Semantic Representations

Yannick Versley Heike Zinsmeister

University of Tübingen
Seminar für Sprachwissenschaft / SFB 441
E-mail: {versley,zinsmeis}@sfs.uni-tuebingen.de

1 Introduction

In the past, a divide could be seen between 'deep' parsers on the one hand, which construct a semantic representation out of their input, but usually have significant coverage problems, and more robust parsers on the other hand, which are usually based on a (statistical) model derived from a treebank and have larger coverage, but leave the problem of semantic interpretation to the user.

More recently, approaches have emerged that combine the robustness of data-driven (statistical) models with more detailed linguistic interpretation such that the output could be used for deeper semantic analysis. Cahill et al. (2002) use a PCFG-based parsing model in combination with a set of principles and heuristics to derive functional (f-)structures of Lexical-Functional Grammar (LFG). They show that the derived functional structures have a better quality than those generated by a parser based on a state-of-the-art hand-crafted LFG grammar.

Advocates of Dependency Grammar usually point out that dependencies already are a semantically meaningful representation (cf. Menzel, 2003). However, parsers based on dependency grammar normally create underspecified representations with respect to certain phenomena such as coordination, apposition and control structures. In these areas they are too 'shallow' to be directly used for semantic interpretation.

In this paper, we adopt a similar approach to Cahill et al. (2002) using a dependency-based analysis to derive functional structure, and demonstrate the feasibility of this approach using German data. A major focus of our discussion is on the treatment of coordination and other potentially underspecified structures of the dependency data input.

F-structure is one of the two core levels of syntactic representation in LFG[1] (Bresnan, 2001). Independently of surface order, it encodes abstract syntactic functions that constitute predicate argument structure and other dependency relations such as subject, predicate, adjunct, but also further semantic information such as

[1]The second core level is constituent (c-)structure which encodes language-specific word order and constituent hierarchies. It is often represented as a context-free phrase structure tree.

Hajič, J. and Nivre, J. (eds.): Proceedings of the TLT 2006, pp. 115–126.

the semantic type of an adjunct (e.g. directional). Normally f-structure is captured as a recursive attribute value matrix, which is isomorphic to a directed graph representation. Figure 5 depicts an example target f-structure.

As mentioned earlier, these deeper-level dependency relations can be used to construct logical forms as in the approaches of van Genabith and Crouch (1996), who construct underspecified discourse representations (UDRSs), and Spreyer and Frank (2005), who have robust minimal recursion semantics (RMRS) as their target representation. Frank and Semecký (2004) use f-structures for semantic role labeling. We therefore think that f-structures are a suitable target representation for automatic syntactic analysis in a larger pipeline of mapping text to interpretation.

In this paper, we report on the conversion from dependency structures to f-structure. Firstly, we evaluate the f-structure conversion in isolation, starting from hand-corrected dependencies based on the TüBa-D/Z treebank and Versley (2005)'s conversion. Secondly, we start from tokenized text to evaluate the combined process of automatic parsing (using Foth and Menzel (2006)'s parser) and f-structure conversion. As a test set, we randomly selected 100 sentences from TüBa-D/Z which we annotated using a scheme very close to that of the TiGer Dependency Bank (Forst et al., 2004)[2].

In the next section, we sketch dependency analysis, the underlying theory of our input representations, and introduce four different representations of coordination. We also describe Weighted Constraint Dependency Grammar (WCDG), the dependency parsing formalism that we use in our experiments. Section 3 characterises the conversion of dependencies to f-structures. Our evaluation is presented in section 4, and finally, section 5 summarises our results and gives an overview of problems remaining to be solved.

2 Dependencies as Input Representation

Existing frameworks for dependency parsing assume that the dependency structure is a tree-shaped directed graph. The words of the sentence function as nodes, and possibly a special root node is added. The nodes are related by labeled edges that encode the functional information of the clause. In the simplest case, the dependency and functional structures are isomorphic. Each node has at most one governor, in other words there is at most one edge that relates a dependent node to its immediate governor ('single-parent assumption').

2.1 Coordination

Not all phenomena can be captured by the single parent assumption. Coordination is a major example, since redundant material is often left unexpressed, such that a single word in the string may be related to more than one governing conjunct.

