Integrating Shallow and Deep NLP for Information Extraction

Feiyu Xu and **Hans-Ulrich Krieger**
DFKI GmbH, Language Technology Lab
Stuhlsatzenhausweg 3
D-66123 Saarbrücken
Germany
{feiyu, krieger}@dfki.de

Abstract

This paper describes a novel approach to information extraction by developing strategies for combining techniques from shallow and deep NLP. We propose a hybrid template filling strategy, which employs shallow partial syntactic analysis for extracting local domain-specific relations and uses predicate-argument structures delivered by deep full-sentence analysis for extracting relations triggered by verbs. Heuristics have been developed for calling deep NLP on demand. The initial evaluation shows that the integration of deep analysis improves the performance of the scenario template-generation task.

1 Introduction

In current information extraction (IE) research, performance and domain adaptability are two essential issues. Pattern-based grammars embedded in IE systems, which employ finite-state techniques (Hobbs et al. 1996) and which are subsumed under the term shallow natural language processing (SNLP), often mix general linguistic information with domain-specific interpretation and are therefore not always portable. In addition, due to the inherent complexity of natural language, same semantic relations can be expressed in different syntactic forms: in particular, via linguistic constructions, such as long distance dependencies, passive, control/raising. Such constructions are not easy to capture by pattern-based grammars. In contrast to SNLP, "traditional" full sentence analysis, called deep NLP (DNLP), can, in principle, detect relationships expressed as complex constructions. Furthermore, most DNLP systems are based on linguistically-motivated grammars, covering a huge set of linguistic phenomena. Such grammars should be easier adapted to new domains and applications than the pattern-based grammars (Uszkoreit 2002). However, the scepticism of using DNLP in real-life applications results from behaving bad in efficiency and robustness, and also from the huge amount of ambiguous readings.

In the literature, there are several approaches to combing SNLP and DNLP. In the large Verbmobil project (Wahlster 2000), the deep parser runs in parallel to the shallow and statistical parsing components, embedded in a concurrent system architecture. Tsujii (2000) briefly describes an experiment of applying the combination of SNLP and DNLP to IE in the genome science domain. Riezler et al. (2002) present a stochastic system for parsing UPenn's Wall Street Journal (WSJ) treebank. The system combines full and partial parsing techniques by extending the full grammar with a grammar for fragment recognition.

In this paper, we present a new IE system *WHIES* (WHiteboard Information Extraction System), which tries to combine the best of SNLP and DNLP and to keep template-filling task independent of the general linguistic analysis. Our system is built on top of an integrated system called *WHAM* (WHiteboard Annotation Machine), which provides access to both shallow and deep analysis results. WHIES takes partial syntactic analyses given by SNLP as the primary analysis and integrates deep results only on demand. Its hybrid template-filling strategy uses two kinds of template-filling rules: *pattern-based* and *lexicalized unification-based* rules. The pattern-based rules are applied to SNLP results in order to guarantee efficient and robust recognition of domain-relevant local relations. The unification-based rules are applied to predicate-argument structures, which result from full-sentence parsing done by the deep HPSG parser. Given typed feature structures as our basic data structure for template representation, the merging of partially filled templates is based on the unification operation. Template merging is handled as a two-step constraint resolution process, namely, at sentence and discourse level. The initial evaluation shows that the integration of DNLP improves the performance of the IE task, in particular, in a domain where verbs play an important role.

The remainder of the paper is organized as follows. Section 2 describes WHAM. Section 3 discusses how we integrate DNLP on demand. Section 4 explains our approach. Section 5 gives an evaluation that illustrates performance improvements after the integration of DNLP. We close this paper by explaining future research ideas.

2 WHAM

WHAM (Crysmann et al. 2002) has implemented a hybrid system architecture for integrating SNLP and DNLP. WHAM provides access to linguistic analysis at different levels: tokens, morphological information, named entities, phrase chunks, sentence boundaries, and HPSG analysis results.

