Information Extraction from German Patient Records via Hybrid Parsing and Relation Extraction Strategies

♯Hans-Ulrich Krieger, ♯Christian Spurk, ♯Hans Uszkoreit, ♯Feiyu Xu, ♯Yi Zhang,
⋆Frank Müller, ⋆Thomas Tolxdorff

♯German Research Center for AI
Stuhlsatzenhausweg 3, 66123 Saarbrücken
FirstName.LastName@dfki.de

⋆Institut für Med. Informatik/Charité
Hindenburgdamm 30, 12200 Berlin
{ *f.mueller, thomas.tolxdorff* }@charite.de

Abstract

In this paper, we report on first attempts and findings to analyzing German patient records, using a hybrid parsing architecture and a combination of two relation extraction strategies. On a practical level, we are interested in the extraction of concepts and relations among those concepts, a necessary cornerstone for building medical information systems. The parsing pipeline consists of a morphological analyzer, a robust chunk parser adapted to Latin phrases used in medical diagnosis, a repair rule stage, and a probabilistic context-free parser that respects the output from the chunker. The relation extraction stage is a combination of two systems: *SProUT*, a shallow processor which uses hand-written rules to discover relation instances from local text units and *DARE* which extracts relation instances from complete sentences, using rules that are learned in a bootstrapping process, starting with semantic seeds. Two small experiments have been carried out for the parsing pipeline and the relation extraction stage.

Keywords: information extraction from patient records; hybrid parsing pipeline; hybrid relation extraction strategy.

1. Overview

In recent years, natural language processing in general and information extraction (IE) in particular have been identified as distinguishing frameworks for analyzing and processing clinical texts (Geibel et al., 2013; Goodwin and Harabagiu, 2013; Roberts et al., 2013). One important application deals with the extraction of concepts and relations among concepts from patient records for building medical information systems, such as patient record search engines, patient recruitment information systems, and health information mining systems.

In this paper, we will describe our approach to German patient records. Patient records written or formulated by medical doctors have the following key characteristics:

- There are no uniform or official definitions of structure and form what a patient record should look like.
- The records are dominated by free texts, but often contain some structured data, like tables.
- Many text fragments are not formulated in complete and well-formed sentences. They are often in telegraphic style, sometimes containing only keywords.
- Many sentences are very long, containing several subordinate clauses.
- There is no uniform definition of a sentence marker. Thus, there are often no clear separators among the sentences.
- There are a lot of medical terms occurring in the texts. Thus, the patient records are very domain dependent.
- Sentences often contain vague formulations, such as assumption, speculation, or uncertainties.

In order to deal with this specific and difficult genre, we have applied a *hybrid parsing strategy* that combines robust chunk parsing and deep parsing in a prototypical system. Our parsing strategy integrates chunks, delivered by a chunk parser as well as unrecognized tokens within the same PCFG parser, thus going beyond the standard IE pipeline.

Parallel to the hybrid parsing strategy, we have also developed a *hybrid relation extraction strategy*,

1. by applying lexico-syntactic patterns to extracted relation mentions occurring in local text fragments based on chunking and named entity recognition results via the rule-based shallow processing system *SProUT*, and
2. by using relation extraction, building on named entity recognition and full parsing results, in which the relation extraction rules are learned automatically, utilizing the minimally-supervised machine learning system *DARE*.

2. Hybrid Parsing Strategy

The below subsections give an overview of the hybrid parsing pipeline.

2.1. Morphological Analysis

The morphological analyzer is responsible for the segmentation and tokenization of input sequences of characters into sequences of linguistic tokens. While this step is usually conceived as trivial and not complicated for Indo-European languages such as English and German, corner tricky cases do exist, e.g., for the handling of punctuation marks, multiword expressions, compounding words, etc. A whitespace based tokenization accompanied by specific rules delivers linguistic tokens for the next phase of processing, part-of-speech (POS) tagging and chunking.

