cure the effects of poisonous substances" (which are not requested) and maybe also return documents on "medicines that are poisonous" (which are requested). To find this out, the user however has to read all the returned documents, one by one. The only way to correctly address the above query is not only to analyze the compositional structure of the query, but also to analyze the compositional relation of "poisonous" and "medicine" in every document that is retrieved. Obviously, a Natural Language Processing engine may do this on the output of a statistical run to limit the documents to be processed, but, definitely, a linguistic analysis is required.

The second deficit of pure statistical technology is that it cannot deal with symbolic variation to capture the same content. If the document speaks of "medication" and "poison", or "laxative" and "toxic", then it will not be returned. Again, internet-searches are a special case here. Because of the enormous size of the data on the internet, and because of the redundancy of the information that can be found, the chances are relatively high that a statistical internet search will return another document that contains the exact words found in the query. The same information is stored maybe hundreds of times, capturing most common variation. Users that do not require precise and complete information will never realize that a lot of information is missed instead of found.

Clearly, size and redundancy cannot be expected from intranets and most e-commerce settings. The answer (e.g. a company product or its CEO) is probably stored only once and the query should be mapped to this answer regardless of the phrasing. The only way in which linguistic variation can then be captured is by using linguistic resources (i.e. WordNet with synonyms, hyponyms, hyperonyms and other related terms, Fellbaum 1998) and by linguistic analysis.

Furthermore, precision is much more important in domain-specific intranets and e-commerce applications. In fact, customers do not want a list of documents that may contain the answer or not, they often want a single answer to their question, a solution to their problem or a specific service (Guarino et al. 1998). To deliver precision and to anticipate the specific user-needs puts some specific requirements on the analysis of the data and the user-queries. Within small-scale, precision-sensitive search and dialogue systems, the above linguistic analysis makes the difference.

Finally, the statistical techniques will be less effective for languages that are less analytic than English. In English, word variation and compounding is minimal so that different uses of words can easily be related. This makes English a good 'indexing' language. Synthetic languages however show much more morphologic variation so that it is more difficult to relate inflected, compounded and derived forms to the same stem. For example, languages such as Czech and Russian have over 40,000 verbal root forms in an ordinary dictionary, whereas English has about 9,000 verbs (including phrasal verbs). Finally, there is also variation between the type of analytic constructions themselves. Whereas English will have many Noun-Noun compounds (separated by spaces and hyphens), Romanic languages will have Noun-Adjective or Noun-preposition-Noun constructions or other derivations. These differences cannot be neglected when mapping content across languages.

4. What can theories do for Language technology?

Assuming that language technology can therefore be used to improve the current statistical approaches, what could be the role of Linguistic or Computational Linguistic theories for the technology? Where to go? Obviously, we should not return to the fundamental approaches of the eighties to implement human communication in all its complexity. The difference is that, now, we do have applications, there is a language technology market and we have enormous test-bed: the Internet. The advantage of real systems and applications is that it makes clear where the specific problems are and that we can formulate the system requirements as a realistic goal. Language technology needs fundamental research and better computational linguistic theories or models to solve these crucial problems. To illustrate this, I want to focus on one component of Computational Linguistics where a lot of research is still needed: the use of lexical semantic resources or ontologies like WordNet (Fellbaum 1998).

Lexical semantic resources and ontologies such as WordNet, can directly be used for various applications. In information retrieval they are used for query-expansion or indexation at a concept level rather than a word level (see discussion above). For other applications, such as summarization, entity-recognition, information extraction, they can be used for making limited inferences on the semantic constraints: for resolving anaphora resolution, selectional restrictions, entity recognition, etc. The main problem for using these resources in these ways is however lexical ambiguity. All semantic resources are highly polysemeous and there is not sufficient information to select the senses in context. For information retrieval this poses an enormous problem. Instead of expansion to a set of closely related terms, a query will be expanded to all related terms for all meanings. A more general and frequent word may have 5 up to 50 different meanings, and each meaning is related to different synonyms (1 up to 10), hyponyms, hyperonyms or other related concepts, each of which may have different meanings as well. When we apply this cross-linguistically, there is another multiplication because each expanded word can have multiple translations, again with different meanings in the target language. Blind query expansion using semantic networks therefore generates so much noise that it becomes a-productive: the improved recall will get lost in the increased noise.

