Using a Parameterizable and Domain-Adaptive Information Extraction System for Annotating Large-Scale Corpora?

Thierry Declerck, Günter Neumann

DFKI GmbH (German Research Center for Artificial Intelligence)
Language Technology Lab
Stulsatzenhausweg 3, 66123 Saarbrücken, Germany
{declerck,neumann}@dfki.de

Abstract

In this paper we describe a parameterizable and domain-adaptive Information Extraction (IE) system (for German texts) and present some ideas on how this kind of system could effectively support Corpus Linguistics (CL) tasks. We also tentatively address the complementary question and look in which sense corpus linguistics can be beneficial to IE, specially in the case of automatic learning of templates of interest for IE tasks, a topic which is crucial for the further development of highly flexible IE systems. We describe briefly some steps done for the adaptation of the IE system to a new domain in order to illustrate the points where in our opinion IE and CL should go for a closer cooperation.

1. SMES: A Parameterizable Domain-Adaptive IE System for German

Information Extraction (IE) is generally defined as the task of identifying, collecting and normalizing information from Natural Language (NL) text. The information of interest is typically pre-specified in form of uninstantiated frame-like structures also called *templates*. The templates are domain and task specific. The major task of an IE-system is then the identification of the relevant parts of the text which are used to fill a template's slot. In the context of the PARADIME (Parameterizable Domain-Adaptive Information and Message Extraction) project[1] we have been more specially concerned with the investigation of methodologies for designing an IE system easily adaptable to new domains of application. For this we went for a strict separation of the (shallow) linguistic processing and the domain-modelling modules, thus looking for the maximal degree of reusability of common linguistic resources shared by all domains of application.

The assumption underlying our modular approach to IE is that it speeds up the adaptation of the IE-system to new applications, since in the ideal case the configuration task can be reduced to the (declarative) definition of the domain model, the creation of a domain lexicon and the definition of some specific interpretation rules (constraints) for the output of the generic shallow NL processing tools involved. An overview of the architecture of the system is given in figure 3.

1.1. The NLP Components

The linguistic analysis is performed by an integrated set of text processing tools supporting partial and shallow parsing. The NLP components consist in a tokenizer (describing ca. 60 generic token classes), a morphological analyzer (incl. compound analysis), a POS tagger, a Named Entities recognizer, a fragment recognizer (NPs, PPs and Verb-groups), and a dependency parser on the top of which a heuristic for detecting grammatical functions (GFs) has been implemented[2]. Interleaving with some of the components mentioned an algorithm for reference resolution has also been implemented. An example of the output of the NLP sub-system is given in figure 1. We also call such a structure an Underspecified Functional Description (UFD) of the input sentence.

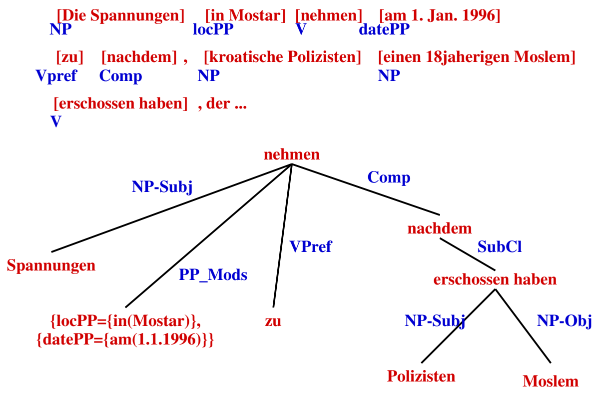

Figure 1: The output of the fragment recognizer (above) and of the (flat) dependency parser, represented as a tree

The results of each component as well as of the whole procedure are also available in form of XML annotated data. The linguistic processing and the XML-marking of text data proved to be quite efficient: ∼32sec are needed for ca. 200,000 items (∼6160 words/sec) on a standard computer (PentiumIII, 500MHz, 128Ram)[3].

[1] See for more information on the PARADIME project: http://www.dfki.de/pas/f2w.cgi?ltp/paradime-e.

[2] Up to the parser and the detection of GFs, all the components have been re-implemented in C++ and embedded in a user-friendly environment (Piskorski and Neumann, 2000). The remaining components are currently being ported from the existing Lisp system (Neumann et al., 1997) in the new platform.