The frameworks for modern dependency parsing usually linearise the conjuncts and have the first conjunct stand for the whole coordination, so that each conjunct still has only one parent (see figure 1). This is in line with Mel'čuk (1988)'s version

[2]Martin Forst kindly provided us with a copy of the TiGer Dependency Bank, but due to the problems mentioned in section 4, we decided to create our own test set, while remaining close to the general annotation scheme of the TiGer Dependency Bank.

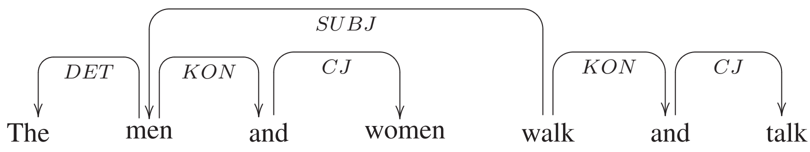


Figure 1: Single-parent (Mel'čuk-style) dependencies

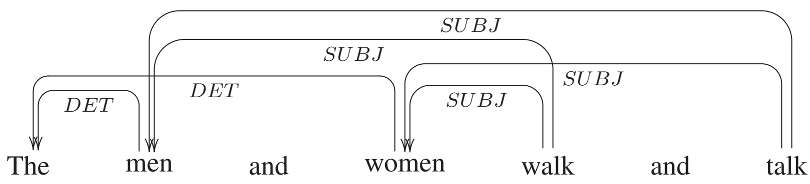


Figure 2: Tesnière-style dependencies

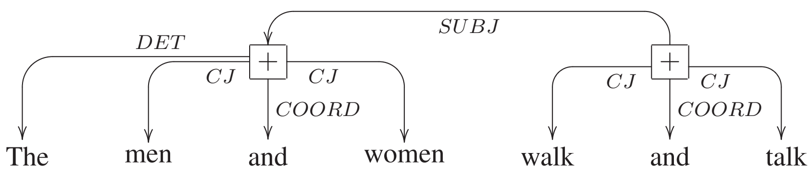


Figure 3: intermediate representation

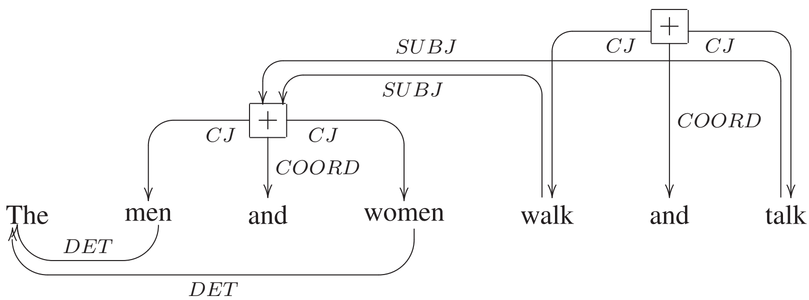


Figure 4: representation used in the TiGer Dependency Bank

of Dependency Grammar[3], and is to be seen in contrast to Tesnière (1959), who analyses coordination differently: as shown in Figure 2, a word can have multiple parents, e.g. the subject of this sentence has two governors.

Figure 3 depicts the intermediate representation in our algorithm for enriching underspecified input structures. In this hybrid dependency/phrase structure, a coordination is represented as an additional node, which modifies its syntactic parent and has as dependents both its conjuncts and the elements that modify the whole coordination (distributed to all conjuncts). The single-parent output of state-of-the-art dependency parsers underspecifies certain relationships, usually between a dependent modifying either its parent or the coordination that the latter belongs to. Because of this, it is necessary to use certain heuristics to get a fully specified structure.

