Automatic Assignment of Frame Semantics Using Syntax-Semantics Interface in LFG

.

Jiří Semecký

Abstract

Recently, there is a global outlook that quality NLP applications are in need of deeper semantic analysis. In order to obtain larger semantically annotated data, we need a method for automatic assignment of semantic structures. In this article we present a method for transferring semantic annotation from the SALSA project to the LFG parsing architecture, as well as a method for assigning semantic structures based on rules extracted from data. The paper is divided as follows: in the first part we give an overview of ongoing projects similar or related to our task, and of formalisms we built on in our work. In the second part we describe our approach in details concentrating on the technical aspects of the solution. In the end of the second part we summarize and discuss our results and suggest further development.

Contents

Part I
State of the Art

In this part, first we describe three projects dealing with semantic annotation: FrameNet (section 1), PropBank (section 2), and the Prague Dependency Treebank (section 3). Then, in section 4, we introduce approaches dealing with automatic semantic labeling. Last, we give an overview of two basic grounds of our work, the SALSA project(section 5), and the LFG parsing architecture(section 6).

1 FrameNet

FrameNet is a Berkeley University project that creates a large semantic lexicon of English for NLP applications providing information on predicate-argument structure. FrameNet is based on the theory of frame semantics, originally introduced by Fillmore in ((0)).

Frames are considered to be conceptual structures or prototypical situations. They are evoked by predicates (**frame evoking elements, FEE**s) and they are associated with other constituents (**frame elements, FE**s) which correspond to the participants of the situations.

A particular combination of frame elements in FrameNet is local to a given frame – their names are domain specific (e.g. SPEAKER, MESSAGE, and TOPIC in COMMUNICATION frame) – some of the frame elements are more general, some of them are specific to a small group of lexical items. A frame definition in the FrameNet database consists of a frame description, and a list of frame elements and their descriptions. Moreover, the frame definition is also accompanied by a list of predicates (verbs, and nouns) that can evoke this frame, i.e. can serve as frame evoking elements of a particular frame (e.g. frame COMMUNICATION can be evoked by the verbs *speak*, *talk*, the noun *dialog*, ...). Furthermore, FrameNet contains links to other lexical resources – e.g. WordNet. Figure 1 presents an example of a FrameNet frame definition.

Sentences are described in terms of frames, each frame is evoked by one frame evoking element, and some of its frame elements[1] are assigned to syntactic constituents of the sentence. Figure 2 shows an example sentence with an assigned STATEMENT frame.

FrameNet defines relation of inheritance among frames, a frame can inherit from one or more other frames. For example, STATEMENT and COMMUNICATION_NOISE inherit from COMMUNICATION frame. Moreover, FrameNet defines relation of *using*, which describes using of a frame within another frame, e.g. COMMUNICATION frame uses TOPIC frame and is used by ATTEMPT_ SUASION, CANDIDNESS, COMMITMENT, and other frames.

The FrameNet database is accessible via Internet at the address of the FrameNet project[2] and currently contains 482 frames and thousands of lexical entries.

2 Proposition Bank

Proposition Bank (PropBank) (0) is a project of the University of Pennsylvania which aims at adding a layer of semantic annotation to the Penn English TreeBank.[3] The basis for semantic annotation are syntactically hand-annotated sentences from the Penn Treebank II Wall Street Journal corpus of a million of words.

Each predicate defined in PropBank is assigned arguments which are numbered sequentially as **Arg0**, **Arg1**, **Arg2**, ..., and the numbering is predicate dependent. **Arg0** is usually the subject of a verb, **Arg1** direct object of a transitive verb, etc. This is a conceptual difference from the FrameNet project,

[1] Not all frame elements have to be present in the sentence (i.e. event).

[2] http://www.icsi.berkeley.edu/~framenet/

[3] http://www.cis.upenn.edu/~treebank/

Frame:	STATEMENT	This frame contains verbs and nouns that communicate the act of a Speaker to address a Message to some Addressee using language. A number of the words can be used performatively, such as *declare* and *insist*.
Frame elements:	*Speaker*	is the person who produces the Message (whether spoken or written). It is normally expressed as the External Argument of predicative uses of the TARGET word, or as the Genitive modifier of the noun.
	Addressee	receives a Message from the Communicator (Speaker).
	Message	is the FE that identifies the content of what the Speaker is communicating to the Addressee. It can be expressed as a clause or as a noun phrase.
	Medium	is the physical entity or channel used by the Speaker to transmit the statement.
	Topic	The Topic is the subject matter to which the Message pertains. It is normally expressed as a PP Complement headed by "about", but in some cases it can appear as a direct object.
Frame evoking elements:	*add.v, address.v, admission.n, admit.v, affirm.v, affirmation.n, allegation.n, allege.v, announce.v, announcement.n, assert.v, assertion.n, attest.v, aver.v, avow.v, avowal.n, boast.n, boast.v, brag.v, caution.v, claim.n, claim.v, comment.n, comment.v, complain.v, complaint.n, concede.v, concession.n, confess.v, confession.n, ...*	

Figure 1: Example of STATEMENT frame definition

Speaker	FEE	Addressee	Medium
Kim	*QUESTIONED*	*me*	*over the phone.*

Figure 2: Sentence with assigned STATEMENT frame

in which semantic roles are given meaningful frame dependent names, i.e. predicates of the same frame share the role names. Arguments in PropBank are, nevertheless, given mnemonic labels too. These labels are verb specific, however some of them tend to be specific to a group of verbs, closer to FrameNet conventions.