[3] Up to the reference resolution task, all the components of the NLP module have been evaluated and show satisfying results, the precision being in any case higher than 90% (Piskorski and Neumann, 2000).

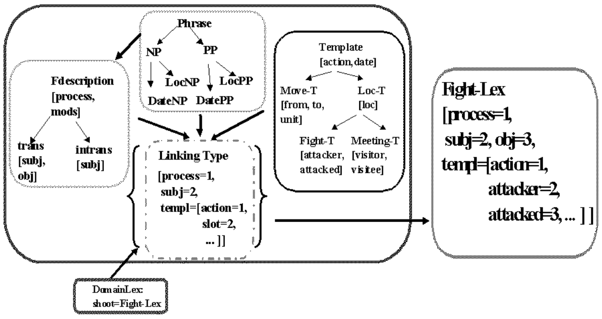

Figure 2: The domain model: in the left frame one can see the combination of conceptual hierarchies (domain and linguistic knowledge) used for defining a set of linking types. Mapping an entry of the domain lexicon into this set generates the uninstantiated template to be filled

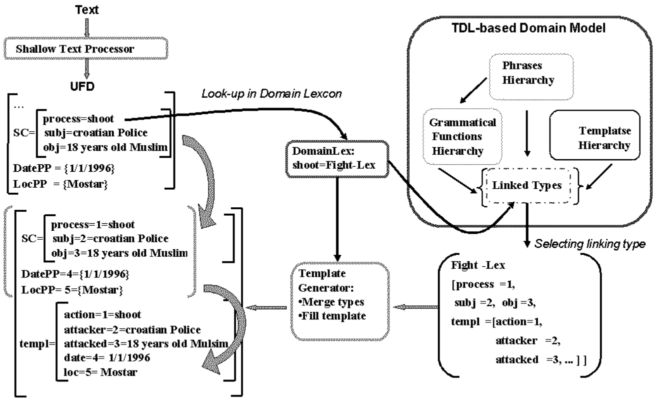

Figure 3: A sketch of the whole processing chain leading to template filling

1.2. The Domain Modelling Component

The domain-specific knowledge is modeled with the help of the Type Description Language (TDL) formalism[4] supporting all kind of operations on (typed) feature structures, including multiple inheritance, and so we can declaratively model the domain under consideration in the form of a conceptual hierarchy (ontology) of (under-specified) feature structures representing the IE-templates to be filled, which in this sense can be seen as the formulation of 'semantic' constraints on the output of the shallow NLP tools (see the right box in the left frame of figure 3). Thus we make use of a well-defined high-level (linguistic) formalism for modelling the domain. The domain model can be embedded in a general purpose ontology or conceptual structure (like WordNet). So an "attacker" can further been described as the subtype (or supertype) of a "soldier" which further is a subtype of "human" etc.

1.3. The Language Resources of SMES

The NLP components of SMES have at their disposal a generic lexicon (with more than 150,000 morpho-syntactically marked stems), including a database containing valence information for 12,000 verbs, and large specialized lexicons (gazetteers). The template-filling task is supported by domain lexicons, in which lexical anchors for guiding the detection of the relevant information in text are stored. The domain lexicon associates domain-relevant entries with specific templates of some domain, as can be seen in figure 2.

1.4. The Integration of the two main Modules of the IE System

An abstract mapping between the domain model and an hierarchy abstracting over the data provided by the NLP tools ensures in a straightforward manner the linking of the domain-independent and the domain-specific knowledge. Those mapping (or linking) rules also guide the generation of the templates to be filled by the application (see 3). Thus the use of such a highly declarative formalism allows the easy definition of an interface between the modules of the IE-system, and also to some external general purpose information repository. This has also the interesting side-effect that those semantic (or conceptual) constraints can be used for the XML annotation of the running text.