2.2. Robust Chunk Parser

The goal of the chunking system is to map the free texts of clinical documents onto the abstract concepts of a medical ontology. The most simple solution would be a kind of a bag-of-words approach in which essentially all *content* words occurring in the texts were mapped onto medical concepts without taking into account their linguistic relations. But inherent linguistic structures and contexts are very useful for concept mapping. Such linguistic structures can now be delivered by a chunk parser.

Chunk parsing with manually-crafted rules developed here can deliver linguistic structures

- for the mapping task, namely, assigning words or phrases to their corresponding medical concepts, and
- as input for further linguistic processing, e.g., deeper syntactic parsing.

The advantage of using a chunk parser is that it can be quickly constructed (and adapted), simply by using the POS information of the words without any further information like lexical selection criteria or morphology (Abney, 1996; Müller, 2007).

The chunk parser of the system utilizes the Stuttgart-Tübinger-Tagset (STTS) tagset and a standard probabilistic tagger to generate the POS tags. It also adds POS tags for *Latin* to the list of the STTS tags, since German medical texts use a lot of Latin phrases, which *differ* from *German* word order. The most prominent example is the post-modifier word order between adjectives and nouns, e.g., the Latin modifying adjectives follow the modified noun. This is the reason why we can neither use a generic chunker for German, nor an annotated corpus for German newspaper as training data.

The sentence below is an example of a chunked medical text, in this case, a diagnosis of a cerebral infarction. The example amplifies the importance of chunking for the matching of medical concepts. In case we have concepts like *Arteria cerebri posterior*, *Arteria cerebri media*, and *Arteria cerebri anterior*, it is important to understand that the words in the phrase *der Arteria cerebri posterior, media und anterior beidseits* (*ischaemic cerebral infarction in the supply region of the posterior, middle and anterior cerebral arteries on both sides*) belong together, meaning that *posterior, media und anterior* are all related to *Arteria cerebri*, and that the attribute *beidseits* (on both sides) is related to all of them.

(1) *[Ischämische Hirninfarkte]$_{np}$ [im Versorgungsgebiet]$_{pp}$ [der Arteria cerebri posterior, media und anterior beidseits]$_{np}$*

2.3. Repair Rules

Within the proposed hybrid parsing architecture, we envisage a layer in which *repair rules* are applied just **after** the chunking stage and **before** deeper PCFG parsing takes place. Our idea is motivated by wrongly-assembled chunks that we have found in the output of the chunker. For instance, the sentence *KM affiner SD-Knoten rechts basal* is bracketed and labelled

(2) *[KM]$_{np}$ [affiner SD-Knoten rechts basal]$_{np}$*

However, what we would like to see is

(3) *[KM affiner]$_{ap}$ [SD-Knoten rechts basal]$_{np}$*

or

(4) *[KM affiner SD-Knoten rechts basal]$_{np}$*

or even better a correction that adds a hyphen between *KM* and *affiner* (its absence being the reason why chunking went wrong):

(5) *[KM-affiner SD-Knoten rechts basal]$_{np}$*

Such a behavior can be implemented through repair rules after chunking whose application is guided by a trained error model and triggered by lexical items or even domain-semantic/ontological classes.

Such rules are either *monotonic*, meaning that they add a further interpretation to wrongly-assembled chunks, or *non-monotonic* in that they act as rewrite rules by partly "destroying" the output from an earlier stage of the processing cascade. Given the PCFG models described in the next section, we would opt for the first "enriching" approach, especially since contradictory results are still kept in the PCFG model, and lower ranked analyses can even be requested from post-PCFG stages.

Repairing after (and not before or during) chunking has several advantages. Firstly, the chunk grammar can be kept restrictive and need not be changed. Secondly, the potentially wrong analyses are still available for further processing. Finally, post-chunking "repair" rules can be employed to assemble partial intra-sentence analyses (see (Kasper et al., 1999)).

2.4. Full PCFG Parsing

While the chunking output already includes the partial grouping of words into larger constituents, to fully understand the attachment relations between chunks, one needs to employ a full-fledged grammar. Unlike traditional parsing which operates directly on word units, the grammar needs to also respect the output from the chunker. The result of grammatical analysis is a fully syntactic constituent tree that covers all the words in the input utterance. The nodes in the tree encodes both the syntactic category of the constituent and the grammatical function between the head and its dependents.

