Towards an LFG Syntax-Semantics Interface for Frame Semantics Annotation

Anette Frank[1] and Katrin Erk[2]

[1] DFKI GmbH, Language Technology Lab, Stuhlsatzenhausweg 3,
66123 Saarbrücken, Germany,
Anette.Frank@dfki.de,
WWW home page: http://www.dfki.de/~frank

[2] Universität des Saarlandes, Computational Linguistics Department,
66123 Saarbrücken, Germany
erk@coli.uni-sb.de,
WWW home page: http://www.coli.uni-sb.de/~erk

Abstract. We present an LFG syntax-semantics interface for the semi-automatic annotation of frame semantic roles for German in the SALSA project. The architecture is intended to support a bootstrapping cycle for the acquisition of stochastic models for frame semantic role assignment, starting from manual annotations on the basis of the syntactically annotated TIGER treebank, with smooth transition to automatic syntactic analysis and (semi-)automatic semantic annotation of a much larger corpus, on top of a free-running LFG grammar of German.

Our study investigates the applicability of the LFG formalism for modeling frame semantic role annotation, and designs a flexible and extensible syntax-semantics architecture that supports the induction of stochastic models for automatic frame assignment. We propose a method familiar from example-based Machine Translation to translate between the TIGER and LFG annotation formats, thus enabling the transition from treebank annotation to large-scale corpus processing.

1 Introduction

This paper is a first study of an LFG syntax-semantics interface for frame semantic role assignment. The architecture is intended to support semi-automatic semantic annotation for German in SALSA – the Saarbrücken Semantics Annotation and Analysis project[3] – which is based on Frame Semantics and is conducted in cooperation with the Berkeley FrameNet project [1, 15].

The aim of SALSA is to create a large lexical semantics resource for German based on Frame Semantics, and to develop methods for automated assignment of corpora with frame semantic representations.

In the first (and current) phase of the SALSA project, semantic annotation is fully manual, and takes as its base the syntactically annotated TIGER treebank

[3] See [8] and the SALSA project homepage http://www.coli.uni-sb.de/lexicon

[2].[4] Due to the inherently sparser data seeds for semantic frames (as opposed to syntactic structures), it will be of utmost importance for the acquisition of high-performing stochastic models to process and collect data from larger corpora. In the second project phase we will thus proceed to semi-automatic semantic annotation of a much bigger, unparsed corpus. Here, a reliable and informative syntactic parse is essential: first, as a basis for semantic annotation, and second, since part of the information to be acquired is in itself syntactic.

Similar to the approach taken for syntactic annotation of the NEGRA corpus in [3, 4], SALSA aims at a bootstrapping approach for semantic annotation. Stochastic models for frame annotation are learned from a seed set of manual annotations, thus speeding up the manual annotation process and yielding more data for learning. Ultimately, we will learn increasingly refined models for frame assignment, by automatic annotation and re-training on larger corpora.

In the remainder of this paper, we discuss diverse architectures to implement a bootstrapping cycle for frame annotation that bridges the gap between treebank-based and large-scale free text processing. We investigate the applicability of the LFG formalism for the frame annotation task, and design an LFG syntax-semantics interface for frame assignment. We discuss alternative models for the interface, in terms of co-description and description by analysis and discuss their implications in terms of disambiguation effects and the integration of additional knowledge sources. Finally, we present a method for learning the required mappings between LFG and TIGER-SALSA representations.

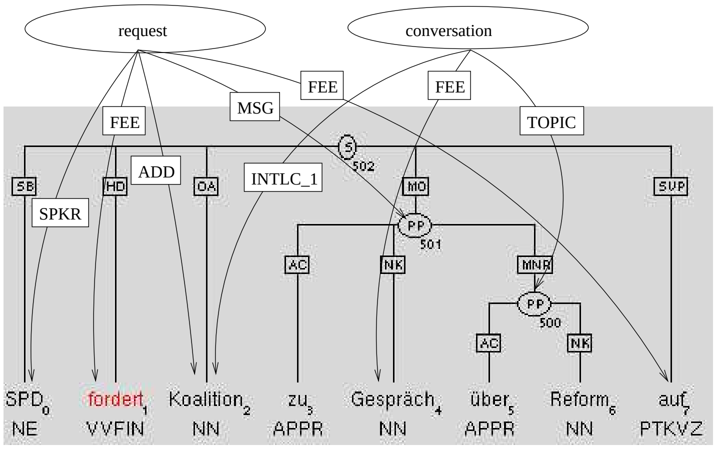
2 Annotating TIGER with Frame Semantic Roles

The TIGER Corpus [2] is a large syntactically annotated corpus of German. The annotation scheme is surface oriented and comparably theory-neutral. The dependency-oriented constituent structures provide information about *grammatical functions* (on edge labels) and *syntactic categories* (on constituent node labels). An example is given in the shaded tree of Fig. 1.