The noun “line” from WordNet is illustrative for the complexity of the problem. It has 26 different senses, relates to 50 other synonyms and when mapped to Dutch via a bilingual English-Dutch dictionary these 51 words expand to over 1200 concepts in Dutch (each synonym having multiple meanings, with multiple translations with multiple meanings).

Synsets containing “line” in WordNet1.5:

SYNSET: business-# 2; line-# 1; line of work-# 1; occupation-# 3
GLOSS: the principal activity in your life; he's not in my line of business
HYPER: activity-# 1
SYNSET: line-# 2
GLOSS: acting in conformity; in line with or he got out of line or toe the line
HYPER: abidance-# 3
SYNSET: business line-# 1; line-# 3; line of business-# 2; line of merchandise-# 1; line of products-# 1; product line-# 1
GLOSS: a particular kind of product; a nice line of shoes;
HYPER: commodity-# 1: sideline-# 2
SYNSET: cable-# 3; electrical cable-# 1; line-# 4; transmission line-# 1
GLOSS: an electrical conductor connecting telephones or television or power stations
HYPER: conductor-# 1:
SYNSET: line-# 5
GLOSS: something long and thin and flexible
HYPER: artefact-# 1:
SYNSET: line-# 6; rail line-# 1; railway line-# 1
GLOSS: railroad track and roadbed
HYPER: railroad track-# 1:
SYNSET: line-# 7
GLOSS: a commercial organization serving as a common carrier
HYPER: carrier-# 4
SYNSET: line-# 8; pipeline-# 1
GLOSS: a long pipe used to transport liquids or gases; a pipeline runs from the wells to the seaport
HYPER: pipage-# 1
SYNSET: assembly line-# 1; line-# 9; production line-# 1
GLOSS: a factory system in which an article is conveyed through sites at which successive operations are performed on it
HYPER: system-# 1
SYNSET: line-# 10; phone line-# 1; telephone line-# 1
GLOSS: a telephone connection
HYPER: connecter-# 1
SYNSET: contrast-# 3; demarcation-# 1; dividing line-# 1; line-# 11
GLOSS: a conceptual separation or demarcation: there is a narrow line between sanity and insanity
HYPER: differentiation-# 2
SYNSET: argumentation-# 1; line-# 12; line of reasoning-# 1; logical argument-# 1
GLOSS: methodical reasoning; I can't follow your line of reasoning
HYPER: abstract thought-# 1
SYNSET: line-# 13; note-# 3; short letter-# 1
GLOSS: drop me a line when you get there
HYPER: personal letter-# 1
SYNSET: line-# 14
GLOSS: a mark that is long relative to its width; He drew a line on the chart or The substance produced characteristic lines on the spectroscope
HYPER: mark-# 7
SYNSET: line-# 15
GLOSS: a linear string of words expressing some idea; the letter consisted of three short lines
HYPER: linguistic string-# 1
SYNSET: air-# 7; line-# 16; melodic line-# 1; melodic phrase-# 1; melody-# 2; strain-# 6; tune-# 1
GLOSS: a succession of notes forming a distinctived sequence; she was humming an air from Beethoven
HYPER: music-# 4
SYNSET: ancestry-# 2; blood-# 5; blood line-# 1; bloodline-# 2; descent-# 4; line-# 17; line of descent-# 1; lineage-# 2; origin-# 4; parentage-# 1; pedigree-# 3; stock-# 8
GLOSS: the descendants of one individual; he comes from good lineage
HYPER: family tree-# 1
SYNSET: line-# 18
GLOSS: a formation of people or things one after another; the line stretched clear around the corner
HYPER: formation-# 4
SYNSET: line-# 19
GLOSS: a formation of people or things beside one another; the line of soldiers advanced with their bayonets fixed; they were arrayed in line of battle
HYPER: formation-# 4
SYNSET: course-# 6; line-# 20
GLOSS: a connected series of events or actions or developments; the government took a firm course or historians can only point out those lines for which evidence is available
HYPER: series-# 4
SYNSET: line-# 21
GLOSS: a spatial location defined by a real or imaginary unidimensional extent
HYPER: location-# 1
SYNSET: line-# 22
GLOSS: in games or sports; a mark indicating positions or bounds of the playing area
HYPER: mark-# 7
SYNSET: line-# 23
GLOSS: a single frequency (or very narrow band) of radiation in a spectrum
HYPER: electromagnetic radiation-# 1