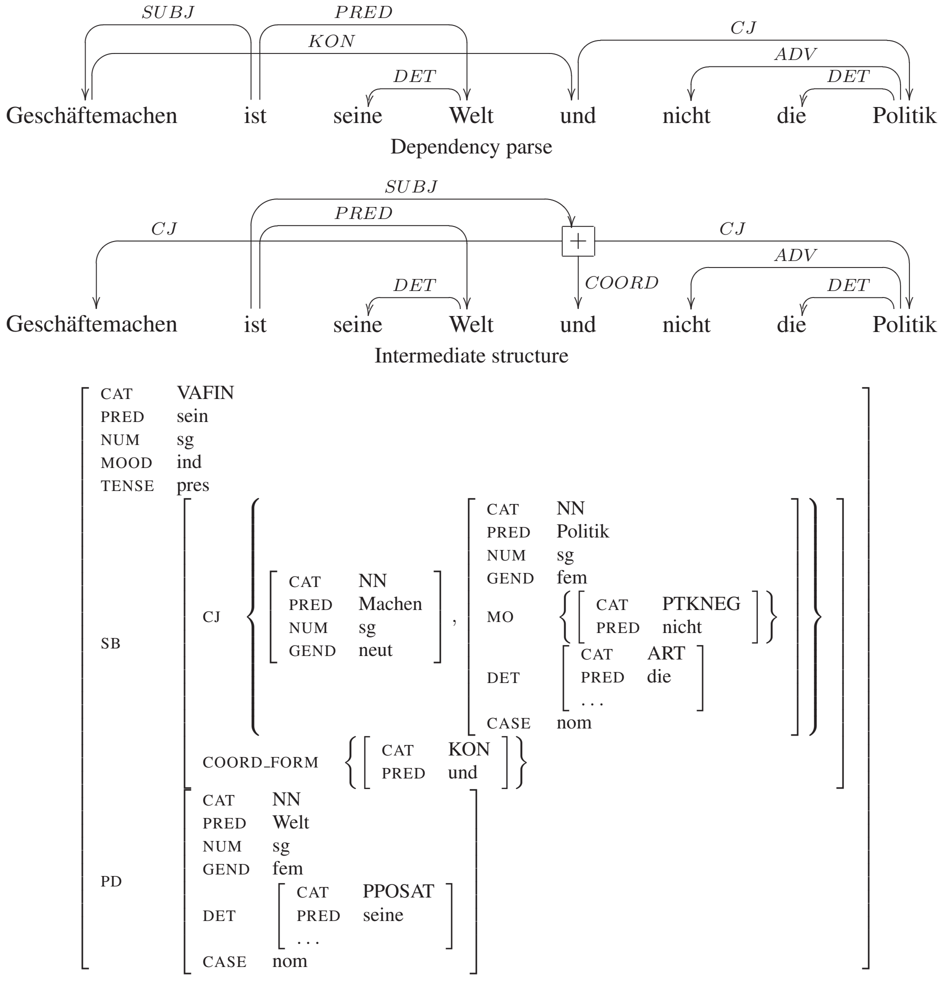
Our target representation, depicted in figure 4, combines intermediate structure nodes for coordination with a multi-parent analysis by relating a dependent to all its governors explicitly. This is also how coordination is analysed in the TiGer Dependency Bank.

2.2 The Weighted Constraint Dependency Grammar formalism

Our experiments are based on dependency structures conforming to the Weighted Constraints Dependency Grammar (WCDG) formalism. This formalism is a lexicalised dependency grammar formalism where grammatical constraints are checked locally on adjacent edges (except for some constraints that examine a whole subtree) and they can be weighted. Because the formalism does not create or aggregate new information by unification or structure-building, feature checking is approximated by local constraints. For example, number agreement is not only checked between determiner and noun and between subject and verb, but also between the verb and the subject's determiner, to cover the case when the noun's lexical entry is underspecified for number. This local feature checking serves in part to make the parsing problem more tractable, but also allows for increased robustness in cases where parts of the surface information are in conflict with each other.

The parsing process itself consists in choosing an optimal assignment of lexical entries and dependencies with respect to the cumulated weights of violated constraints in the grammar. Since this problem is generally not tractable, a heuristic search method (taboo search, cf. (Glover, 1986)) is used. In order to increase parsing efficiency for long sentences, an initial analysis is first sought for each part of the sentence (roughly corresponding to single (sub)clauses) before the next stage considers all variables at once (Foth and Menzel, 2003).

In a hybrid parsing framework, input from statistical components, namely a part-of-speech tagger, a PP attacher and a shift-reduce parser is integrated (Foth and Menzel, 2006).