As a concrete example, let us consider the following sentence from a patient record:

> *Wir empfehlen die schmerz- und befundadaptierte Belastungssteigerung innerhalb der nächsten Wochen. (we suggest the pain- and finding-adapted increase of load within the next weeks.)*

After POS tagging and chunking, the above sentence receives the following annotation:

[Wir/PPER]$_{np}$ [empfehlen/VVFIN]$_{vp}$ die/ART [schmerz-/NN und/KON befundadaptierte/ADJA]$_{ap}$ [Belastungssteigerung/NN]$_{np}$ [innerhalb/APPR [der/ART nächsten/ADJA Wochen/NN]$_{np}$]$_{pp}$ ./$.

As we see, the chunking output has already identified the basic adjective and nominal phrases and the boundary of the prepositional phrase. But the attachment between the chunks are left underspecified. When applying the PCFG, we reach an annotation as shown in Figure 1.
The POS tags are rendered in *blue*, and the chunking categories are shown in *red*. All nodes in *rectangles* correspond to the non-terminal symbols in the PCFG. Clearly, such additional structure offers more syntactic information, as it specifies attachments, categories, and types of dependencies for the given input.

Now, to achieve such syntactic analyses, one can adopt a cascaded architecture to integrate PCFG parsing with POS tagging and chunking results. It is worth noting that typical PCFG parsing accepts as input sequences of words and their POS tags. Here, however, we need to also take into account the chunking hypotheses. More specifically, we need to map the chunk types onto the possible PCFG categories.

In order to establish such a mapping, it is necessary to investigate the definition of constituent categories in both the chunking outputs and the PCFG. By definition, a chunking result is a non-self-recursive group of consecutive words, typed by its major syntactic category. It is not always complete in the sense that it might only contain a (central) part of a fully saturated phrase in the linguistic sense. Therefore, when one maps the chunk categories into their equivalent PCFG categories, one should include both full and partial phrasal categories.

For instance, a "np" chunk not only maps to various NPs in the PCFG (e.g., NP-SB, NP-OA, CNP-OA, etc.), but also to the active/incomplete states such as NP-OA^VP|NN_ (an incomplete accusative NP governed by a VP headed with a noun and which can potentially take further arguments or modifiers).

PCFGs for parsing can be automatically obtained from an annotated corpus (i.e., treebank). The ways of doing this has been thoroughly investigated and widely reported in the literature. In case of patient record parsing, we suggest to use the unlexicalized PCFG models proposed by (Klein and Manning, 2003). The unlexicalized PCFG models with linguistically motivated annotation produce humanly interpretable generative PCFG grammars that perform robustly across domains. It is also straightforward to integrate their generative probabilistic models with pre- and post-processing modules.

As an example, the PCFG model automatically extracted from the NEGRA corpus (distributed together with the Stanford Parser) uses both vertical and horizontal markovizations to enrich the information encoded in the grammar. More information on this small experiment is reported in Section 4.1.

After the mapping is established, the PCFG parsing chart is initialized with both POS tags and chunk-related non-terminal symbols with probability 1.0. The PCFG parser then continues to complete the parsing chart with the CYK algorithm, and assigns probabilities by combining the subtree probabilities with the rule probabilities. After the chart is completed, a Viterbi-like decoding algorithm can be used to extract the n-best readings from the parsing chart.

The full PCFG parsing model described above was originally developed for newspaper texts. When applied to parsing patient records, necessary adaptation must be carried out. From the literature, the main source for cross-domain parser degradation is the change of vocabulary. Since we rely on the morphological analyzer, the POS tagger, and the chunker to deal with the lexical analysis, the unlexicalized PCFG model itself is less affected by the change of domain. On the other hand, we have found that the type of linguistic expressions in the patient records vary significantly between different sections in the document. Since document structure analysis must take place before linguistic annotation, it might be possible to choose specialized PCFG models for the analysis of the specific sections of a patient record. We have not tried this yet, however.