SYNSET:	bank line-# 1; credit line-# 2; line-# 24; line of credit-# 1; personal credit line-# 1; personal line of credit-# 1
GLOSS:	the maximum credit that a customer is allowed
HYPER:	credit-# 7
SYNSET:	agate line-# 1; line-# 25
GLOSS:	space for one line of print (one column wide and 1/14 inch deep) used to measure advertising
HYPER:	area unit-# 1
SYNSET:	line-# 26
GLOSS:	a length (straight or curved) without breadth or thickness; the trace of a moving point
HYPER:	form-# 1

What is lacking to make a choice between these meaning is not only additional information about the usage in context, but also what the relations are between these senses, how relevant it is to differentiate between all of them? Does the relevance of the sense-differentiation also vary from context to context? Clearly senses can be clustered and related, e.g. sense-1, sense-3, sense-7, sense-9 are somehow related to industrial production, all senses for long thin objects (4,6,8,10,11,15,18,19,21,22,23,25,26) are related to the generic sense-5. The latter seem to form a productive group. They all differ in the material they consist of and the reason or purpose they are constructed for. New senses can therefore be derived easily, e.g. “a line of cocaine”.

How can we then solve the lexical ambiguity problem? Many researchers have tried to come up with an answer. The problem is however not easy to solve. In fact, there is even strong disagreement between human annotators when tagging corpora with word senses. Furthermore, it turns out that lexical resources differ extremely in the way they differentiate senses. Not only across different resources but also within the same resource for different but related words. Various people have suggested that a word sense is an artificial notion (Cruse 1986, Lakoff 1986). Senses form a continuum and meaning is not rigid. However, there is also some hope. Others claim that polysemy is regular and can to some extend be predicted (Pustejovsky 1995, Copestake and Briscoe1991).

In addition to polysemy, there is also another problem called vertical ambiguity by Resnik (1995, 1998). Once we know the meaning of a word, what is the level of specificity to which we should expand? Should we match “Mercedes” to “car”, “motor vehicle”, ”vehicle”, or “object”? It makes an enormous difference for retrieval if we expand to higher levels or to middle regions. If we know the best level for abstracting, then we can also make the most efficient and intuitive index for representing concepts. What is then the most appropriate level of abstraction to match concepts? Intuitively, we would say that “car” is the best level for representing concepts. This corresponds to the notion of Basic Level as defined by Rosch (1977). However, there are no hard criteria to define what the Basic Level is and there is evidence that the level can even vary depending on the interest and specialization of the speakers.

The problem is that we do not have solid principles to define what a concept is. What makes a concept a concept? Why not have two concepts when we have one, why not create one when we have two? And if we have defined a concept, how do we relate the lexicalizations of a language to these concepts? Are there any concepts without words, how should different words be related to the same concept, and how do they differ? Such fundamental questions come into mind when people try to merge different ontologies and resources that exhibit different but legitimate choices (Hovy 1998, Guarino 1998, Vossen and Bloksma 1998). It is striking for the status of semantic theories, that a resource such as Wordnet is built around the notion of a synset: a set of synonymous words, whereas nobody can define synonymy or give a water-proof method for detecting them. Most scholars don't even believe in their existence. How can we then expect that lexicographers and linguists make consistent decisions when defining words in a lexicon? How can we expect that language technology can make efficient use of these definitions?