The FrameNet Project [1, 15] is based on Fillmore's Frame Semantics. A *frame* is a conceptual structure describing a situation. It is introduced by a *target* or *frame-evoking element* (*FEE*). Roles, called *frame elements* (FEs), are local to particular frames and identify the participants and props of the described situations. The aim of FrameNet is to provide a comprehensive frame-semantic description of the core lexicon of English. The current on-line version of the frame database consists of about 400 frames, covering about 6,900 lexical entries.

The SALSA Project [8] annotates frames on top of the TIGER treebank. Frames are represented as flat, independent trees, as shown in the white-labeled trees with curved edges in Fig. 1. The root is labeled with the frame name. Edges are labeled by frame elements or by 'FEE' and point to syntactic constituents.

[4] With 80.000 sentences, TIGER is comparable, in size, to the Penn Treebank. From our current gold corpus we estimate an average of about 3 frames per sentence, thus approx. 240.000 frame annotations for the entire TIGER corpus. This number is comparable to the English FrameNet resource used in [13, 10].



SPD requests that coalition talk about reform

Fig. 1. TIGER-SALSA graphical annotation

Fig. 1 contains two FEEs: *fordert... auf* (*auffordern*) and *Gespräch. auffordern* evokes the frame REQUEST. As the FEE does not form a single syntactic constituent, the label FEE is assigned to two edges. The SPEAKER is the subject (SB) NP *SPD*, the ADDRESSEE is the direct object (OA) NP *Koalition*, and the MESSAGE is the modifier (MO) PP *zu Gespräch über Reform*. The second FEE, the noun *Gespräch*, introduces the frame CONVERSATION, in which two groups talk to one another. The only NP-internal frame element is the TOPIC ("what the message is about") *über Reform*, whereas the INTERLOCUTOR-1 ("the prominent participant in the conversation") is realized by the direct object of *auffordern*.

Both the syntactic annotation of the TIGER corpus and the frames and frame elements that SALSA is adding are encoded in a modular XML format.

3 A Bootstrapping Architecture for Frame Annotation

The bootstrapping cycle for automatic frame and frame element assignment that we envision is similar to the process applied for NEGRA in [4]: First, stochastic models for frame and frame element annotation are learned from a seed set of manual annotations of the TIGER corpus. These models are applied to support interactive semi-automatic annotation of new portions of TIGER, with human annotators accepting or correcting assignments proposed by the system. New stochastic models derived from this larger set of TIGER data are applied for (semi-)automatic annotation of a larger, automatically parsed corpus, which again yields more training data, and continuously refined stochastic models.

3.1 From Treebank Annotation to Free Text Processing

To implement this bootstrapping cycle, we need a syntactic analyzer for free German text processing that (i) provides fine-grained syntactic information that is compatible with TIGER syntactic annotations and allows us to map the analyses of the syntactic analyzer to and from the TIGER-SALSA syntactic and semantic annotation format, and that (ii) delivers a high percentage of correctly analyzed data. Finally, (iii) we aim at a probabilistic parsing architecture that allows us to study the potential of semantics-driven syntactic disambiguation.

The most straightforward scenario is to employ a parser that delivers the same type of representations as used in the TIGER treebank. Yet, while first attempts to derive probabilistic grammars for German from the TIGER (or NEGRA) treebank [7, 6] are encouraging, they are still in need of improvement.[5]

Another possibility is to employ a broad-coverage parser for German that provides comparable fine-grainedness of analysis as exploited in the TIGER-SALSA annotations, and to provide a conversion routine for its output to match the TIGER format, or – conversely – to port the manually created TIGER-SALSA annotations to the output representation of such a parser.

In the first case, with TIGER syntax as main format, stochastic models would be derived from a combination of TIGER syntax and frame annotation. Transfer from the parser's output to the TIGER format would be needed in all phases of the cycle. In particular, the parser output for any corpus would have to be transformed to TIGER syntax. In the second case, with the parser's output as main format, stochastic models would be derived from a combination of the parser's format and frame annotation, which means that a semantic frame projection for the parser output is needed. Transfer between TIGER-SALSA and parser output representation would be needed only in the first phases of the cycle, while processing data of the TIGER corpus.[6] Moreover, this scenario lends itself to an integrated semantic disambiguation model in the sense of (iii).