The basic strategy in WHAM can be simply stated as "shallow-guided" and "shallow-supported" deep parsing. The integration takes place at various levels: lexicon, named entities, phrase level, and topological structure. A German text is at first analysed by SPPC, a rule-based shallow system for German texts, performing tokenization, morphological analysis, POS filtering, named entity recognition, phrase recognition, and clause boundary recognition (Piskorski & Neumann 2000). WHAM passes the shallow analysis results for each sentence to a deep analyser, an efficient HPSG parser (Callmeier 2000) applied to the German grammar. The semantic analysis of the deep parser uses a kind of underspecified semantic representation, called MRS (minimal recursive semantics); see (Copestake et al. 1999).

3 Integrating DNLP on Demand

Shallow IE methods have been proven to be sufficient to deal with extraction of relationships among chunks, expressed relatively locally and explicitly (Grishman 1997). Normally, the interpretation of a sequence of chunks by SNLP is unambiguous and domain-specific, e.g., the relationships between a noun phrase (NP) and its adjacent prepositional phrase (PP modifier) or its adjacent NP (appositive modifier). For DNLP, the decision of the attachment of modifiers is very difficult, and thus, their analysis often ambiguous. Nevertheless, deep grammars are more suitable to express precise relationships between verbs and their arguments in complex linguistic constructions, involving, e.g., passive, free word order, long-distance dependencies and control/raising. For example, sentence (1) contains a passive and a control construction. The relationship between *Hans Becker* and the division name *Presseabteilung* cannot be formulated easily by regular expressions. In particular, the relatively free word order of German allows reversing the order of the two names, by keeping the same meaning; see (2).

(1) Hans Becker wurde aufgrund des Rücktritts von Peter Müller gebeten, die Presseabteilung zu übernehmen.
Hans Becker was due to the resignation of Peter Müller asked, to take over the press division.

(2) Aufgrund des Rücktritts von Peter Müller wurde Hans Becker gebeten, die Presseabteilung zu übernehmen.
Due to the resignation of Peter Müller Hans Becker was asked, to take over the press division.

In comparison to most shallow approaches, our DNLP system can recognize the embedded relationships in (1) and (2) straightforwardly, normalizing them into a predicate-argument structure. Although some of the shallow systems perform also full sentence analysis, most of them (like SPPC) provide only partial analysis and cannot capture these kinds of embedded relationships without any additional efforts (Grishman 1995).

Given the pros and cons of shallow and deep analysis, we decide to use shallow analysis as our primary linguistic resources to recognize local realtionships and have developed heuristics, which are used to trigger DNLP only on demand.

In (Xu et al. 2002), a semi-supervised method was developed to recognize domain-relevant terms (including term collocations) and their relations. Each term is assigned a relevance weight. An interesting observation is that the distribution of relevant terms in a specific domain is related to the POS information. For example, in the stock market and the crime drug domain, most relevant terms are nouns, while verbs play an important role in the management succession domain. This observation is a good indicator for deciding whether and when DNLP should be integrated to into IE for a new domain. If the domain-relevant terms are mostly verbs, we suggest to integrate DNLP for obtaining predicate-argument structures, since relationships triggered by the verbs can be expressed in various syntactical forms and cannot be easily covered by a small set of pattern-based rules. For example, sentence (3) and (4) express the same meaning, but in different word orders, as (1) and (2).

(3) Generaldirektor Eugen Krammer (59), ..., wird per 31. Mai 1997 aus seinen Funktionen ausscheiden.
General manager Eugen Krammer (59), ..., will resign from his office on May 31. 1997

(4) Aus seinen Funktionen wird Generaldirektor Eugen Krammar (59)...., per 31. Mai 1997 ausscheiden.
General manager Eugen Krammer (59), ..., will resign from his office on May 31. 1997

Both of them are about resignment of the person *Eugen Krammer*. The domain-relevant verb predicate “ausscheiden” (resign) triggers the resignment relation, taking *Eugen Krammer* as argument. In this case, DNLP can detect the predicate-argument structures in (3) and (4). Although (3) and (4) have different surface constructions, only a single rule has to be defined, which maps the argument of the predicate “ausscheiden” to its domain role.

In comparsion to verbs, nouns (incl., nominalization of verbs) and adjectives are good indicators for pattern-based rules, which are suitable to deal with local relationships expressed by complex noun phrases, containing PP-attachment and appositions. (5) and (6) give examples of adjectives and nouns as trigger words in the management succession domain.