F-structure

Figure 5: Conversion from dependencies to f-structure
(Example sentence from (Cahill et al., 2005))

3 Converting to F-structures

The input from the WCDG parser, consisting of dependencies as well as the selected lexicon entries (containing features such as number and gender for nouns, or tense for verbs), is transformed into a representation with (nonprojective) phrase structures, and coordinations are identified along with their conjuncts, yielding an additional phrase for each coordination. In the next step, a complete f-structure is created by mapping dependency labels to f-structure equations and unifying these. This f-structure contains all the necessary information except for some representational differences between WCDG and the target representation. In a final step, graph rewriting is used to transform the graph from the f-structure into a representation that is maximally similar to that of the TiGer Dependency Bank.

3.1 Identifying conjunctions

In the WCDG grammar, coordinations as well as appositions and multi-part names (the latter two are both expressed as APP dependencies) are linearized into a left-headed chain. For each node in the dependency graph, we look for KON or CJ edges[4] that are connected to this node via zero or more APP edges, remove these from their parents and make conjuncts out of these (cf. figure 5).

Since the single-parent dependencies output by the WCDG parser are underspecified with respect to scope, we have to fully specify these using some heuristics and modify the intermediate representation accordingly.

This is not always straightforward, as the following examples show:[5]

(1) a. Peter $\stackrel{\text{APP}}{\rightarrow}$ Müller $\stackrel{\text{KON}}{\rightarrow}$ und $\stackrel{\text{CJ}}{\rightarrow}$ [seine Frau]

b. [die Altaktionäre], $\stackrel{\text{APP}}{\rightarrow}$ [der Niederländer von Zadelhoff] $\stackrel{\text{KON}}{\rightarrow}$ und $\stackrel{\text{CJ}}{\rightarrow}$ [der Brite John Morgan]

c. [15 Jahre] $\stackrel{\text{APP}}{\rightarrow}$ Aufenthalt $\stackrel{\text{KON}}{\rightarrow}$ und $\stackrel{\text{CJ}}{\rightarrow}$ [der Verzicht auf die alte Staatsbürgerschaft]

(For clarity, dependency structure inside brackets is omitted).

All three have an APP-KON-CJ chain, but the implied structure is different: In examples (a) and (c), the coordination encompasses the whole noun phrase, but in example (b), it is only the part after the apposition. Examples (a) and (b) can be distinguished on the ground that "die Altaktionäre $\stackrel{\text{APP}}{\rightarrow}$ der Niederländer von Zadelhoff" would not make sense as one conjunct since there is disagreement in number. But the structure for the measure phrase "15 Jahre $\stackrel{\text{APP}}{\rightarrow}$ Aufenthalt" also disagrees in number, but needs to be represented as a single conjunct. Although

[3] Mel'čuk, as well as the Prague Dependency Treebenk, assume additional representation layers where elements can be duplicated.

[4] CJ indicates the conjunct that follows the conjunction whereas KON relates all other elements of the coordination.

[5] Translations: Peter $\rightarrow_{APP}$ Müller $\rightarrow_{KON}$ and $\rightarrow_{CJ}$ his wife;
The existing shareholders, $\rightarrow_{KON}$ the Dutch von Z. $\rightarrow_{KON}$ and $\rightarrow_{CJ}$ the Englishman John M.;
15 years *(of)* $\rightarrow_{APP}$ residence $\rightarrow_{KON}$ and $\rightarrow_{CJ}$ relinquishing the old citizenship

there is no overt marking of measure phrases, the WCDG grammar specifically allows these cases of number disagreement, using lexical information including set-like words (*a handful*, *a bag (of)*), time units, measure and currency units (*Hertz*, *Inch* or *Dollar*) and percentages.

3.2 The scope of modifiers

A modifier of a conjunct is eligible for wide scope only if its position is compatible with wide scope and the modifier structure of the other conjuncts is compatible with the modifier having scope over them.

For coordinated nouns, the left dependents of the first conjunct and the right dependents of the last conjunct are positionally compatible with wide scope.

For coordinated full verbs, the subject is positionally compatible with wide scope even when it is *not* a left dependent. In subject-gap-fronted constructions (Höhle, 1983), the subject, as a right dependent of the first conjunct, is shared.