To illustrate the different ontological choices that can be made in building a general and generic ontology, just compare the most frequent classifications in Cyc (Lenart and Guah 1990) and in WordNet1.5. In Cyc, we find many artificial classes and categories that are used to capture certain inferences but are not natural lexicalizations in language. Classes, such as:

AnimalBodyPart, *ContainerProduct*, *SolidTangibleThing*, S*omethingExisting,*

are not intended as lexical entries in a lexicon but are purely designed for structuring knowledge. These categories are not used in WordNet1.5 and therefore the definition of concepts across these ontologies can never be the same (although they can still be compatible). In principle, we can create categories for any property that we can imagine (Gruber 1992) and apply these to our concepts or lexicon. This means that there is no a priori way of establishing the categories for differentiating senses (horizontal ambiguity), nor the levels for differentiating hierarchies (vertical ambiguity). Not surprisingly, ontologies can and do make different choices in (relevant) categories and/or apply the same categories in different order, e.g. first from a functional perspective (as medicines) and then from a constitutional perspective (powders, liquids) or the other way around.

This is illustrated in Figure-1 below. Different choices are made in the Longman Dictionary of Contemporary English (Procter 1978, LDOCE), the Van Dale monolingual Dutch dictionary (van Sterkenburg and Pijnenburg 1984) and WordNet1.5 for a relative easy concept such as *dog*. We see here that in LDOCE *pet, mammal* and *dog* are all directly defined as subtypes of *animal* (probably due to the use of the controlled vocabulary). The relation between *dog* on the one hand and *mammal* and *pet* on the other hand is not indicated. In Van Dale and WordNet, we see that each expresses one of these more specific relations but none of them expresses both (because only one perspective or conceptualization is given in a definition and in WordNet1.5).

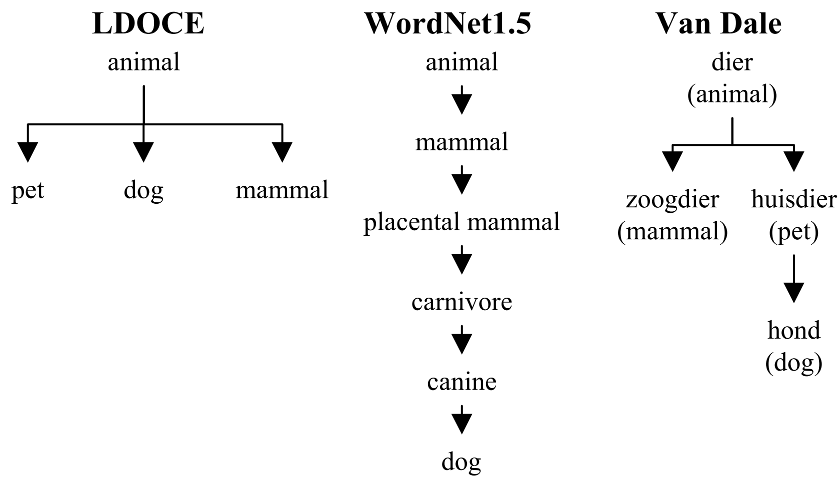

Figure 1: Different classification choices for "dog" in LDOCE, Van Dale and WordNet

If we compare wordnets across different languages, we also see many lexically motivated differences. For example, we would expect that the Dutch nouns *doos* (box), *tas* (bag), *asbak* (ashtray), *lepel* (spoon) are related to the same hyperonym **container** as the WordNet1.5 equivalents. However, in Dutch there is **no** direct equivalent for *container*. [1] As a result of this we see that these concepts are directly linked below the equivalent of *object* (voorwerp) in the Dutch wordnet, see Figure 1.

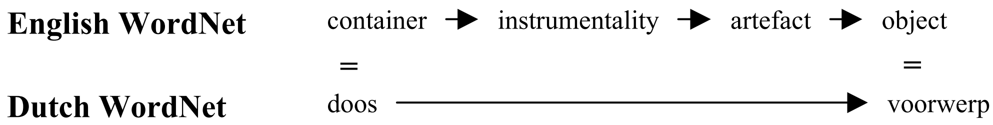

Figure 2: Different lexicalized classes in the Dutch and English wordnet

Since the lexicalizations across languages are all different, wordnets cannot have the same hierarchical structure by definition and thus cannot lead to the same expansion in retrieval.