In addition to numbered arguments, a predicate can be assigned additional mandatory adjuncts[4], which are not numbered but rather labeled with 'ArgM-' extended with a secondary functional tags: (LOC for location, TMP for time, MNR for manner, DIR for direction, CAU for cause, NEG for negation marker, MOD for modal verb, PRP for purpose, and ADV for general-purpose modifier). Secondary predication is marked with tag PRD in the cases where one argument of a verb is a predicate upon another argument of the same verb.

In PropBank, verbs take usually three or four arguments:

[4]If the predicate requires the particular adjunct strongly enough.

obtain.01 ("get")	
Arg0:	receiver
Arg1:	thing gotten
Arg2:	received from

They can take no arguments (e.g. weather predicates):

hail.01 ("weather phenomenon")

Maximally, some verbs take six arguments:

edge.01 ("move slightly")	
Arg1:	Logical subject, patient, thing moving
Arg2:	EXT, amount moved
Arg3:	start point
Arg4:	end point
ArgM-LOC:	medium
Arg5:	direction–REQUIRED

The semantics of arguments is predicate dependent but it follows certain guidelines. The authors try to keep consistency across semantically related verbs. For instance *buy* and *purchase* have the same set of arguments, and they are similar to the set of arguments of *sell*, cf. Figure 3. However, two senses of a single verb can have different argument labels.

Figure 4 shows an example of PropBank annotation.

Purchase	**Buy**	**Sell**
Arg0: buyer	Arg0: buyer	Arg0: seller
Arg1: thing bought	Arg1: thing bought	Arg1: thing sold
Arg2: seller	Arg2: seller	Arg2: buyer
Arg3: price paid	Arg3: price paid	Arg3: price paid
Arg4: benefactive	Arg4: benefactive	Arg4: benefactive

Figure 3: Semantic roles of predicates *buy*, *purchase*, and *sell*

Arg0	REL	Arg1	Arg3
The holder	*buys*	*$1000 principal amount*	*of debentures at par.*
Arg0	REL	Arg4	Arg1
John	*bought*	*his mother*	*a dozen roses.*

Figure 4: Sentences with PropBank annotation

3 Valency in the Prague Dependency Treebank

The theory of valency in Praguian school is based on the framework of Functional Generative Description (FGD) (0). In the FGD, language is described on different layers where adjacent layers are related in the way that elements of the upper layer are functions of elements of the lower one, and elements of the the lower one are forms of elements of the upper one. Going from lower layers to higher ones means going from the surface representation to the (literal) meaning.

The Prague Dependency Treebank (PDT) is a manually annotated corpus based on the FGD theory. Data of the PDT are part of the Czech National Corpus.[5] Data are annotated on three different layers (0), namely morphological, analytical, and tectogrammatical layer.

Whereas the **morphological layer** deals with individual words, the higher levels (analytical, and tectogrammatical layer) use the tree-based sentence (syntactic) structure. The **analytical layer** consists of (surface) syntactic annotation using dependency relations – sentences are described purely in terms of analytical dependences (subject, object, ...), and the representation includes all and only the surface lexical items. The **tectogrammatical layer** describes the underlying syntactic structure – sentence is described in terms of tectogrammatical dependences (actor, patient, ...), and abstracting from the surface representation, only autosemantic words remain and items deletecd in the surface shape of the sentence are reconstructed.

The theory of valency (0) is based on the tectogrammatical representation. Valency is understood as an attribute of auto-semantic lexical units. On the tectogrammatical level we assume that every verb, noun, adverb, and adjunct has valency, which is described by valency frames. Valency frame consists of possible modifiers of the lexical unit – actants and free modifiers (adjuncts).

3.1 The VALLEX Lexicon

VALLEX (0) is a manually created valency lexicon of verbs for Czech, based on the valency theory. VALLEX is being built since 2001 and the work is still in progress. The VALLEX version 1.0[6] defines valency for over 1400 Czech verbs and contains over 3800 frames.