3.2 German LFG for Corpus Processing and Frame Annotation

We propose to use a German LFG grammar to support the bootstrapping cycle for frame annotation. The TIGER annotation process was supported by semi-automatic processing with a German LFG grammar [2, 20].[7] In addition to the LFG-to-TIGER transfer module developed there, [11] has recently built a mapping from TIGER to LFG f-structures. These automatic conversions ensure that LFG representations are rich enough to match the syntactic TIGER representations. [2] report a coverage of 50% for the LFG grammar, with 70% precision. Newer figures are not yet available, but we expect the German grammar to soon reach the performance of the English LFG grammar described in [18].

We further opt for the second scenario of the previous paragraph: using LFG f-structures as the primary basis for building stochastic models. This scenario

[5] [7] do not assign functional labels, whereas [6] produce LFG (proto) f-structures. Though not fully comparable, [6] could be used for our purposes in similar ways, and possibly in tandem with the manually developed LFG grammar described below.

[6] Manual annotation is aided by an annotation tool based on the TIGER-SALSA format [9]. Also, the TIGER-SALSA corpus is intended as a theory-neutral reference corpus, and must include sentences that are out-of-coverage for the chosen parser.

[7] The grammar is being developed at the IMS, University of Stuttgart.

requires the design of an LFG semantics projection for frame assignment, and a mapping between TIGER-SALSA and LFG syntax-semantics representations to implement the bootstrapping cycle. However, it restricts transformations between syntactic formats to the learning phase, and lends itself to an exploration of semantic features for syntactic disambiguation. A further advantage of this model is that it allows for the extension of existing probabilistic methods for syntactic disambiguation in [18] to online semantic classification and disambiguation. The stochastic tools employed in [18, 17] – provided with the LFG processing platform XLE – support training and online application of loglinear models. We can thus explore the disambiguation effects of semantic annotation in combination with, or independent from syntactic disambiguation.

4 LFG for Frame Annotation: Chances and Challenges

In the following sections we investigate the applicability of LFG for the frame annotation task, and design a syntax-semantics interface for Frame Semantics.

Lexical Functional Grammar [5] assumes multiple levels of representation for linguistic description. Most prominent are the syntactic representations of c(onstituent)- and f(unctional)-structure. The correspondence between c- and f-structure is defined by functional annotations of CFG rules and lexical entries. This architecture can be extended to semantic (and other) projection levels [14].

The f-structure representation abstracts away from surface-syntactic properties, and allows for uniform reference to syntactic dependents in diverse syntactic configurations. This is important for the task of frame annotation, as it abstracts away from aspects of syntax that are irrelevant to frame (element) assignment.

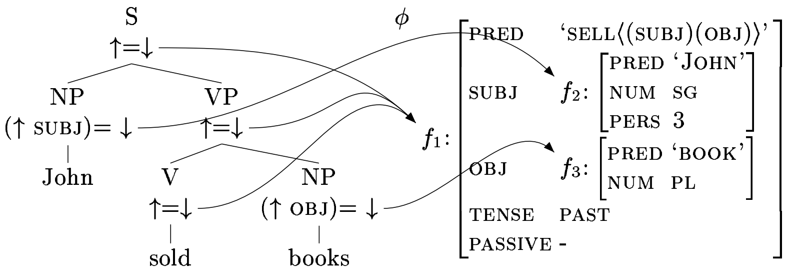


Fig. 2. LFG projection architecture with c– and and f–structure representation

The LFG syntactic analysis of word order, control and raising constructions, long-distance dependencies and coordination provides f-structure representations where non-local or implicit arguments are localized and thus allow for uniform association of local grammatical functions with frame semantic roles.

In (1), the SELLER role can be uniformly associated with the local SUBJect of *sell*, even though it is realized as a relative pronoun of *come* that controls the SUBJect of *sell*, (b.) an implicit second person SUBJ, (c.) a non-overt SUBJ controlled by the OBLique object of *hard*, and (d.) a SUBJ (*we*) in VP coordination.

(1) a. .. *the woman who* had come in to *sell flowers* to the customers overheard their conversation .. (from [15])
b. Don't *sell the factory* to another company.
c. It would be hard for *him* to *sell newmont shares* quickly. (from [15])
d. .. *we* decided to sink some of our capital, buy a car, and *sell it* again before leaving. (from [15])

More challenging are phenomena as in (2.a,b), where the SUBJ of *sell* is not syntactically represented as identical to (a.) the passive SUBJ of the matrix clause, or (b.) the matrix SUBJ of an embedded adjunct clause containing *sell.* Here the SELLER semantic role has to be assigned nonlocally (unless coreference information is made available).

