Integrating Information Extraction and Automatic Hyperlinking

Stephan Busemann, Witold Drożdżyński, Hans-Ulrich Krieger,
Jakub Piskorski, Ulrich Schäfer, Hans Uszkoreit, Feiyu Xu
German Research Center for Artificial Intelligence (DFKI GmbH)
Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany
sprout@dfki.de

Abstract

This paper presents a novel information system integrating advanced information extraction technology and automatic hyper-linking. Extracted entities are mapped into a domain ontology that relates concepts to a selection of hyperlinks. For information extraction, we use SProUT, a generic platform for the development and use of multilingual text processing components. By combining finite-state and unification-based formalisms, the grammar formalism used in SProUT offers both processing efficiency and a high degree of decalrativeness. The ExtraLink demo system showcases the extraction of relevant concepts from German texts in the tourism domain, offering the direct connection to associated web documents on demand.

1 Introduction

The utilization of language technology for the creation of hyperlinks has a long history (e.g., Allen et al., 1993). Information extraction (IE) is a technology that can be applied to identifying both sources and targets of new hyperlinks. IE systems are becoming commercially viable in supporting diverse information discovery and management tasks. Similarly, automatic hyperlinking is a maturing technology designed to interrelate pieces of information, using ontologies to define the relationships. With ExtraLink, we present a novel information system that integrates both technologies in order to reach at an improved level of informativeness and comfort. Extraction and link generation occur completely in the background. Entities identified by the IE system are mapped into a domain ontology that relates concepts to a structured selection of predefined hyperlinks, which can be directly visualized on demand using a standard web browser. This way, the user can, while reading a text, immediately link up textual information to the Internet or to any other document base without accessing a search engine.

The quality of the link targets is much higher than with standard search engines since, first of all, only domain-specific interpretations are sought, and second, the ontology provides additional structure, including related information.

ExtraLink uses as its IE system SProUT, a generic multilingual shallow analysis platform, which currently provides linguistic processing resources for English, German, Italian, French, Spanish, Czech, Polish, Japanese, and Chinese (Becker et al., 2002). SProUT is used for tokenization, morphological analysis, and named entity recognition in free texts. In Section 2 to 4, we describe innovative features of SProUT. Section 5 gives details about the ExtraLink demonstrator.

2 Integrating Typed Feature Structures and Finite State Machines

The main motivation for developing SProUT comes from the need to have a system that (i) allows a flexible integration of different processing modules and (ii) to find a good trade-off between processing efficiency and linguistic expressiveness. On the one hand, very efficient *finite state* devices have been successfully applied to real-world applications. On the other hand, *unification-based grammars* (UBGs) are designed to capture fine-grained syntactic and semantic constraints, resulting in better descriptions of natural language phenomena. In contrast to finite state devices, unification-based grammars are also assumed to be more transparent and more easily modifiable. SProUT's mission is to take the best from these two worlds, having a finite state machine that operates on typed feature structures (TFSs). I.e., transduction rules in SProUT do not rely on simple atomic symbols, but instead on TFSs, where the left-hand side of a rule is a regular expression over TFSs, representing the recognition pattern, and the right-hand side is a sequence of TFSs, specifying the output structure. Consequently, equality of atomic symbols is replaced by *unifiability* of TFSs and the output is constructed using TFS *unification* w.r.t. a type hierarchy. Such rules not only recognize and classify patterns, but also extract fragments embedded in the patterns and fill output templates with them.

Standard finite state techniques such as minimization and determinization are no longer applicable here, due to the fact that edges in our automata are annotated by TFSs, instead of atomic symbols. However, not every outgoing edge in such an automaton must be analyzed, since TFS annotations can be arranged under subsumption, and the failure of a general edge automatically causes the failure of several, more specialized edges, without applying the unifiability test. Such information can in fact be precompiled. This and other optimization techniques are described in (Krieger and Piskorski, 2003).

When compared to symbol-based finite state approaches, our method leads to smaller grammars and automata, which usually better approximate a given language.

3 XTDL – The Formalism in SProUT

XTDL combines two well-known frameworks, viz., typed feature structures and regular expressions. XTDL is defined on top of TDL, a definition language for TFSs (Krieger and Schäfer, 1994) that is used as a descriptive device in several grammar systems (LKB, PAGE, PET).

Apart from the integration into the rule definitions, we also employ TDL in SProUT for the establishment of a type hierarchy of linguistic entities. In the example definition below, the *morph* type inherits from *sign* and introduces three more morphologically motivated attributes with the corresponding typed values:

```
morph := sign & [ POS  atom, STEM atom, INFL infl ].
```

A rule in XTDL is straightforwardly defined as a recognition pattern on the left-hand side, written as a regular expression, and an output description on the right-hand side. A named label serves as a handle to the rule. Regular expressions over TFSs describe sequential successions of linguistic signs. We provide a couple of standard operators. Concatenation is expressed by consecutive items. Disjunction, Kleene star, Kleene plus, and optionality are represented by the operators **|**, *****, **+**, and **?**, resp. **{*n*}** after an expression denotes an *n*-fold repetition. **{*m*,*n*}** repeats at least *m* times and at most *n* times.