2. Relevance of the SMES System for Corpus Linguistics?

When we started to think about this topic, looking in a first step only at the modular strategy implemented in SMES, we discovered that the separation of generic NL analysis and domain-modelling tools within an IE-system can have the side-effect of bringing IE and Corpus Linguistics (CL) closer together, since this kind of architecture considerably weakens the assumption generally done, that one of the gap between the two disciplines is due to the fact that IE is concerned only with domain-specific texts whereas CL deal with heterogeneous texts covering a large number of domains. As a consequence of the modular architecture of the IE-system, following can be assumed:

- The efficiency (and robustness) of the generic NL processing makes such systems attractive for CL, since a (partial) automatic (domain-independent) linguistic annotation (at the levels and the quality offered by the NLP tools) of a huge amount of data can be provided in a reasonable time. The available XML ouput of the core NL analysis can be used as an interface to this kind of annotation task.
- The linguistic annotations can be enriched with information taken from the templates instantiated by the IE system if some domain relevant structures is detected in the text. So the running text can not only be (partially) annotated with the semantic information associated with the available syntactic information, but also – after merging entries of the domain lexicon with the activated templates and other conceptual structures –

[4] See (Krieger and Schaefer, 1994)

with domain and/or world knowledge. The declarative mapping in the form of linking types between the results of the NL processing and the uninstantiated templates on the one hand and the well-defined embedding of the templates in hierarchical conceptual structures on the other hand ensure a consistent merging of the distinct kind of information. Here we still have to look for an adequate XML representation to be derived from the hierarchical organization of (typed) feature structures.

- Since it is basically possible to combine distinct domain models in such an adaptive IE system (positive experiments have been done with respect to this), it will also be possible to reduce the degree of 'partiality' of the semantic/domain-specific annotation of the syntactically processed text. And for the parts of the text not being annotated at all with semantic/domain information, one could just propose some annotation derived from general purpose ontologies. In any case, the co-presence of annotations from different domain models can probably help CL in detecting some new kind of regularities in the texts, which might have not been detected on the base of a sole shallow processing.

3. Adapting the IE-System to a New Domain

For adapting the IE-system to a new domain experience showed that at least three processing steps are necessary:

1. Data collection, corpus and domain analysis, identification of typical terms, relations and events, and description of the templates to be filled for the application. This task is a constant one for every adaptation to new domains (can be tackled by the user or by the developer, or a combination of both). The efficiency and accuracy of this task depends on the expertise of the persons and on the quality of the tools involved.

2. Integration of the templates into a conceptual hierarchy (ontology) in order to describe the domain model and (partially) merge this conceptual structure into existing ontologies. This is the base of the definition the linking types for template filling. The linking types will have to define specific *interpretations* of the data delivered by the generic NLP tools and also describe semantic and pragmatic *constraints* on this output in order to ensure the accuracy of the template selection and filling.

3. Selective adaptation of the modules of the NLP component of the IE, if necessary, and description of the domain lexicon (containing at least the typical event words). Ideally no module of the NLP component should be affected, and so the improvement work made necessary by a new application will tend to make those tools still more generic.

4. A Case Study: Adapting SMES to the Soccer Domain

The concrete adaptation of the IE-system to the soccer domain has been done following the three steps mentioned in section 3., according to the sub-tasks defined in (SAIC, 1998).

In step 1) we have been collecting 323 texts about the Soccer World Championship 1998 from the Frankfurter Rundschau (German newspaper available on-line) out of which 62 game reports have been selected for a detailed corpus analysis, which allowed to detect domain specific terms, relations and events:

- Terms as descriptors for the recognition of Named Entities (NE) – TEAM: *Titelverteidiger* Brasilien; PLAYER: *Superstar* Ronaldo, von *Bewacher* Calderwood noch von *Abwehrchef* Hendry; REFEREE: vom spanischen *Schiedsrichter* Garcia Aranda; TRAINER: Schottlands *Trainer Brown*; Location: *im Stade* de France; ATTENDANCE: Vor 80000 *Zuschauer*;

- Terms for NE recognition – TIME: *in der 73. Minute*, von Roberto Carlos *(16.)*, scheiterte Rivaldo *(49./52.)*; DATE: *am Mittwoch*; SCORE/RESULT: Brasilien besiegt Schottland *2:1*, einen *2:1 (1:1)-Sieg*, der zwischenzeitliche *Ausgleich*;

- Relations for the detection of relevant relations – OPPONENTS: *Brasilien* besiegt *Schottland*, feierte *der Top-Favorit ...* einen glücklichen 2:1 (1:1)-Sieg über den *respektlosen Aussenseiter Schottland*; PLAYER_OF: hatte *Cesar Sampaio den vierfachen Weltmeister ...* in Führung gebracht, *Collins* gelang ... der zwischenzeitliche Ausgleich für *die Schotten*; TRAINER_OF: *Schottlands* Trainer *Brown*;

- Events for detecting relevant scenarios – GOAL: in der 4. Minute *in Führung gebracht*, das schnellste Tor ... *markiert*, Cesar Sampaio *köpfte* zum 1:0 *ein*; FOUL: als er den durchlaufenden Gallacher im Strafraum allzu energisch am Trikot *zog*; SUBSTITUTION: und musste in der 59. Minute für Crespo *Platz machen*.