3. Hybrid Relation Extraction

We have developed a robust strategy for the extraction of relations between concepts that is applicable to both incomplete and complete sentences (see Section 5.). This strategy finally requires the application of two further components which are introduced below, viz., *SProUT* (Section 3.1.) and *DARE* (Section 3.2.).

3.1. Relation Extraction from Local Text Units

We apply the *SProUT* system developed by DFKI's LT Lab for both recognizing named entities (e.g., person names, organizations, locations, numbers, measure units, date and time) and for extracting relation instances from *local* textual parts.
SProUT (Shallow Processing with Unification and Typed Feature Structures) is a platform for the development of multilingual shallow text processing and IE systems (Becker et al., 2002; Drożdżyński et al., 2004; Krieger et al., 2004). The reusable core components of *SProUT* are a finite-state machine toolkit, a regular compiler, a finite-state machine interpreter, a typed feature structure package, and a set of linguistic processing resources. The advantages of the *SProUT* system are that

- it allows a flexible integration of different processing modules in a cascaded system pipeline, such as tokenization, morphological analysis, named entity recognition and phrase recognition;
- it combines regular expression matching with typed feature structures to achieve efficiency and expressiveness.

SProUT is able to extract arguments of relations or events occurring close to each other in the text mentions. This is both suitable for incomplete and complete sentences. The following example rule extracts a relation containing three arguments, viz., body part, symptom, and a time duration:

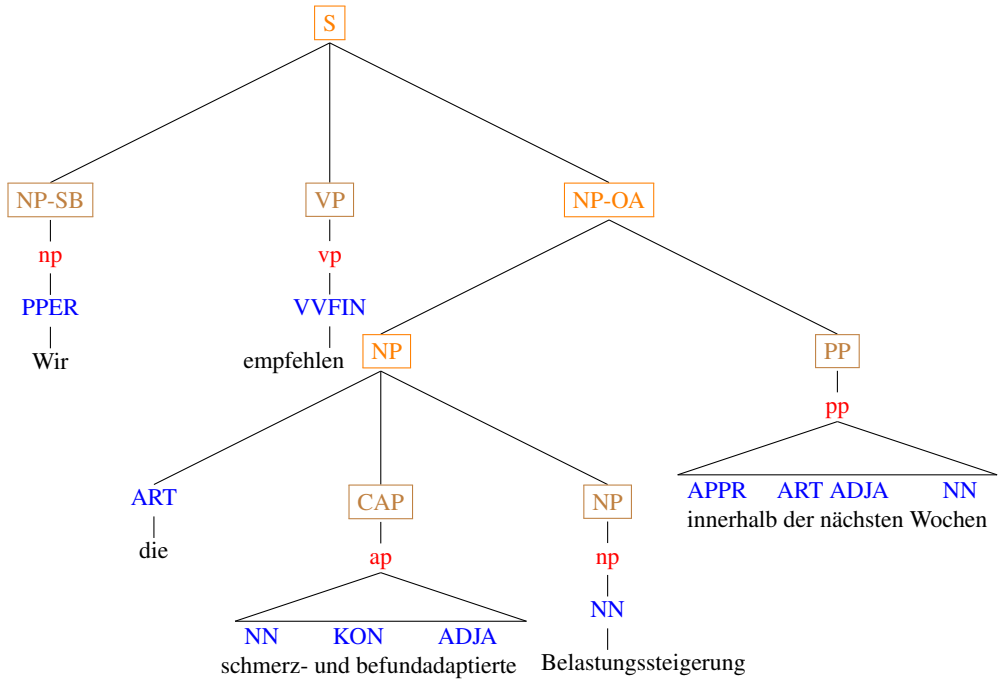

Figure 1: An example of a full phrase structure parse tree, based on the POS tagging and chunking results.

```
bodypart_symptom_duration_relation :>
  gazetteer &
    [GTYPE gaz_bodypart, CONCEPT #id,
     CSTART #c1, CEND #c2]
  gazetteer &
    [GTYPE gaz_symptom, CONCEPT #symptom,
     CSTART #c3, CEND #c4]
  gazetteer & [GTYPE gaz_time_action]?
  gazetteer &
    [GTYPE gaz_comparison_operator]?
  @seek(en_time) & #time
-> t_relation &
    [ARG1 body_part &
       [BODYPART #id, CSTART #c1,CEND #c2],
     ARG2 symptom &
       [CLASSIFY #symptom,
        CSTART #c3, CEND #c4],
     ARG3 #time].
```