The XTDL grammar rule below may illustrate the syntax. It describes a sequence of morphologically analyzed tokens (of type *morph*). The first TFS matches one or zero items (**?**) with part-of-speech *Determiner*. Then, zero or more *Adjective* items are matched (*****). Finally, one or two *Noun* items (**{1,2}**) are consumed. The use of a variable (e.g., **#1**) in different places establishes a coreference between features. This example enforces agreement in case, number, and gender for the matched items. Eventually, the description on the RHS creates a feature structure of type *phrase*, where the category is coreferent with the category *Noun* of the right-most token(s), and the agreement features corefer to features of the *morph* tokens.

```
np :>
  (morph & [    POS  Determiner,
             INFL  [CASE #1, NUM #2, GEN #3 ]] )?
  (morph & [    POS  Adjective,
             INFL  [CASE #1, NUM #2, GEN #3 ]] )*
  (morph & [    POS  Noun & #4,
             INFL  [CASE #1, NUM #2, GEN #3 ]] ){1,2}
-> phrase & [CAT #4,
             AGR agr & [CASE #1, NUM #2, GEN #3 ]].
```

The choice of TDL has a couple of advantages. TFSs as such provide a rich descriptive language over linguistic structures and allow for a fine-grained inspection of input items. They represent a generalization over pure atomic symbols. Unifiability as a test criterion in a transition is a generalization over symbol equality. Coreferences in feature structures express structural identity. Their properties are exploited in two ways. They provide a stronger expressiveness, since they create dynamic value assignments on the automaton transitions and thus exceed the strict locality of constraints in an atomic symbol approach. Furthermore, coreferences serve as a means of information transport into the output description on the RHS of the rule. Finally, the choice of feature structures as primary citizens of the information domain makes composition of modules very simple, since input and output are all of the same abstract data type.

Functional (in contrast to regular) operators are a door to the outside world of SProUT. They either serve as predicates, helping to locate complex tests that might cancel a rule application, or they construct new material, involving pieces of information from the LHS of a rule. The sketch of a rule below transfers numerals into their corresponding digits using the functional operator normalize() that is defined externally. For instance, "one" is mapped onto "1", "two" onto "2", etc.

```
… numeral & [ SURFACE #surf, ... ] …. ->
digit & [ ID #id, ... ], where #id = normalize(#surf).
```

4 The SProUT System

The core of SProUT comprises of the following components: (i) a finite-state machine toolkit for building, combining, and optimizing finite-state devices; (ii) a flexible XML-based regular compiler for converting regular patterns into their corresponding compressed finite-state representation (Piskorski et al., 2002); (iii) a JTFS package which provides standard operations for constructing and manipulating TFSs; and (iv) an XTDL grammar interpreter.

Currently, SProUT offers three online components: a tokenizer, a gazetteer, and a morphological analyzer. The tokenizer maps character sequences to tokens and performs fine-grained token classification. The gazetteer recognizes named entities based on static named entity lexica.

The morphology unit provides lexical resources for English, German (equipped with online shallow compound recognition), French, Italian, and Spanish, which were compiled from the full form lexica of MMorph (Petitpierre and Russell, 1995). Considering Slavic languages, a component for Czech presented in (Hajič, 2001), and Morfeusz (Przepiórkowski and Wolinski, 2003) for Polish. For Asian languages, we integrated Chasen (Asahara and Matsumoto, 2000) for Japanese and Shanxi (Liu, 2000) for Chinese.

The XTDL-based grammar engineering platform has been used to define grammars for English, German, French, Spanish, Chinese and Japanese allowing for named entity recognition and extraction. To guarantee a comparable coverage, and to ease evaluation, an extension of the MUC-7 standard for entities has been adopted.

```
ne-person := enamex & [ TITLE list-of-strings,
                        GIVEN_NAME list-of-strings,
                        SURNAME list-of-strings,
                        P-POSITION list-of-strings,
                        NAME-SUFFIX string,
                        DESCRIPTOR string ].
```

Given the expressiveness of XTDL expressions, MUC-7/MET-2 named entity types can be enhanced with more complex internal structures. For instance, a person name *ne-person* is defined as a subtype of *enamex* with the above structure.

The named entity grammars can handle types such as person, location, organization, time point, time span (instead of date and time defined by MUC), percentage, and currency.

The core system together with the grammars forms a basis for developing applications. SProUT is being used by several sites in both research and industrial contexts.

A component for resolving coreferent named entities disambiguates and classifies incomplete named entities via dynamic lexicon search, e.g., *Microsoft* is coreferent with *Microsoft corporation* and is thus correctly classified as an organization.