(5) Der bisherige Vorstandsvorsitzende des Auto-Zulieferers Kolbenschmidt, Heinrich Binder, ...
The previous president of car supplier Kolbenschmidt, Heinrich Binder

(6) Nachfolger vom Amtsinhaber Hans Günter Merk
Successor of the office holder Hans Günter Merk

Thus, we take relevant verbs as clues for deciding when or whether to trigger DNLP during online processing: if a sentence contains relevant verb terms in addition to relevant nouns and adjectives, it will also be passed to DNLP; otherwise, SNLP will be sufficient.

4 WHIES

In this section, we will give a detailed description of our IE system *WHIES* and will explain the hybrid template filling strategy, plus the two-step template merging component.

4.1 WHIES Architecture

As depicted in Figure 1, WHIES consists of two main components for *template filling* and *template merging*. Template filling takes WHAM analysis results as input (chunks and MRSs) and employs a hybrid template-filling strategy. Template merging attempts to unify the partially filled templates first at the sentence level and later at the discourse level.

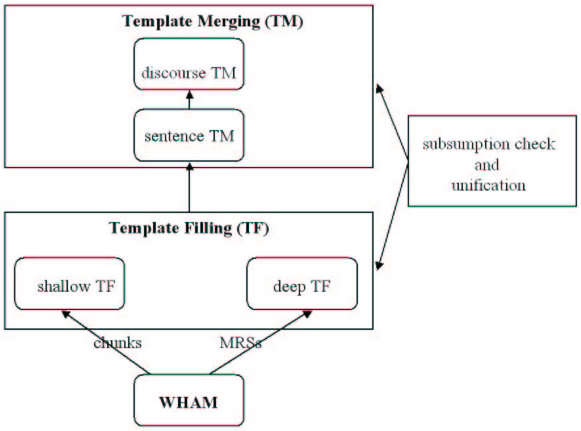

Figure 1: Architecture of WHIES

Both WHAM and WHIES use the Java typed feature structure package JTFS (Krieger 2002), which allows the online construction of templates and the dynamic extension of the type hierarchy, an important feature to deal with unknown words in IE. JTFS provides basic data structures (TFSs) and operations (subsumption check and unification).

4.2 A Hybrid Template Filling Strategy

The linguistic annotations provided by WHAM are domain independent. Our hybrid strategy allows two kinds of template-filling rules, which map general linguistic analysis to domain-specific interpretations:

- pattern-based template filling rules (P-rules)[1]
- lexicalized unification-based rules (U-rules)

Here we use the management succession domain for presenting our ideas. P-rules are applied to shallow results, in particular to tokens, lexical items, named entities and phrases, using relevant adjectives and nouns as trigger terms. A P-rule consists of two parts: the left-hand side is a regular expression over TFSs, whereas the right-hand side is a TFS, corresponding to a partially-filled scenario template, e.g.,

(7) *Rücktritt von* $\boxed{1}$Person → [Person_out $\boxed{1}$]

(7) matches an expression which contains two tokens, *Rücktritt* (retirement) and *von* (of), followed by a person name, and fills the slot Person_out. *Rücktritt* is the trigger word. Applying (7) to the shallow analysis of sentence (1) and (2), the Person_out slot of the template is then filled with the name *Peter Müller*. The SProUT system described in (Becker et al. 2002) supports the definition of P-rules.

A U-rule makes use of the predicate-argument structures embedded in MRSs, provided by the deep HPSG parser. Hence, a U-rule might look as follows:

(8) [Semantics [Pred *übernehm* (take over),
Agent $\boxed{1}$,
Theme $\boxed{2}$]]
→ [IE_Template [Person_in $\boxed{1}$,
Division $\boxed{2}$]]

Applying (8) to the deep analysis of (1) or (2), the Person_in slot is filled with *Hans Becker* and the Division slot with *Presseabteilung*.

In fact, our hybrid template-filling strategy can also be directly applied to a relatively deep SNLP system, which can provide predicate-argument structures in addition to fragments.