This rule can extract semantic arguments from a local textual fragment, such as the following noun phrase:

(6) **chest pain** *lasting more than* **30 minutes**

Symbols starting with # express coreference relationships among arguments. `gaz_symptom` and `gaz_bodypart` refer to elements from the gazetteer list for symptoms and body parts. *SProUT* allows users to add different gazetteer lists to the grammars. All gazetteer types are subtypes of the predefined *SProUT* type `gtype`. Entries in the gazetteer list look like the following:

```
pain | GTYPE:gaz_symptom |
       CONCEPT:pain | LANG:en
slow | GTYPE:gaz_symptom |
       CONCEPT:heart_beat | LANG:en
```

The words *pain* and *slow* will be recognized as being of the type `gaz_symptom` and have corresponding semantic concepts *pain* and *heart_beat*. The gazetteer approach in *SProUT* facilitates the definition of multilingual variants for the same semantic concepts.

3.2. Relation Extraction from Complete Sentences

DARE (Xu, 2007; Xu et al., 2007) is a minimally-supervised machine learning system for relation extraction from free text, consisting of two parts: (i) a rule learning and (ii) a relation extraction (RE) stage, feeding each other in a bootstrapping framework, starting from so-called "semantic seeds", small sets of instances of the target relation. The rules are extracted from sentences which contain the seeds and which are annotated with semantic entity types and parsing results (e.g., dependency structures or annotated parse trees from a PCFG; see Section 5.). RE applies acquired rules to a text in order to discover more relation instances, which in turn are employed as seeds for further iterations. The entire bootstrapping stops when no further rules or instances can be derived. Relying entirely on semantic seeds as domain knowledge, DARE can accommodate new relation types and domains with a relatively minimal effort.

We have conducted first experiments with DARE for extracting relation instances from medical reports (a preliminary evaluation is described in Section 4.2.). The following relation types were considered:

- *symptom–body-part*
- *disease–body-part*

A semantic seed for the symptom–body-part relation in German is, e.g.,:

(7) *Zyanose–Haut, Schleimhaut*

Example seeds for the disease–body-part relation in German are:

(8) *Ischämischer Schlaganfall–Gehirn*

(9) *Kolorektales Karzinom–Blinddarm, Mastdarm, Dickdarm, Colon*

(10) *Siegelringkarzinom–Drüse, Exokrine Drüse, Magen, Schleimhaut*

The following sentence mentions an instance of the relation between a symptom and a body-part.

(11) *Einerseits eine Fehlsteuerung des lokalen Nervensystems aufgrund zurückliegender traumatischer Ereignisse sowie eine psychogene chronische* **Verspannung** *der Muskulatur des* **Beckenboden***s.*

Figure 3 shows a learned DARE rule from the parse tree in Figure 2.

4. Evaluation

What follows is a short preliminary evaluation of the parsing pipeline and the relation extraction stage in isolation.

4.1. Parsing Performance

The unlexicalized probabilistic context free grammar extracted from the NEGRA corpus (see Section 2.4.) uses both vertical and horizontal markovizations (Klein and Manning, 2003) to enrich the information encoded in the grammar. The obtained grammar contains a total of 107 different preterminal tags, and 7,782 non-terminal categories. With a total of nearly 100K lexical entries, 1.3K unary rules, and 34.6K binary rules, the PCFG achieves high parsing coverage of more than 91% when applied to the finding and diagnosis sections of 19 German patient records. Due to the fact that the PCFG takes POS tags and chunks as input and despite the fact that it was trained on the NEGRA newspaper corpus, we obtain a attachment accuracy (which coincides with precision here) of about 73%.