5 ExtraLink: Integrating Information Extraction and Automatic Hyperlinking

A methodology for automatically enriching web documents with typed hyperlinks has been developed and applied to several domains, among them the domain of tourism information. A core component is a domain ontology describing tourist sites in terms of sights, accommodations, restaurants, cultural events, etc. The ontology was specialized for major European tourism sites and regions (see Figure 1). It is associated with a large selection of

Figure 1: Link Target Page (excerpt). The instance the web document is associated to (Isle of Capri) is shown on the left, together with neighboring concepts in the ontology, which the user can navigate through.

link targets gathered, intellectually selected and continuously verified. Although language technology could also be employed to prime target selection, for most applications quality requirements demand the expertise of a domain specialist. In the case of the tourism domain, the selection was performed by a travel business professional. The system is equipped with an XML interface and accessible as a server.

The ExtraLink GUI marks the relevant entities (usually locations) identified by SProUT (see second window on the left in Figure 2). Clicking on a marked expression causes a query related to the entity being shipped to the server. Coreferent concepts are handled as expanded queries. The server returns a set of links structured according to the ontology, which is presented in the ExtraLink GUI (Figure 2). The user can choose to visualize any link target in a new browser window that also shows the respective subsection of the ontology in an indented tree notation (see Figure 1).

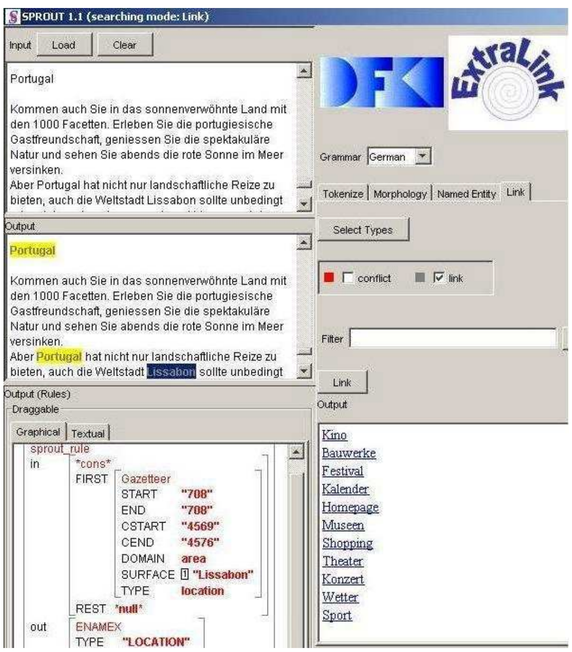

Figure 2: ExtraLink GUI. The links in the right-hand window are generated after clicking on the marked named entity for Lisbon (marked in dark). The bottom left window shows the SProUT result for “Lissabon”.

The ExtraLink demonstrator has been implemented in Java and C++, and runs under both MS Windows and Linux. It is operational for German, but it can easily be extended to other languages covered by SProUT. This involves the adaptation of the mapping into the ontology and a multilingual presentation of the ontology in the link target page.

Acknowledgements

Work on ExtraLink has been partially funded through grants by the German Ministry for Education, Science, Research and Technology (BMBF) to the project Whiteboard (contract 01 IW 002), by the EC to the project Airforce (contract IST-12179), and by the state of the Saarland to the project SATOURN. We are indebted to Tim vor der Brück, Thierry Declerck, Adrian Raschip, and Christian Woldsen for their contributions to developing ExtraLink.

References

J. Allen, J. Davis, D. Krafft, D. Rus, and D. Subramanian. *Information agents for building hyperlinks*. J. Mayfield and C. Nicholas: Proceedings of the Workshop on Intelligent Hypertext, 1993.

M. Asahara and Y. Matsumoto. *Extended models and tools for high-performance part-of-speech tagger*. Proceedings of COLING, 21-27, 2000.

M. Becker, W. Drożdżyński, H.-U. Krieger, J. Piskorski, U. Schäfer, F. Xu. *SProUT–Shallow Processing with Typed Feature Structures and Unification*. In Proceedings of ICON, 2002.

J. Hajič. *Disambiguation of rich inflection–computational morphology of Czech*. Prague Karolinum, Charles University Press, 2001.

H.-U. Krieger and U. Schäfer. *TDL–A Type Description Language for Constraint-Based Grammars*. Proceedings of COLING, 893-899, 1994.

H.-U. Krieger and J. Piskorski. *Speed-up methods for complex annotated finite state grammars*. DFKI Report, 2003.

K. Liu. *Research of automatic Chinese word segmentation*. Proceedings of ILT&CIP, 2001.

D. Petitpierre and G. Russell. *MMORPH–the Multext morphology program*. Multext deliverable report 2.3.1. ISSCO, University of Geneva, 1995.

J. Piskorski, W. Drożdżyński, F. Xu and O. Scherf. *A flexible XML-based regular compiler for creation and converting linguistic resources*. Proceedings of LREC 2002, Las Palmas, Spain, 2002.

A. Przepiórkowski and M. Wolinski. *The Unbearable Lightness of Tagging: A Case Study in Morphosyntactic Tagging of Polish*. Proceedings of the Workshop on Linguistically Interpreted Corpora, 2003.