4.3 Template Merging

Given partially filled templates, the next step is to combine them into reasonable scenario template instances. In the literature, only little information is reported on template merging strategies (Hirschman 1992; Hobbs et al. 1996; Appelt and Israel 1999; Surdeanu and Harabagiu 2001). Our motivation for template merging comes from the need to properly integrate the partially filled templates, returned by the shallow and the deep processor. As shown in Figure 1, the partially filled templates are merged initially at the sentence level and are then combined at the discourse level. The domain dependent constraints are formulated as template merging rules. (9) is a merging rule, saying that the same person in the management succession domain cannot accept and resign the same position at the same time (we use the negation sign to indicate feature structure inequality).

(9) [Person_in $\boxed{1}$,
Person_out $\neg\boxed{1}$]

Merging at the Sentence Level

For each sentence, we have two sets of partially filled templates, one originating from shallow template filling (*ST*) and another one from deep template filling (*DT*). Our current merging algorithm takes $SDT := ST \cup DT$ as input and works as follows (we treat the set as an ordered sequence):

```
1. start with an empty set for the merging result Result.
2. loop1: for (i=0 ; i< size of SDT ; i++) {
```

[1] Shallow template filling corresponds to the *scenario pattern matching* component of the IE system presented by Grishman (1997).

```
        loop2: for (m=i+1; m < size of SDT ; m++){
            unify $T_i$ with $T_m$ and the template-merging rules:
                $RT = T_i \wedge T_m \wedge MR$,
            where $T_i$, $T_m \in SDT$; $\wedge$ denotes TFS unification;
            and $MR$ abbreviates the unification of all template
            merging rules (a constant TFS);
            if unification succeeds,
              then add $RT$ to Result and break loop2;
        } // end loop2
        if all unifications fail, then add $T_i$ to Result;
    } // end loop 1
3.  apply subsumption check to the templates in Result,
    until the most specific templates remain.
4.  if Result == SDT,
      then return Result;
      else SDT = Result and goto 1.
```

Our algorithm allows multiple templates in the output and ensures that each partially filled template contributes only once to a final template — each template is intended to describe a single elementary event. If a template cannot unify with other templates, it will be kept in the result and respresents a result template. Termination is guaranteed, since the two for-loops yield *Result* whose cardinality is smaller or equal than *SDT*.

Merging at the Discourse Level

The merging process at the discourse level is modelled as an incremental construction of domain relevant information across sentences, compatible with Discourse Representation Theory (DRT) (Kamp & Reyle 1993). The construction process is shown in Figure 2.

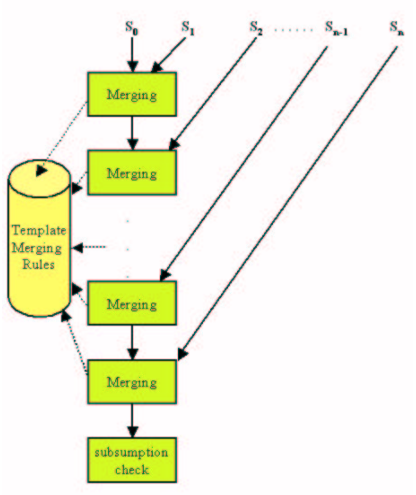

Figure 2. Merging templates across sentences.

The merging results of previous sentences and the template set from the current sentence are then used as input for a discourse level merging operation. The merging algorithm between two sets of templates is very similar to the above algorithm. After each merging operation, a heuristics is applied to fill additional template slots. According to DRT, the earlier introduced named entities can be referred to later in the text to account for the gaps in the discourse. In the example domain, the organization names are often mentioned at the beginning of a text. Hence, the organization slot of templates in subsequent sentences cannot be filled. To cope with this problem, we build a stack of organization names for the whole discourse. Now, if the organization slot of a template is still not filled after the merging operation, it will be filled with the top-most compatible element of the stack.