4.2. Relation Instance Extraction

We were able to extract 1,699 relation instances for the above two relation types *symptom–body-part* and *disease–body-part*. About 600 of these relation instances were manually checked for correctness. This lead to a precision of about 83%.

A core problem which prevents an even better precision is due to the named entity recognizer, as it often annotates NE occurrences of certain concepts with hyponyms of the actual concepts. For example, the more specific concept *Leistenbruch* (*inguinal hernia*) is used for the occurrence *Hernie* (*hernia*). Similarly, *Zahnfehlstellung* (*malocclusion*) is annotated for *Fehlstellung* (*deformity*), and so on. This leads to wrongly-recognized relation instances in the result; more precisely, to relation instances which might be valid for the more general concept but not for the more specific one. For example, the phrase *Fehlstellung des Fußes* (*deformity of the foot*) leads to the extracted relation instance *Zahnfehlstellung–Fuß* (*malocclusion–foot*).

5. Combined Architecture

Our presentation so far and the preliminary evaluation directly above has focused on two isolated subparts of a system for extracting relation instances from medical findings and diagnoses:

1. a parsing pipeline consisting of (i) a morphological analyzer, (ii) a robust chunker, (iii) a repair rule stage, and (iv) a PCFG parser;

2. a relation extraction component that was evaluated on the output of a dependency parser.

Neither have we combined these two stages, nor have we interfaced *SProUT* (as presented in Section 3.1.) with subsystem therein so far.

The reason for this is related to the output of the PCFG parser, viz., parse trees *without* any semantic information. In order to enrich these parse trees with ontological categories, we would like to feed *SProUT* with the highly-safe phrase islands, predicted by the PCFG parser and let *SProUT* annotate these structures. As we have seen, the semantic categories are injected into *SProUT* through gazetteer entries (see Section 3.1.), and we envisage to automatically generate them from ontological resources.

Given the semantically-annotated PCFG parse trees and a set of semantic seeds, *DARE* then is responsible for generating relation extraction rules that we can ultimately use to find relation instances in new documents.

6. Conclusion

In this paper, we have presented a hybrid strategy both for parsing and relation extraction, dealing with patient record texts which contain both complete and incomplete sentences. Robust chunk parsing can cover almost all textual input. However, it is important to integrate repair rules to correct wrong and eager decisions made by a chunker, so that new results can be utilized later during deeper PCFG parsing. Our preliminary experiments have shown that the hybrid parsing strategy can ensure on the one hand robustness and coverage, and on the other hand the extraction of richly-structured linguistic information. In addition, the hybrid relation extraction strategy is a useful solution for obtaining relation mentions from textual fragments and complete sentences and for storing them in a medical information system for later search (e.g., to obtain patient cohorts for clinical studies).

7. Acknowledgement

The research reported here has been partially funded by the *Berliner Forschungsplattform Gesundheit* (BFG), a project funded by the European Regional Development Fund (ERDF) and the state of Berlin for building medical information systems; by the research project *Deependance* (funded by the German Federal Ministry of Education and Research, BMBF, contract no. 01IW11003) in the area of parsing and information extraction; and by the project *MEDIXIN* (funded by the German Federal Ministry for Economics and Technology, BMWi, contract no. KF2013012KM1) in the area of information extraction.

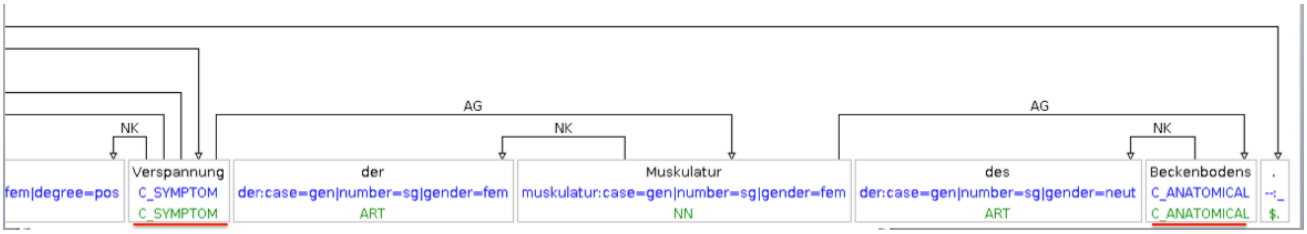

Figure 2: A dependency parsing analysis.