In step 2) the templates for the soccer domain have been defined on the base of the corpus analysis and the design of the distinct IE-subtasks. The relations between the attributes of the templates are implicitly encoded in the hierarchy of domain-specific objects and events. An example of a soccer *entity* template is giving in figure 4, showing also how this specific entity template is being embedded in an *event* template, where *information-sharing* is provided for the value of identical attributes at distinct levels of embedding. The level of embedding itself is depending on the domain and the detail of the corpus analysis. In our case the top level is the one of a game of the championship, being identified by the date and the opponents involved (see figure 4).

For the third step defined in section 3. we defined the *linking types* on the base of the classification of the domain-specific verbs detected by the corpus analysis, taking into account the various verb frames, the polarity and the realization of certain modifiers within the sentential clauses under consideration. The top level linking type is called *soccer-lex* and associates at the abstract level a domain-specific verb with the general template *wm98-template* (first example in figure 5).

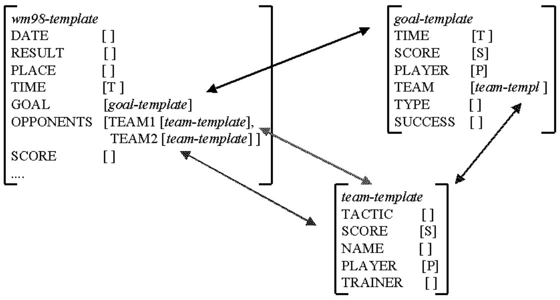

Figure 4: An *entity* template for the soccer domain: the TEAM-template and its embedding in various level of template definition (*event* and *scenario* templates), where the information-sharing is represented by variables

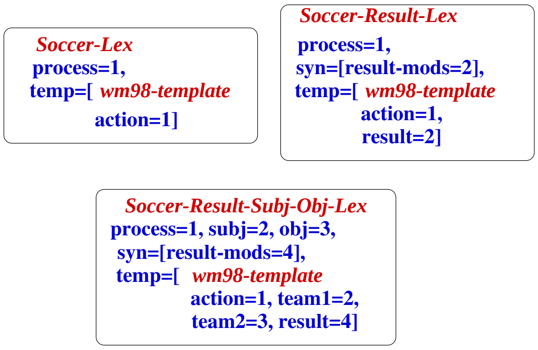

Figure 5: An example of hierarchy of linking types

On the base of further properties of the verb and the associated linguistic material within the clause boundaries, a subtyping of the main linking type is provided. So for a verb refering to a "game result", a linking type *soccer-result-lex* has been introduced corresponding to the classification of such verbs in the domain-lexicon. So for example: "entry=besieg, cat=v, dom=soccer, type=goal-subj-obj") where also the subcategorization information is playing a role, defining thus a further subtyping in the linking type hierarchy: *soccer-result-subj-obj-lex*. Not only verb arguments are taken into consideration for filling the domain templates, also adjuncts are playing a role for describing relations between entities, and for example lexically restricted PP modifications can be selected for detecting a template relevant entity.

5. How can Corpus Linguistics support further Developments of the SMES System

Once one has been going through the experience of analyzing collection of texts for designing the domain model for a specific application (as it has been described in section 4.), it becomes clear that advanced techniques for corpus analysis can considerably speed up this important task. Till now we have used only small "home-made" corpus analysis tools for establishing the frequency and distribution of certain (possibly morho-syntactically annotated) items within specific collection of texts in order to get lists of the most relevant terms to be integrated in a conceptual structure and for building the domain-lexicon. From more sophisticated CL tools and methods (extracting for example domain-specific sub-categorization frames) we expect to get better support for this task, being aware that larger annotated corpora will be necessary for (automatically) detecting some of the relations and events described in the last section.