5 Initial Evaluation

We have conducted initial evaluations to see which kinds of improvement of template generation performance one can expect from such a hybrid IE architecture. Standard precision/recall metrics are used in the evaluation. The sentences in the example domain from DPA (German Press Agency) are manually annotated with partially-filled templates, which are manually extracted, based on shallow and deep template filling rules. Template filling rules are semi-automatically constructed by using the relevant terms extracted by the tool mentioned in section 2. The main reason for manual annotation is that our deep grammars are not yet fully adapted to the chosen domain and the syntactic analysis provides still too many readings.[2] The total corpus contains 299 documents. Each document contains 3 to 5 sentences. The average length of a sentence is 17 words.

For the evaluation of template merging at sentence level, we chose the 50 top-relevant sentences. The relevance of a sentence is calculated in terms of the relevance of its containing verbs, nouns, and adjectives. Three scenarios are constructed for the evaluation, viz., applying sentential merging to (i) templates filled by shallow rules (*shallow*), (ii) templates filled by deep rules (*deep*), and (iii) union of (i) and (ii) (*shallow+deep*).

shallow		*deep*		*shallow + deep*	
cover	recall	cover	recall	cover	recall
0.56	0.31	0.94	0.68	0.96	0.92

Table 1. Merging at sentence level.

In the shallow scenario, nouns and adjectives are trigger words, while verbs are indicators for the deep scenario. Table 1 summarizes the evaluation result. Since the partially-filled templates are manually annotated and the number of STs and DTs occurring in a sentence is small, the precision of all three scenarios turns out to be 1.0. Coverage (cover) is defined as the percentage of sentences, from which different kinds of filling rules (shallow, deep, shallow+deep) can extract templates.

The evaluation results confirm the assumption, that the integration of DNLP is useful, if verbs play an important role in the domain. Table 1 shows that 94% of the relevant sentences contain relevant verbs. In addition, the annotation shows that information extracted by *shallow* and *deep* is complementary to each other. Hence, the recall value of *shallow+deep* is almost the sum of *shallow* and *deep*.

[2] In WHIES, we preliminarily choose the first reading, which is not always the best reading for the domain.

6 Conclusion and Future Work

We have presented strategies for extracting domain-relevant templates via a combination of different linguistic resources, namely, SNLP and DNLP. The initial evaluation shows that DNLP improves the performance of the IE task, in particular, in domains where verbs play an important role. In fact, our hybrid template filling strategy can also be applied to standard shallow systems, which can provide predicate-argument structures in addition to chunks. The template merging component is also a general approach to combining partially filled templates represented as TFSs.

There are still many interesting questions to pursue to make DNLP more attractive for domain-adaptive information extraction. Given the domain-relevant verbs, a further step can be the automatic acquisition of their domain-specific subcategorization frames and the learning of domain-relevant and domain-specific grammar constructions. We believe that further experiments need to be conducted to discover also linguistic-driven heuristics for the online integration of DNLP. Furthermore, alternative template-merging strategies and missing discourse components must be developed to improve the performance.

We also suggest to replace standard unification between T_i and T_m in the above algorithm by a kind of prioritized default unification $\wedge_D$, which takes care of the "origin" of its arguments. In case that both arguments either originate from SNLP or DNLP, $\wedge_D$ behaves like standard unification. However, if their origin differs and conflicts occur at specific features in the TFS, the information from DNLP is preferred during unification, due to its high precision. $\wedge_D$ is related to Kaplan's *priority union* (Kaplan, 1987), except that leaves in his feature structures are atoms, whereas leaves here are usually types, which can be made specific during TFS unification. However, the wellformedness constraints, imposed by the template merging rules, are still integrated by standard unification, since they express some domain-specific truth. Hence, the computation of *RT* in the merging algorithm changes to

$$RT := (T_i \wedge_D T_m) \wedge MR$$

The application of $\wedge_D$ can be triggered, for instance, by distance of the two templates T_i and T_m in the sentence, i.e., by the overlap of their spanning text fragments.

In the near future, an automatic evaluation of the hybrid system is planned. In addition, we want to evaluate also domains where verbs donot play a dominant role.

Acknowledgement

WHITEBOARD is a DFKI project supported by BMBF. The work reported here is part of the WHITEBOARD project. Thanks to WHITEBOARD and SPROUT colleguages for useful discussions and technical supports.