(2) a. .. the old adage about *most people* simply refusing to move rather than *sell their house* ..
b. .. *we*'d do the maintenance and watering instead of just *selling* the plants .. (both from [15])

There are cases where a frame-evoking element and one of its FEs are both parts of a single compound, e.g. in (3) the noun modifier *Auto* fills the GOODS role in the COMMERCE frame evoked by the head noun *Verkäufer.* The LFG f-structure analysis of nominal compounds provides a (flat) decomposition into a nominal head and a set NMOD of noun modifiers. The NMOD modifier *Auto* can thus be represented to fill the GOODS role in the frame evoked by the head noun.

(3) *Autoverkäufer* geben zur Zeit bis zu 10% Rabatt.
Car dealers offer nowadays up to 10% reduction.

Formal Devices. The LFG formalism provides powerful descriptional devices that are essential for the design of a flexible syntax-semantics interface.

The regular expression-based specification of uncertain embedding paths within f-structures – both outside-in and inside-out [16] – makes it possible to refer to any piece of f-structure from anywhere within the f-structure.

The restriction operator [19] permits reference to partial f-structures. It can be used to link semantic roles to partial f-structures, such as grammatical functions to the exclusion of embedded material (e.g. sentential adjuncts).

Examples that use these devices will be discussed in Section 5.1.

5 LFG Syntax-Semantics Interface for Frame Semantics

5.1 A Frame Semantics Projection

As a direct transposition of the SALSA annotation format we can define a Frame Semantics projection σ_f from the level of f-structure (compare Figs. 1 and 3).

While in the traditional LFG projection architecture (as in [14]) f-structure predicates are related to predicate-argument structures in s-structure, we define the σ_f–projection to introduce elementary frame structures, with attributes

FRAME, FEE (frame-evoking element), and frame-specific role attributes. Fig. 3 displays the σ_f–projection for the sentence in Fig. 1.[8]

Fig. 4 states the lexical constraints that define this mapping. σ_f is defined as a function of f-structure. Thus, the verb *auffordern* introduces a node $\sigma_f(\uparrow)$ in the frame semantics projection of $\uparrow$, its local f-structure, and defines its attributes FRAME and FEE. The frame elements are defined as σ_f–projections of the verb's SUBJ, OBJ, and OBL OBJ functions. For example, the SPKR role, referred to as $(\sigma_f(\uparrow)\ \mathrm{SPKR})$, the SPKR attribute in the frame projection $\sigma_f(\uparrow)$ of $\uparrow$, is defined as identical to the σ_f–projection of the verb's SUBJ function, $\sigma_f(\uparrow\ \mathrm{SUBJ})$.[9]

The noun *Gespräch*, which evokes the CONVERSATION frame, illustrates the use of inside-out functional equations to refer to material outside the local f-structure of a frame evoking predicate. The INTERLOCUTOR1 (INTLC_1) role corresponds to the OBJ of *auffordern*. This function is accessible from the noun's f-structure via the inside-out equation $((\mathrm{OBL\ OBJ}\uparrow)\ \mathrm{OBJ})$: starting from $\uparrow$ (the f-structure of *Gespräch*), the path leads inside-out to the f-structure $(\mathrm{OBL\ OBJ}\ \uparrow)$ of the verb, from which it descends to the verb's OBJ: $((\mathrm{OBL\ OBJ}\uparrow)\ \mathrm{OBJ})$.

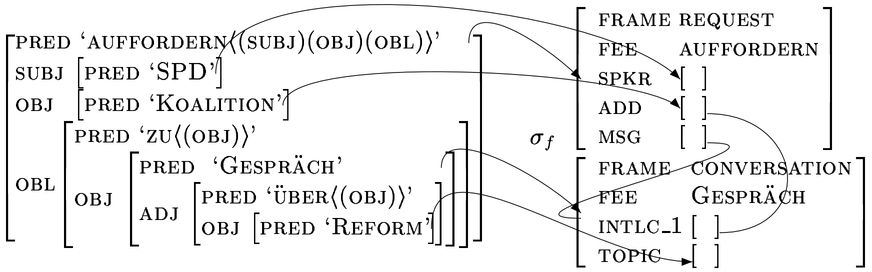


Fig. 3. LFG projection architecture for Frame Annotation

fordert V,
$(\uparrow\mathrm{PRED})=$‘AUFFORDERN$\langle(\uparrow\mathrm{SBJ})(\uparrow\mathrm{OBJ})(\uparrow\mathrm{OBL})\rangle$’
...
$(\sigma_f(\uparrow)\ \mathrm{FRAME}) = \mathrm{REQUEST}$
$(\sigma_f(\uparrow)\ \mathrm{FEE}) = (\uparrow\ \mathrm{PRED\ FN})$
$(\sigma_f(\uparrow)\ \mathrm{SPKR}) = \sigma_f(\uparrow\ \mathrm{SUBJ})$
$(\sigma_f(\uparrow)\ \mathrm{ADD}) = \sigma_f(\uparrow\ \mathrm{OBJ})$
$(\sigma_f(\uparrow)\ \mathrm{MSG}) = \sigma_f(\uparrow\ \mathrm{OBL\ OBJ})$