The purpose of an ontology or a lexical semantic network is definitely important for the principles by which concepts are distinguished and defined (so-called identity criteria, Guarino 1998, Sowa 1999). If the purpose is to predict substitution of words in text (information retrieval, language generation) we need something like a *wordnet* that precisely represents the relations between the lexicalized classes of a language and thus predicts how the same content can be paraphrased differently in a language. The lexicalizations in a language (and eventually also other more complex phrases and expressions) have to be the starting point for the ontology. If the purpose of the semantic network is to predict properties we may need a very different design (more like the Cyc ontology). Many words in a language may not be relevant for storing the relevant inferences and many concepts may be needed that are not lexicalized at all.

As long as there is no consensus on these issues and as long as there are no methods and principles how to define concepts and relate words to these concepts, we cannot motivate a huge investment in high-quality semantic resources. Fundamental research should develop the instruments and criteria to build consistent and high-quality

[1] The word *container* in Dutch does exist but is only used for big containers on ships or for big garbage cans.

resources. But these theories should have a solid empirical grounding in data. They should try to solve real problems and they should not loose themselves in endless discussions on formalisms or ontological classifications.

5. Conclusion

The early Computational Linguistics Systems had a strong focus on formalisms and often tried to implement specific Linguistic models. Most of these attempts failed to produce working systems for various reasons:

- too much focus on representational issues and formalisms;
- lack of descriptive adequacy, both because of the size of the lexicons and the scope of the grammars;
- inadequacy of the theories to provide a descriptive framework for many phenomena;
- being experimental implementations (e.g. using Prolog or Lisp) that cannot easily be integrated in realistic environments;
- lacking a requirement-specification of what is really needed for language technology applications;

A first conclusion is that Computational Linguistics nowadays is freed from linguistic formalisms. There is now a whole variety of techniques available to process language, such as finite-state machines, feature-structure unification, problem solving, ontology-based reasoning, bayesian networks, supervised and unsupervised machine learning, vector-space comparison. Computational Linguistics can make the most adequate choice depending on required task, the available resources, the level of precision, the scale of the implementation, not necessarily adopting a specific linguistic theory or approach. A typical approach is in this respect the data-oriented parsing (DOP) method developed by Bod and Scha (1997). DOP learns to parse sentences by training on parse trees, without explicit formulation of grammar rules. The training can be based on purely descriptive encoding with any choice of labels and structures. No assumptions have to be made on the way these patterns are derived. In such an approach, linguistic rules are not even used to arrive at a linguistic analysis and the grammar remains fully implicit.

Secondly, real language-applications do exist, even though they are less ambitious and of less quality as initially expected. However, it turned out that technology does not have to be perfect to develop useful applications. Statistical approaches with shallow processing have been most successful so far diminishing the role of language technology. However, the expectation is that Language technology will become more and more important within these applications when the requirements rise and more precision is required.

A third conclusion is that we still need theory, both linguistic and computational theories. However, instead of supporting these theories by modeling them, the theories should now support Computational Linguistics or more precisely Language Engineering. We need specific research, development and testing of models to tackle some fundamental and crucial problems. Obviously, this does not imply that linguists cannot still use computers to support their work. In fact, every linguist should be a Computational Linguist.

References

Bod, R. and R. Scha, 1997. "Data Oriented Language Processing". In S. Young and G. Bloothooft (eds.) *Corpus-Based Methods in Language and Speech Processing*, Kluwer Academic Publishers, Boston. 137-173.

Briscoe, E J., A. Copestake and V. de Paiva (Editors). 1993. *Default Inheritance in Unification Based Approaches to the Lexicon*. Cambridge UK: Cambridge University Press.

Briscoe, E.J. and B. Boguraev, (Editors). 1989. *Computational Lexicography for Natural Language Processing*. London/New York: Longman.