References

(Becker et al. 2002) M. Becker, W. Drozdzynski, H.U. Krieger, J. Piskorski, U. Schäfer, F. Xu. *SProUT - Shallow Processing with Typed Feature Structures and Unification*. Proceedings of ICON 2002, Mumbai, Indian, 2002.

(Callmeier 2000) U. Callmeier. *PET–A Platform for Experimentation with Efficient HPSG Processing Techniques*. Natural Language Engineering, 6(1), 99–108, 2000.

(Crysmann et al. 2002) B. Crysmann, A. Frank, B. Kiefer, S. Müller, G. Neumann, J. Piskorski, U. Schäfer, M. Siegel, H. Uszkoreit, F. Xu, M. Becker and H.-U. Krieger. *An Integrated Architecture for Shallow and Deep Processing*. Proceedings of ACL, 2002.

(Copestake et al. 1999) A. Copestake, D. Flickinger, I. Sag and C. Pollard, draft. *Minimal Recursion Semantics: An introduction*. http://www-csli.stanford.edu/~aac/papers/newmrs.pdf, 1999.

(Grishman 1995) R. Grishman. *The NYU system for MUC-6 or where's the syntax?* Proceedings of MUC-6, Columbia, MD, Morgan Kaufmann, 1995.

(Grishman 1997) R. Grishman. *Information Extraction: Techniques and Challenges*. In M. T. Pazienza, ed., Information Extraction. Springer-Verlag, Lecture Notes in Artificial Intelligence, Rome, 1997.

(Hirschman 1992) L. Hirschman. *An Adjunct Test for Discourse Processing in MUC-4*. Proceedings of MUC-4, 67–77, Morgan Kaufmann, 1992.

(Hobbs et al. 1996) J. Hobbs, D. Appelt, J. Bear, D. Israel, M. Kameyama, M. Stickel, and M. Tyson. *FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text*. Finite State Devices for Natural Language Processing. E. Roche and Y. Schabes, eds. MIT Press, 1996.

(Kamp and Reyle 1993) H. Kamp and U. Reyle. *From Discourse to Logic*. Kluwer, 1993.

(Kaplan 1987) R.M. Kaplan. *Three Seductions of Computational Psycholinguistics*. Linguistic Theory and Computer Applications. P. Whitelock, M.M. Wood, H.L. Somers, R. Johnson, and P. Bennett, eds. Academic Press, 1987.

(Kiefer et al. 2000) B. Kiefer, H.-U. Krieger, M.-J. Nederhof. *Efficient and Robust Parsing of Word Hypotheses Graphs*. Wahlster (2000), p. 280–295, 2000.

(Krieger 2002) H.-U. Krieger. *JTFS–A Java Implementation of Typed Feature Structures*. Technical Report, DFKI, 2002.

(Piskorski and Neumann 2002) J. Piskorski and G. Neumann. *An Intelligent Text Extraction and Navigation System*. In Proceedings of RIAO-2000, Paris, 2000.

(Riezler et al. 2002) S. Riezler, T.H. King, R.M. Kaplan, R. Crouch, J.T. Maxwell III, and M. Johnson. *Parsing the Wall Street Journal using a Lexical-Functional Grammar and Discriminative Estimation Techniques*. Proceedings of ACL, 271–278, 2002.

(Surdeanu and Harabagiu 2001) M. Surdeanu and S. Harabagiu. *Infrastructure for Open-Domain Information Extraction*. Proceedings of HLT 2002, San Diego, California, 2001.

(Tsujii 2000) J. Tsujii. *Generic NLP Technologies: Language, Knowledge and Information Extraction*. Proceedings of ACL, p. 11–18, 2000.

(Uszkoreit 2002) H. Uszkoreit *New Chances for Deep Linguistic Processing*. Proceedings of COLING 2002, Taipei, 2002.

(Wahlster 2000) W. Wahlster ed. *Verbmobil: Foundation of Speech-to-Speech Translation*. Springer, 2000.

(Xu et al. 2002) F. Xu, D. Kurz, J. Piskorski, S. Schmeier. *A Domain Adaptive Approach to Automatic Acquisition of Domain Relevant Terms and their Relations with Bootstrapping.* Proceedings of LREC 2002, Spain, 2002.