Gespräch N,
$(\uparrow\mathrm{PRED})=$ ‘GESPRÄCH’
...
$(\sigma_f(\uparrow)\ \mathrm{FRAME})= \mathrm{CONVERSATION}$
$(\sigma_f(\uparrow)\ \mathrm{FEE})= (\uparrow\ \mathrm{PRED\ FN})$
$(\sigma_f(\uparrow)\ \mathrm{INTLC1})=\sigma_f((\mathrm{OBL\ OBJ}\uparrow)\ \mathrm{OBJ})$
$(\sigma_f(\uparrow)\ \mathrm{TOPIC})= \sigma_f(\uparrow\ \mathrm{ADJ\ OBJ})$

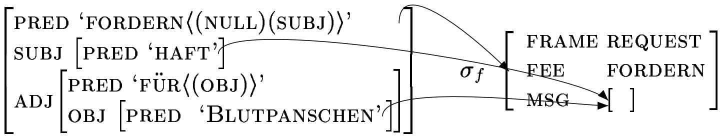
Fig. 4. Frame projection by co-description

Frames in Context. The projection of frames in context can yield partially connected frame structures. In Fig. 3, *Gespräch* maps to the MSG of REQUEST and also introduces a frame of its own, CONVERSATION. Due to the syntactic relation $(f_1\ \mathrm{OBL\ OBJ})=f_2$, (with f_1 and f_2 the f-structures of *auffordern* and *Gespräch*, respectively), the equations $(\sigma_f(f_1)\ \mathrm{MSG})$, $\sigma_f(f_1\ \mathrm{OBL\ OBJ})$ and $(\sigma_f(f_2))$ all refer to a single node in the σ_f–projection. The CONVERSATION frame is thus defined as an instantiation, in context, of the MSG role of a REQUEST frame.

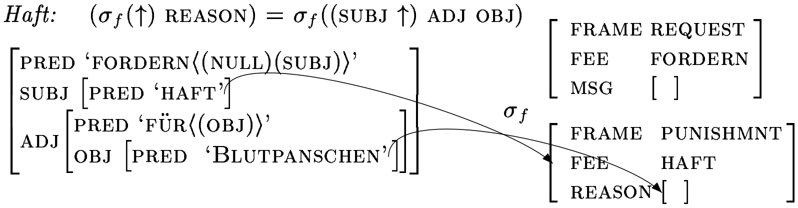
[8] In this paper we omit details involving set-based representations for ADJuncts.
[9] The MSG is coindexed with the lower frame, a projection of the noun *Gespräch*.

(4) a. Haft für Blutpanschen gefordert
[NP_SB Haft] [PP_MO für Blutpanschen] [VV_HD gefordert]
'Prison sentence demanded for unsanitary blood collection'

b. *fordert*: $(\sigma_f(\uparrow)\ \text{MSG}) = \sigma_f(\uparrow\ \text{SUBJ})$
$(\sigma_f(\uparrow)\ \text{MSG}) = \sigma_f(\uparrow\ \text{ADJ OBJ})$



c. *Haft:* $(\sigma_f(\uparrow)\ \text{REASON}) = \sigma_f((\text{SUBJ}\ \uparrow)\ \text{ADJ OBJ})$



Special Configurations. Potentially problematic are configurations where multiple syntactic constituents are mapped to a single semantic role, as they may lead to an inconsistency in the σ_f–projection.[10]

An example is shown in (4). The SUBJect *Haft* and the modifier PP *für Blutpanschen* have jointly been annotated as the MSG role in the REQUEST frame of *fordern*. The projection of the MSG role from two constituents can be modeled by the equations in (4.b). Yet this simple model will lead to an inconsistency if the involved predicates introduce individual frames, at the same level of embedding.