The WCDG grammar covers these constructions by requiring a finite verb to either have a dependent with a subject edge (SUBJ or SUBJC) or be part of a coordination.

(2) Zwar ist [$_{\text{sb}}$ er] nicht [$_{\text{pd}}$ so populär wie Mandela], führt als
Though is he not as popular as Mandela, leads as
Vizepräsident aber praktisch schon längst [$_{\text{oa}}$ die Regierungsgeschäfte].
vice-president but practically already the government affairs.
"He may not be as popular as Mandela, but as the vice-president, he already conducts the governmental affairs."

In coordinations of nonfinite verbs, topicalized arguments or adjuncts are shared and obey the across-the-board principle, but non-topicalized arguments can be shared as well:

(3) [$_{\text{da}}$ Einem Praktikanten] haben [$_{\text{sb}}$ Kollegen] [$_{\text{oa}}$ seine ungespülte Tasse]
one intern have colleagues his unwashed cup
mit einem Kaktus bepflanzt und an der Decke aufgehängt
with a cactus planted and at the ceiling hung
"One intern had his unwashed cup planted with a cactus and hung from the ceiling by his colleagues."

3.3 From labeled edges to f-structure

In the resulting intermediate structure, the dependency labels are to be mapped to f-structure equations. In most cases, the mapping is functional. For example, the dependency label SUBJ is always mapped to the f-structure equation $\uparrow\text{SUBJ} = \downarrow$. The mapping is not bijective, since TiGer-DB, like the TiGer treebank, uses, for example, the label MO for all adjunct relations, whereas the WCDG dependencies have different labels for attributive adjectives, adverbs, or prepositional phrases. In other cases, the labels in the WCDG grammar are ambiguous between several interpretations which are distinct from each other in TiGer DB. For example, the GMOD edge label is used both for premodifying genitives ("Martins Auto" / Martin's car) and for postmodifying genitives ("das Kochen der Kartoffeln" / the cooking of the potatoes) which are labeled GL and GR respectively in TiGer-DB. This distinction is useful, since premodifying genitives usually also indicate definiteness and thus having a similar function to definitive articles.

The WCDG AUX label links auxiliaries and modals to the main verb. To get a very rough interpretation, one could underspecify the relation and just use the equations $\downarrow = \uparrow\text{AUX}\ ;\ \downarrow\text{SUBJ} = \uparrow\text{SUBJ}$. In our conversion, however, we distinguish modals from auxiliaries and we also reconstruct passives and tenses (future as well as perfect), which are formed analytically with a nonfinite verb plus an auxiliary and are thus only implicitly encoded in the AUX edges of the dependency representation. For example, *werden* as auxiliary together with an infinitive indicates future tense (analogous to the English *will*), whereas *werden* with a past participle indicates passive in German. The conversion program determines f-structure equations for AUX labels by using part-of-speech information and the lexical entries of the verbs, both of the governed verb as well as of the modal/auxiliary:

if is_aux_verb($Head$)
  $Eqns_0 = \{\downarrow = \uparrow\text{AUX}, \downarrow\text{SUBJ} = \uparrow\text{SUBJ}, \downarrow\text{MOOD} = \uparrow\text{MOOD}\}$
  if $Head.lemma =$ werden
    if is_past_participle($Head$)
      $Eqns = Eqns_0 \cup \{\downarrow\text{TENSE} = \uparrow\text{TENSE}, \downarrow\text{PASSIVE} = \text{dynamic}\}$
    else if is_infinitive($Head$)
      $Eqns = Eqns_0 \cup \{\downarrow\text{TENSE} = \text{fut}, \downarrow\text{PASSIVE} = \uparrow\text{PASSIVE}\}$
    ...
  else if $Head.lemma =$ sein
    ...

3.4 Graph rewriting

In some cases, the WCDG grammar posits a different head than our target f-structure, for example, in sentence subordination, the TiGer-DB representation has the complementiser as the head of the subclause, while the WCDG dependencies have the complementiser as a dependent of the finite verb of the subclause. This problem can be easily solved by rewriting the dependency triples by means of transfer rules like the following:

$$\text{mo}(A, B), \text{konj}(B, C) \Rightarrow \text{mo}(A, C), \text{obj}(C, B)$$

TiGer-DB collapses sequences of auxiliaries, while our conversion yields an f-structure similar to earlier LFG representations, with additional nodes for auxiliaries. To get a comparable representation we also used rewriting rules to 'flatten' the verbal dependencies.