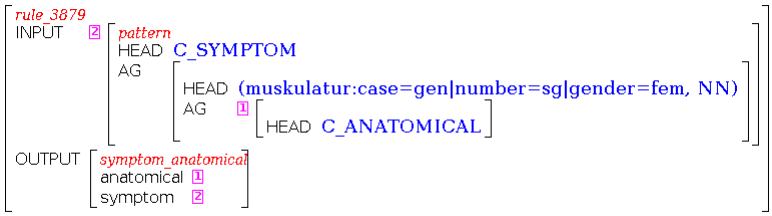

Figure 3: A relation extraction rule for the symptom–body-part relation.

The authors would like to thank our reviewers for their detailed comments.

8. References

Abney, Steven. (1996). Partial parsing via finite-state cascades. *Natural Language Engineering*, 2(4):337–344.

Becker, Markus, Drożdżyński, Witold, Krieger, Hans-Ulrich, Piskorski, Jakub, Schäfer, Ulrich, and Xu, Feiyu. (2002). SProUT—Shallow Processing with Unification and Typed Feature Structures. In *Proceedings of the International Conference on Natural Language Processing, ICON-2002*.

Drożdżyński, Witold, Krieger, Hans-Ulrich, Piskorski, Jakub, Schäfer, Ulrich, and Xu, Feiyu. (2004). Shallow Processing with Unification and Typed Feature Structures—Foundations and Applications. *KI*, 04(1):17–23.

Geibel, Peter, Trautwein, Martin, Erdur, Hebun, Zimmermann, Lothar, Krüger, Stefan, Schepers, Josef, Jegzentis, Kati, Müller, Frank, Nolte, Christian Hans, Becker, Anne, Frick, Markus, Setz, Jochen, Scheitz, Jan Friedrich, Tütüncü, Serdar, Usnich, Tatiana, Holzgreve, Alfred, Schaaf, Thorsten, and Tolxdorff, Thomas. (2013). Ontology-based semantic annotation of documents in the context of patient identification for clinical trials. In *OTM Conferences*, pages 719–736.

Goodwin, Travis and Harabagiu, Sanda M. (2013). The impact of belief values on the identification of patient cohorts. In *CLEF*, pages 155–166.

Kasper, Walter, Kiefer, Bernd, Krieger, Hans-Ulrich, Rupp, C.J. and Worm, Karsten L.˙ (1999). Charting the depths of robust speech parsing. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 405–412.

Klein, Dan and Manning, Christopher D.˙ (2003). Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 423–430, Sapporo, Japan. Association for Computational Linguistics.

Krieger, Hans-Ulrich, Drożdżyński, Witold, Piskorski, Jakub, Schäfer, Ulrich, and Xu, Feiyu. (2004). A Bag of Useful Techniques for Unification-Based Finite-State Transducers. In *Proceedings of KONVENS 2004*, pages 105–112.

Müller, Frank Henrik. (2007). *A Finite-State Approach to Shallow Parsing and Grammatical Functions Annotation of German*. Phd thesis, Tübingen University.

Roberts, Kirk, Rink, Bryan, and Harabagiu, Sanda M. (2013). A flexible framework for recognizing events, temporal expressions, and temporal relations in clinical text. *JAMIA*, 20(5):867–875.

Xu, Feiyu, Uszkoreit, Hans, and Li, Hong. (2007). A seed-driven bottom-up machine learning framework for extracting relations of various complexity. In *Proceedings of ACL 2007, 45th Annual Meeting of the Association for Computational Linguistics*, Prague, Czech Republic, 6.

Xu, Feiyu. (2007). *Bootstrapping Relation Extraction from Semantic Seeds*. Phd thesis, Saarland University.