Carpenter, R.1990. "Typed feature structures: inheritance, (in)equality and extensionality". *Proceedings of the Workshop on Inheritance in Natural Language Processing*. Tilburg: 9-18.

Chomsky, N., 1965, *Aspects of the theory of syntax*, Cambridge, Mass., MIT Press.

Copestake A. and Briscoe, T (1991). *Lexical operations in a unification-based framework*. In: Pustejovsky J. and Bergler S. (eds.), Lexical Semantics and Knowledge Representation Association for Computational Linguistics.

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

Dik, S.C. 1989a, FG*C*M*NLU: Functional Grammar Computational Model of the Natural Language User. In: Conolly and Dik (eds.), 1-28.

Dik, S.C. 1989b, The Theory of Functional Grammar, Part I: The Structure of the Clause, Functional Grammar Series, 9, Foris, Dordrecht.

Dik, S.C. 1992, Nederlandse Functionele Grammatica in Prolog, Amsterdam Linguistic Software, Amsterdam.

Fellbaum, C. (ed.) 1998. *WordNet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

Gruber, T.R. (1992) *Ontolingua: a Mechanism to Support Portable Ontologies*. Report KSL 91-66. Stanford University. 1992

Guarino, N. 1998. "Some Ontological Principles for Designing Upper Level Lexical Resources".*Proceedings of First International Conference on Language Resources and Evaluation* ed by In: A. Rubio, N. Gallardo, R. Catro and A. Tejada, Granada, 28-30 May 1998: 527-534.

Guarino, N., C. Masolo, G. Vetere. 1998. *OntoSeek: Using Large Linguistic Ontologies for Accessing On-Line Yelow Pages and Product Catalogs*. LADSEB-CNR, Technical Report 02/98.

Hovy, E. 1998. "Combining and Standardizing Large-Scale, Practical Ontologies for Machine Translation and Other Uses".*Proceedings of First International Conference on Language Resources and Evaluation,* ed. by In: A. Rubio, N. Gallardo, R. Catro and A. Tejada, Granada, Spain, 535-542.

Lakoff, G. 1986. "Classifiers as a reflection of mind", In: C. Craig (ed.) *Noun classes and categorization*. Bejamin, Amsterdam, 31-51.

Lenat, D.B. and R.V. Guha. 1990. *Building Large Knowledge-based Systems. Representation and Inference in the CYC Project.* Addison Wesley.

Pollard C. and Sag I. 1987. *Information-Based Syntax and Semantics, Vol 1: Fundamentals* CSLI Lecture Notes no. 13. Stanford.

Procter, P. (Editor) 1978. *Longman Dictionary of Contemporary English*. Longman, Harlow and London.

Pustejovsky, J. 1995. *The Generative Lexicon*. MIT Press, Cambridge, MA.

Resnik, P. 1995. *Using information content to evaluate semantic similarity in a taxonomy*. *Proceedings of IJCAI.*

Resnik, P. 1998. “Wordnet and Class-Based Probabilities”, *WordNet, An electronic lecixal database*, ed by C. Fellbaum, Cambridge MIT, 239-264

Rosch, E. 1977. "Human Categorisation". Ed. N. Warren Studies in *Cross-Cultural Psychology*, Vol. I, 1-49. Academic Press. London.

Sterkenburg J. van, and W.J.J. Pijnenburg (eds) 1984 *Groot woordenboek van hedendaags Nederlands*, Van Dale Lexicografie, Utrecht.

Sowa, J. 1999. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, PWS Publishing Co., Boston.

Voorhees, E. 1999. “Natural Language Processing and Information Retrieval”. *Information Extraction: Towards Scalable, Adaptable Systems* ed. by M. T. Pazienza. Germany: Springer, 32-48.

Vossen P. and L. Bloksma. 1998. “Categories and classifications in EuroWordNet”. *Proceedings of First International Conference on Language Resources and Evaluation* ed by In: A. Rubio, N. Gallardo, R. Catro and A. Tejada, Granada, 28-30 May 1998. 399-408

Vossen, P. (Editor) 1998. *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*, Kluwer Academic Publishers, Dordrecht