An area where the representation of the TiGer-DB and the WCDG framework differ most visibly is the treatment of names and name parts. In our conversion, multi-token names are chained up with NAME edges, and titles (i.e. determinerless common nouns that precede a name, such as Mr., Professor, President) are attached with a TITLE edge. For example, in the noun phrase "Staatspräsident Emil Constantinescu", the head "Emil" would have the common noun "Staatspräsident" as a left-dependent TITLE, and “Constantinescu” as a right-dependent NAME. Name parts that serve to further specify or classify a named entity carry the label CFY, as in “die Staatsanwaltschaft Karlsruhe”, where Karlsruhe (the city) specifies its head “Staatsanwaltschaft” (prosecutor’s office). Cases where a full NP precedes the named entity it describes, as in “das Nervengift Curare”, are headed by the describing noun and the named entity is attached with the APP label.

4 Evaluation and Discussion

To assess the quality of the f-structure conversion, we first evaluate it in isolation, starting from hand-corrected dependencies based on the TüBa-D/Z treebank and Versley (2005)’s conversion. In a second evaluation, we start from tokenized text to evaluate the combined process of automatic parsing (using Foth and Menzel (2006)’s parser) and f-structure conversion. Our test set consists of 100 randomly selected sentences from TüBa-D/Z which we annotated using a scheme very close to that of the TiGer Dependency Bank. By using parses from the WCDG parser (or those from a statistical dependency parser such as Nivre and Nilsson (2005); McDonald (2006)’s, together with appropriate lexical entries) instead of the hand-corrected trees, it is possible to perform a meaningful parser comparison across frameworks as different as LFG, HPSG and Dependency Grammar.

In a preliminary experiment, we first considered using dependencies automatically converted by Daum et al. (2004)’s DepSy conversion tool, and evaluated it against the gold standard f-structures from TiGer-DB (Forst et al., 2004). However, a meaningful comparison was hindered by a number of issues:

- Differences in the tokenization can lead to different IDs for the same word in each version, which is a serious problem for zu-infinitives, where the particle ‘zu’ is not counted as a token in TiGer-DB.

- Certain grammatical phenomena are modeled in TiGer-DB by introducing additional nodes. Matching these nodes in the conversion and TiGer-DB turned out not to be straightforward tue to a rather complex numbering scheme for these nodes in TiGer-DB.

- Predicative arguments of non-copular predicative constructions (such as “jmd. als X betrachten”, to regard somebody as X) are annotated as PD (predicative) instead of MO (modifying adjunct) and they get assigned the main clause’s subject. In this respect, the Tiger-DB gold standard goes further than is necessary for parser evaluation in the stricter sense, and while it is interesting to see how these additional deep dependencies are influenced by errors that the parsing engine makes, we need to be aware of the danger of not simply comparing one parser against the other, but the ParGram project’s excellent lexical resources against whatever else is available to the research community at large.

The results from the isolated evaluation on 100 sentences from TüBa-D/Z (see table 1) show that, while not trivial, it is indeed possible to reconstruct full information from the dependencies in most cases - the two sentences where this failed

Relation	*from gold dependencies* Precision	Recall	F-score	*from dependency parses* Precision	Recall	F-score
sb	0.98	0.93	0.95	0.88	0.83	0.85
oa	1.00	0.96	0.98	0.81	0.81	0.81
da	1.00	1.00	1.00	1.00	0.50	0.67
oc_inf	0.95	0.91	0.93	0.94	0.73	0.82
name	0.97	1.00	0.98	0.80	0.83	0.81
app	0.91	0.95	0.93	0.33	0.57	0.42

Table 1: Conversion results: selected features.

involve gapping, the analysis of which is only approximated in the WCDG framework (Foth, 2004), making the construction of our target representation virtually impossible. Since subjects are often shared in coordination, recall is somewhat below precision for subjects, something that also occurs in (Cahill et al., 2005).