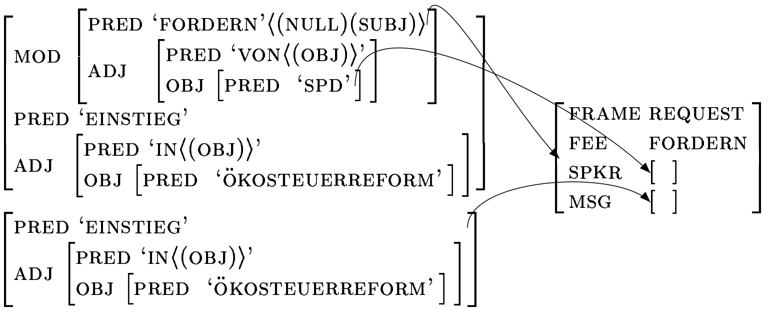
In (4), the SUBJ *Haft* evokes a frame PUNISHMENT, in which the modifier *für Blutpanschen* fills the REASON role, as defined in (4.c). Due to this embedding asymmetry of SUBJ and modifier at the semantic level, the joint equations in (4.b,c) do not lead to inconsistency, but a circular semantic structure: By (4.b), ADJ OBJ and SUBJ are mapped to the same σ_f value both in (4.b) and (4.c), so in (4.c) the REASON of PUNISHMENT and the PUNISHMENT frame itself have to be equal – which is not a correct representation of the meaning of the sentence.

We found that in the SALSA annotations asymmetric embedding at the semantic level is the typical pattern for constituents that jointly constitute a single frame element. We therefore propose to make use of **functional uncertainty** equations to accommodate for embedded frames within either one of the otherwise re-entrant constituents. In (4.b), we thus relax the equation mapping the PP to MSG to $(\sigma_f(\uparrow)\ \text{MSG ROLE}*)=\sigma_f(\uparrow\ \text{ADJ OBJ})$, with ROLE instantiating to REASON in (4). In this way, the functional uncertainty over possible semantic roles accommodates for (possibly unassigned) asymmetrically embedded frames.

Another typical configuration where discontinuous constituents correspond to a single semantic role is illustrated in (5): *der* and *Einstieg in eine Ökosteuerreform* correspond to the MSG of a REQUEST, which is introduced by the

[10] In the existing annotations, 909 (or 1.2%) of the frame elements match this pattern.

(5) der von der SPD geforderte Einstieg in eine Ökosteuerreform
the by the SPD demanded start of an ecological tax reform
'the start of an ecological tax reform, demanded by the SPD'



adjectival head *geforderte* within the modifier of the phrase. Its *by-phrase* adjunct fills the SPKR role. This case differs from the one above in that the discontinuous constituents jointly form a headed phrase (with a local PRED in f-structure).

This configuration is similar to the well-known head-switching phenomena, and can be represented by use of the **restriction** operator [19]. The equation $(\uparrow\backslash\{\text{MOD}\})$ refers to the partial f-structure (displayed as a copy in (5)) consisting of $\uparrow$ *without* the function MOD. This unit can be defined to fill the MSG role of REQUEST. Since the frame evoking head is embedded within MOD itself, this involves an inside-out functional equation: $(\sigma_f(\uparrow)\ \text{MSG}){=}\sigma_f((\text{MOD}\ \uparrow)\backslash\{\text{MOD}\})$.

5.2 Co-description vs. Description by Analysis

Co-description. In the projection architecture we just presented, f- and s-structure equations jointly determine the valid analyses of a sentence. This method of defining and evaluating projection levels is called co-description.

With co-description, syntactic and semantic analysis interact, leading to semantics-driven syntactic disambiguation. Our example sentence in Fig. 1 is syntactically four-ways ambiguous. *SPD* and *Koalition*, being unmarked for case, can both be SUBJ or OBJ; the PP *über Reform* can be attached to *Gespräch* (as displayed), or be an adjunct of *auffordern*. However, the semantic constraints for *Gespräch* in Fig. 4 define its role TOPIC as a PP adjunct (ADJ OBJ) of the local head. This eliminates the readings where *Reform* is adjoined to the verb.

Description by Analysis (DBA). An alternative to the co-descriptive model is semantics construction via description by analysis [14]. Here, semantics is built on top of fully resolved (disjunctive) f-structure analyses. Analyses that are consistent with syntax-semantics mapping constraints are semantically enriched – while remaining solutions are left untouched.