An other very important aspect for improving the flexibility of IE systems is the possibility to *learn* from annotated and statistically processed very large corpora the templates to be filled by potential domains of application. This would also help IE technologies to leave the narrow view they might have on texts and to propose a generalized template filling strategy dealing with a larger number of (interrelated) domains and thus covering larger parts of unseen texts, escaping the limitations of the one (user-defined) domain of application, and offering extended querying possibilities through the texts analyzed. We might also expect a grouping of the detected templates as a preliminary step for the hierarchical organization of the templates relevant to a domain.

The remarks made above show that the annotation of large corpora should be quite complex and also linguistically accurate in order to help IE systems in the detection of domain-specific information structures. A fact which points to a possible circularity in the discussion on the relationships between IE and CL, since for the improvement of adaptive and flexible IE systems complex large linguistically annotated corpora are needed, but the task of annotating large corpora can be heavily supported by subsets of the technologies used in IE. But not every circularity is a vicious one and a well-defined interaction between IE and CL on the base of common technologies can probably lead to a well-balanced distribution of tasks: advanced and accurate shallow NL processing can help in generating more complex and linguistically annotated corpora, being the input of CL processing tools, which can again help IE in the (automatic) detection of (complex) domain information and its organization in hierarchical abstract template structures. The IE handling of texts with enriched domain information can then be used as well for providing new types of automatic annotation of corpora: semantic, domain and world-knowledge, which can support CL in fulfilling some of its specific tasks.

One of the crucial point concerns here the degree of maturity of the technologies involved: it is only when a high degree of accuracy is reached that a technology can be used for providing automatic support for some tasks or disciplines. We think that this degree of accuracy is reached for the shallow NL processing of running texts, but there is still some effort to be done in the context of detection and definition of the domain-specific template structures and their interplay, and we hope to have soon some significant progress coming out of the interaction of CL, IE and Machine Learning methodologies.

6. Conclusions and Future Work

We have presented our view – as developers of an adaptive IE-system – on some of the points where we think that IE and CL can (and should) more closely interact. Since modular IE systems separate the (generic) NL processing from the domain-modelling task, IE technologies are no longer limited to the application to a subcategory of texts, but can be used for providing (generic) linguistic annotations for all kinds of texts. The quality of the annotation depends on the accuracy of the NL processing tools involved, which in general have reached a good level of maturity for a larger number of languages and thus request only few manual post-editing. The remaining question will concern the definition and/or use of standards for the annotation.

As we have seen in sections 1. and 2., the declaratively defined mapping between NL tools and the conceptual structures allows to straightforward add semantic information and domain/world-knowledge into the annotations at the places where the IE system has detected relevant information to be extracted. We have still to look at how to integrate the additional information (encoded in form of typed feature structures) into the XML output of the generic linguistic processing.

We also expect from sophisticated CL methods to help in the task of automatically detect structures of relevant information and to guide those into multiple domain models, thus solving the problem of the limitation of the application IE system to a small amount of domains, offering till now only very partial semantic and pragmatic annotation possibilities. Once this task of (automatic) detection of domains of application and their organization into hierarchical template structures has been solved at a satisfying level of accuracy, this aspect of the IE methodology can be used for automatically supporting a more complete annotation of corpora wrt semantic and domain/world knowledge.

Through this paper we have been concerned only with technologies for written texts, but nowaday the question of the relationships between distinct Human Language Technologies (HLT) have to be seen in the context of multimodality and mult-imedia. The question on how to use IE for consistently annotating multi-media material (for subsequent querying) will be addressed in in project funded by the EC in which the SMES system will play a role and we hope to be able to present soon some results.

7. Acknowledgments

The research underlying this paper was supported by grants from the German Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie (BMB+F) to the DFKI project PARADIME (FKZ ITW 9704).

8. References

Krieger, Hans-Ulrich and U. Schaefer, 1994. $\mathcal{TDL}$—a type description language for constraint-based grammars. In *Proceedings of the 15th International Conference on Computational Linguistics, COLING-94*.

Neumann, Guenter, R. Backofen, J. Baur, M. Becker, and C. Braun, 1997. An information extraction core system for real world german text processing. In *Proceedings of the 5th Conference on Applied Natural Language Processing, ANLP-97*.

Piskorski, Jakub and G. Neumann, 2000. An intelligent text extraction and navigation system. In *Proceedings of the 6th Conference on Recherche d'Information Assistée par Ordinateur, RIAO-2000*.

SAIC (ed.), 1998. *Seventh Message Understanding Conference (MUC-7)*. http://www.muc.saic.com/: SAIC Information Extraction.