If we compare the full step of parsing and converting to f-structures to the results of Forst and Rohrer (2006), who use an LFG grammar with a stochastic disambiguation component and give detailed results for all grammatical functions, our results for selected grammatical functions (again, see table 1) are quite encouraging, since they are slightly above the numbers that Forst and Rohrer report for their parser on the TiGer-DB test set. This may also be because the WCDG parser is more robust and achieves sensible results for a larger set of sentences than the (non-fragment) LFG grammar can cover. Since different phenomena can be covered with differing ease in a phrase structure-based framework or in a dependency-based framework (gapping is better handled in phrase structures, while nonprojectivity can be better handled in a depency representation), there is probably potential for a gain in accuracy when the two parsing mechanisms are intelligently combined.

To provide a more detailed picture of the strength and weaknesses of the approach of dependency conversion, it would be necessary to make a detailed evaluation on a larger set of sentences, using all grammatical relations and morphosyntactic features, which is not possible here due to time and space constraints.

5 Related Work

The construction of f-structures from phrase structures as they occur in treebanks or PCFG parses has been successfully attempted before, both for English (Cahill et al., 2002) and for German (Cahill et al., 2005), whereas the conversion of dependency parses to f-structures has not yet been attempted, at least to our knowledge.

In comparison to PCFG parsing, nonprojective dependency parsing has the advantage that nonprojective dependencies, which would otherwise need to be handled by some kind of long-distance-dependency resolution, are naturally part of the parsing model. An alternative approach to the handling of coordinations that we present here is due to McDonald (2006): for parsing dependencies like those proposed by Tesnière, he uses an existing parsing algorithm for single-parent dependencies and afterwards adds additional dependencies for the other conjuncts.

In the Danish Dependency Treebank, which is annotated using Kromann (1999)'s Discontinuous Grammar formalism and includes Tesnière-style handling of coordinations, 5% of the words that have multiple parents, and the parser's recall is improved by 2.5% when adding the additional non-single-parent edges.

Böhmová and Sgall (2000), as well as Hajičová (1998), mention an automatic conversion from surface structures to tectogrammatical ones, where additional nodes are added for zero pronouns and subject/object control, but do not tackle the problem of shared/unshared structure in coordinations.

6 Conclusion

We presented an approach to construct f-structures from syntactic dependencies such as they occur in the output of dependency parsers. In order to apply a conversion like the one we suggest here to the output of a parser other than the WCDG parser, such as a statistical dependency parser (e.g. McDonald, 2006), additional lexical resources are needed beyond the requirements for the parser itself; on a simple level, these are number distinctions (needed for regrouping of appositional/conjunctional chains), which usually presuppose a morphological analyser and/or full-form lexicon. On a deeper level, the identification of stative passives requires knowledge about which auxiliary a verb takes. The correct identification of shared/unshared verb arguments in VP conjunctions is helped by information about verb valencies, and a lexicon of set-like and unit nouns is required for adequate identification of measure phrases. Other information, such as subject/object control and verbs with predicative arguments, is usually not found in publicly available lexical resources and has to be either constructed by hand or left out.

Comparing parsers trained on different treebanks (Kübler, 2005; Maier, 2006), or using different underlying representations (dependencies vs. phrase structures) (Schiehlen, 2004; Versley, 2005) is still an unsolved problem, and, at least for German, an interesting one, since a multitude of different parsers exist which cannot be compared using the popular PARSEVAL measures. It is clear that for a comparison of Forst and Rohrer (2006)'s 77% f-measure on f-structure dependency triples and Foth and Menzel (2006)'s 91% labeled dependency accuracy, it is necessary to look beyond raw numbers, since both representations contain different information and there is no one-to-one mapping between the two. Only by arriving at a common representation is it possible to provide a meaningful comparison.

Acknowledgements This work was supported by the Deutsche Forschungsgemeinschaft as part of the collaborative research center (Sonderforschunsbereich) 441 "Linguistic Data Structures". We wish to thank Martin Forst for allowing us to use the TiGer Dependency Bank, and Piklu Gupta, as well as Erhard Hinrichs, for constructive comments on an earlier version of this paper.

References

Bresnan, J. (2001). *Lexical-Functional Syntax*. Blackwell, Oxford.