Technically, this architecture can be realized by use of a term rewriting system as employed in transfer.[11] In a transfer approach, feature structures are described by sets of predicates. Non-prefixed predicates are constraints on the applicability of a rule, to be used e.g. for describing the shape of the f-structure:

[11] The XLE includes a transfer component that operates on packed f-structures [12].

pred(A,auffordern), subj(A,B), obj(A,C), obl(A,D), obj(D,E)

Here, features are encoded by predicates that take as arguments atomic values or variables for feature structure nodes. Predicates prefixed with + introduce new nodes and values: Encoding the σ_f–projection by a predicate sem_f, we can enrich the matched f-structure with the frame information for *auffordern* and link the SPKR role to the σ_f–projection of the SUBJ *SPD*:

+sem_f(A,SemA), +frame(SemA,request), +fee(SemA,auffordern),
+sem_f(B,SemB), +spkr(SemA,SemB)

Implications. Both models are equally powerful in terms of expressiveness.[12] While co-description integrates the frame semantics projection into the grammar and parsing process, DBA keeps it as a separate module. This means that DBA is more suited for the development phase of LFG-based frame assignment, while co-description, which is particularly interesting for studying joint syntactic and semantic disambiguation, may be used in later stages. With DBA, semantics does not interfere with grammar design and can be developed separately. Subsequently the transfer rule sets can be automatically converted to equivalent co-description constraints. Due to its greater modularity, the DBA approach also facilitates extensions of the projection architecture to include external semantic knowledge sources, such as word sense, named entity typing, and coreference.

6 Learning Translations between Representations

In the previous section, we investigated representational aspects of an LFG syntax-semantics interface for frame assignment. To implement the full bootstrapping cycle for (semi-)automatic frame assignment (cf. Sec. 3), we finally need a mapping to translate between TIGER-SALSA representations and LFG representations with frame semantics projection. With such a mapping, we can (i) port TIGER-SALSA annotations to the LFG format, to build a seed corpus for stochastic modeling, and (ii) extract transfer-based frame assignment rules from the seed annotations, to disjunctively apply them to new sentences. In the reverse direction, we can (iii) convert automatically assigned frames to the TIGER-SALSA format, to be corrected or confirmed by human annotators.

Transfer-based conversions between the LFG and TIGER formats have been built in [20, 11]. But the transfer rules need to be updated with every change of the grammar. Instead, we propose to learn translations between LFG and TIGER formats using a method inspired by Example-based MT. We use the aligned LFG-TIGER treebank of [11] as a “parallel corpus”. Starting out with pairs of TIGER and LFG structures, we want to obtain parallel path descriptions that – within the respective syntactic structures – identify the relevant frame (evoking) elements. Since we are operating on identical sentences, we can use the surface strings to establish the corresponding path descriptions.[13]

[12] Except for functional uncertainty, which in transfer can only be of bounded length.

[13] Hence the relation to EBMT, where translation rules are learned from examples. Here, we learn correspondences between syntactic structures for ‘identical’ languages.

For example, the paths that identify the SPKR in our running example can be described by the correlated TIGER and LFG path expressions (cf. Figs. 1,3):

	TIGER path	string	LFG f-struct. path	LFG c-struct. path
SPKR	[S,SB,NE]	*SPD*	[SUBJ]	[S,NP]

Paths are given from the root down. For TIGER, we use paths with alternating categorial (node) and functional (edge) labels. In the LFG path descriptions functional and categorial descriptions are separated. To avoid spurious ambiguities in case of non-branching structures, we choose the shortest path (highest constituent) that yields the exact target string. For frame (evoking) elements that correspond to multiple or discontinuous elements (such as *fordert auf*) we generate a list of paths for the individual constituents:

	TIGER path	string	LFG f-struct. path	LFG c-struct. path
REQUEST	[S,HD,VVFIN]	*fordert*	[PRED]	[S,VP,V]
	[S,SVP,PTKVZ]	*auf*		[S,SVP,PTKVZ]

With these correspondences we can port frame annotations from the TIGER-SALSA corpus to the parallel LFG corpus, and freely translate between these formats. They can further be used to extract generalized transfer frame annotation rules, for application to new LFG-parsed sentences.

This method depends on a sufficiently rich set of seed annotations as training data, and for refinement of the rule extraction algorithm. This is ensured by the first bootstrapping cycles, with annotations being checked by human annotators.

7 Conclusions and Perspectives

This study investigates a general architecture for (semi-)automatic frame assignment that supports the transition from treebank-based annotation to large-scale corpus processing in a bootstrapping architecture, using LFG as the underlying syntactic formalism. Besides linguistic considerations, this choice is motivated by the availability of a large-scale German LFG grammar and a powerful processing platform that includes a translation component and tools for stochastic modeling. This combination will allow us to study the (combined and individual) effects of syntactic and semantic disambiguation.

We designed an LFG syntax-semantics interface for frame semantics and showed how to address potentially problematic configurations. To our knowledge, this is the first study to investigate frame semantics as a target representation for semantics construction from syntax. We discussed two architectures for this syntax-semantics interface: the co-descriptive model, where semantic construction is integrated into the grammar, and description by analysis, which works as a separate module and is more robust. Rules for frame semantics projection can be derived from the annotated TIGER-SALSA corpus, given a mapping between the TIGER and LFG syntax formats. We propose to learn this mapping from the 'aligned' TIGER and LFG annotations of the TIGER corpus, to alleviate the maintenance problem of hand-coded transfer rules for corpus conversion.

References

1. C.F. Baker, C.J. Fillmore, and J.B. Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of COLING-ACL 1998*, Montréal, Canada.
2. S. Brants, S. Dipper, S. Hansen, W. Lezius, and G. Smith. 2002. The TIGER Treebank. In *Proc. of the Workshop on Treebanks and Linguistic Theories*, Bulgaria.
3. T. Brants, W. Skut, and B. Krenn. 1997. Tagging Grammatical Functions. In *Proceedings of EMNLP*, Providence, RI, USA.
4. T. Brants, W. Skut, and H. Uszkoreit. 1999. Syntactic Annotation of a German newspaper corpus. In *Proceedings of the ATALA Treebank Workshop*, Paris, France.
5. J. Bresnan. 2001. *Lexical-Functional Syntax.* Blackwell Publishers, Oxford.
6. A. Cahill, M. Forst, M. McCarthy, R. O'Donovan, C. Rohrer, J. van Genabith, and A. Way. 2003. Treebank-Based Multilingual Unification-Grammar Development. in: *Proceedings of ESSLII'03 – Workshop on Ideas and Strategies in Multilingual Grammar Development*, Vienna, Austria.
7. A. Dubai and F. Keller. 2003. Probabilistic Parsing for German using Sister-Head Dependencies. In *Proceedings of the ACL 2003*, Sapporo, Japan.
8. K. Erk, A. Kowalski, S. Padó, and M. Pinkal. 2003. Towards a Resource for Lexical Semantics: A Large German Corpus with Extensive Semantic Annotation. In *Proceedings of the ACL 2003*, Sapporo, Japan.
9. K. Erk, A. Kowalski and S. Padó. 2003. The SALSA Annotation Tool. In *Proceedings of the Workshop on Prospects and Advances in the Syntax/Semantics Interface*, Nancy, France.
10. M. Fleischmann, N. Kwon and E. Hovy. 2003. Maximum Entropy Models for FrameNet Classification. In *Proceedings of EMNLP 2003*, Sapporo, Japan.
11. M. Forst. 2003. Treebank Conversion – Establishing a testsuite for a broad-coverage LFG from the TIGER treebank. In *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora (LINC '03)*, Budapest, Hungary.
12. A. Frank. 1999. From Parallel Grammar Development towards Machine Translation. In *Proceedings of Machine Translation Summit VII*, Singapore.
13. D. Gildea and D. Jurafsky. 2002. Automatic Labeling of Semantic Roles. In *Computational Linguistics 28.3.*
14. P.-K. Halvorsen and R.M. Kaplan. 1995. Projections and Semantic Description in Lexical-Functional Grammar. In M. Dalrymple, R.M. Kaplan, J.T. Maxwell, and A. Zaenen, eds, *Formal Issues in Lexical-Functional Grammar*, CSLI Lecture Notes.
15. C.R. Johnson, C.J. Fillmore, M.R.L. Petruck, C.F. Baker, M. Ellsworth, J. Ruppenhofer, and E.J. Wood. FrameNet: Theory and Practice. `http://www.icsi.berkeley.edu/~framenet/book/book.html`.
16. R.M. Kaplan and J.T. III. Maxwell. 1995. An Algorithm for Functional Uncertainty. In M. Dalrymple, R.M. Kaplan, J.T. Maxwell, and A. Zaenen, eds, *Formal Issues in Lexical-Functional Grammar*, CSLI Lecture Notes.
17. S. Riezler, T.H. King, R. Crouch, and A. Zaenen. 2003. Statistical sentence condensation using ambiguity packing and stochastic disambiguation methods for Lexical-Functional Grammar. In *Proceedings of HLT-NAACL'03*, Canada.
18. S. Riezler, T.H. King, R.M. Kaplan, R. Crouch, J.T. III Maxwell, and M. Johnson. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and Discriminative Estimation Techniques. In *Proceedings of ACL'02*, Philadelphia, PA.
19. J. Wedekind and R.M. Kaplan. 1993. Restriction and correspondence-based translation. In *Proceedings of EACL*, Utrecht.
20. H. Zinsmeister, J. Kuhn, and S. Dipper. 2002. Utilizing LFG Parses for Treebank Annotation. In *Proceedings of the LFG 2002 Conference*, Athens, Greece.