Böhmová, A. and Sgall, P. (2000). Automatic procedures in tectogrammatical tagging. In *LINC 2000*.

Cahill, A., Forst, M., McCarthy, M., O' Donovan, R., Rohrer, C., van Genabith, J., and Way, A. (2005). Treebank-based acquisition of multilingual unification grammar resources. In Bender, E., Flickinger, D., Fouvry, F., and Siegel, M., editors, *Journal of Research on Language and Computation; Special Issue on "Shared Representations in Multilingual Grammar Engineering"*. Kluwer Academic Press.

Cahill, A., McCarthy, M., van Genabith, J., and Way, A. (2002). Parsing with PCFGs and automatic f-structure annotation. In *Proceedings of the Seventh International Conference on LFG*. CSLI Publications.

Daum, M., Foth, K., and Menzel, W. (2004). Automatic transformation of phrase treebanks to dependency trees. In *LREC 2004*.

Forst, M., Bertomeu, N., Crysmann, B., Fouvry, F., Hansen-Schirra, S., and Kordoni, V. (2004). Towards a dependency-based gold standard for German parsers - the Tiger dependency bank. In *COLING Workshop on Linguistically Interpreted Corpora (LINC 2004)*.

Forst, M. and Rohrer, C. (2006). Improving coverage and parsing quality of a large-scale LFG for German. In *LREC 2006*.

Foth, K. (2004). Writing weighted constraints for large dependency grammars. In *COLING Workshop on Recent Advances in Dependency Grammar*.

Foth, K. and Menzel, W. (2003). Subtree parsing to speed up deep analysis. In *8th International Workshop on Parsing Technologies (IWPT'03)*.

Foth, K. and Menzel, W. (2006). Hybrid parsing: Using probabilistic models as predictors for a symbolic parser. In *ACL 2006*.

Frank, A. and Semecký, J. (2004). Corpus-based induction of an LFG syntax-semantics interface for frame semantic processing. In *LINC 2004*.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 13:533–549.

Hajičová, E. (1998). Prague dependency treebank: From analytic to tectogrammatical annotation. In *Proc. 1st Workshop on Text, Speech, Dialogue*.

Höhle, T. N. (1983). Subjektlücken in Koordinationen. Unpublished manuscript; available at *http://www.linguistik.uni-tuebingen.de/hoehle/manuskripte/SLF-W5.1_neu.pdf*.

Kromann, M. T. (1999). Towards discontinuous grammar. In *ESSLLI 1999 Student Session*.

Kübler, S. (2005). How do treebank annotation schemes influence parsing results? or how not to compare apples with oranges. In *Recent Advances of Natural Language Processing (RANLP-2005)*.

Maier, W. (2006). Annotation schemes and their influence on parsing results. In *COLING/ACL 2006 Student Research Workshop*.

McDonald, R. (2006). Online learning of approximate dependency parsing algorithms. In *EACL 2006*.

Mel'čuk, I. A. (1988). *Dependency Syntax: Theory and Practice*. State University of New York.

Menzel, W. (2003). Dependenzgrammatik und Semantische Netze. In Agel, V., Eichinger, L. M., Eroms, H.-W., Hellwig, P., Heringer, H. J., and Lobin, H., editors, *Dependenz und Valenz - Ein Internationales Handbuch der zeitgenössischen Forschung*. Walter de Gruyter.

Nivre, J. and Nilsson, J. (2005). Pseudo-projective dependency parsing. In *ACL 2005*.

Schiehlen, M. (2004). Annotation strategies for probabilistic parsing in German. In *Proc. 20th Int. Conference on Computational Linguistics (COLING-2004)*.

Spreyer, K. and Frank, A. (2005). Projecting RMRS from Tiger dependencies. In Müller, S., editor, *LINC 2005*, Stanford, CA. CSLI Publications.

Tesnière, L. (1959). *Éléments de syntaxe structurale*. Klincksieck, Paris.

van Genabith, J. and Crouch, D. (1996). Direct and underspecified interpretations of LFG f-structures. In *COLING'96*.

Versley, Y. (2005). Parser evaluation across text types. In *Fourth Workshop on Treebanks and Linguistic Theories (TLT 2005)*.