Each verb in the VALLEX lexicon is represented by a headword lemma, and consists of one or more frames that correspond to the meanings of the verb. Each frame is described by a list of valency slots and every valency slot is defined by the tectogrammatical function and its possible syntactic realizations. Moreover, each frame is accompanied by an explanation of the meaning (using synonyms or glosses), an example sentence or phrase and the aspectual counterpart (if it exists). Some of the verbs are assigned semantic classes.

An example of a VALLEX entry for the Czech verb *dodat* is displayed in Figure 5. The verb entry contains five frames for different meanings of the verb, namely *supply*, *ship*, *mention*, *add*, and *encourage*:

4 Automatic Labeling of Semantic Roles

One of the first works dealing with automatic assignment of semantic roles could be found in (0) and (0).

In (0), the author assigns tectogrammatical functors in the PDT. The assignment method uses a combination of hand written rules and dictionary based methods.

The hand written rules, which are used first, determine functors of tectogrammatical nodes according to their morphological categories (part of speech, voice of governing verb), and the analytical functions.

The dictionary based methods come into force if none of the hand written rule succeeds. Based on the training data, they find which adverbs and subordinate conjunctions are unambiguous in the tectogrammatical function, and assign the extracted functor to them. After that, the statistics about combinations of prepositions and nouns are used. Last, the methods decides upon the similarity with instance seen in the training data.

The author reported accuracy 78.2 % on relatively small data (the training set contained 6049 annotated nodes).

[5] http://ucnk.ff.cuni.cz/english/index.html

[6] http://ckl.ms.mff.cuni.cz/zabokrtsky/vallex/1.0/

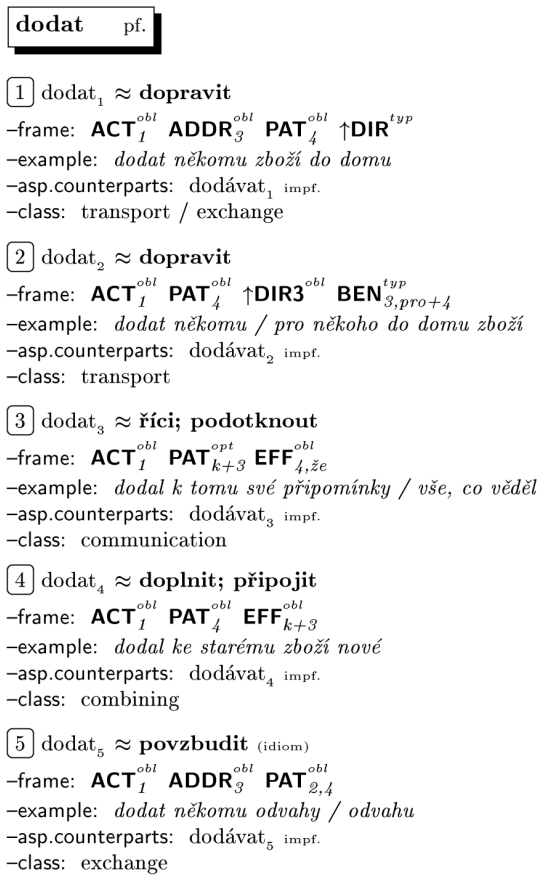


Figure 5: Example of VALLEX verb definition.

In (0) the authors propose a statistical method for automatic assignment of FrameNet roles. The system uses parsed sentences automatically determined by the statistical parser ((0)). Semantic roles are assigned on the basis of a probabilistic model combining the following syntactic features:

- **Phrase type (pt):** states the syntactic type of the phrase that is being assigned a semantic role. The phrase types include *NP*, *PP*, *VP*, *S*, etc.

- **Governing category (gov):** states the type of the governing node. It can be of two values: *S* and *VP*, corresponding to subject and object, respectively. Only NPs are assigned this feature.

- **Path in the parse tree (path):** states the complete path from predicate to the phrase listing all the phrase names on the way up the tree followed by the phrase names on the way down to the phrase. An example of the path is
  $VB \uparrow VP \downarrow NP$
  for a path going from main verb to its object.

- **Position (pos):** states whether the phrase is before or after the predicate in the surface representation of the sentence. This feature is strongly correlated to the governing category; however, its presence should be to the benefit in case of wrong parses or sparse data.

- **Voice (v):** distinguishes whether the predicate is in the active or in the passive voice. Note that this information is essential, for the subject in a sentence with a verb in the active voice is expressed as an object in a sentence with the verb in the passive voice and vice versa.

- **Head word (hw):** is a lexical dependency feature. Head words of noun phrases can be used to express selectional restrictions on the semantic types of fillers. For example, in a COMMUNICATION frame, nouns headed by *Bill*, *brother*, *he* are likely to be the SPEAKER, whereas nouns headed by *proposal*, or *story* are likely to be a MESSAGE.

These syntactic features are combined into a probabilistic model stating a conditional probability of assignment of the role as:

$$P(role|pt, gov, path, pos, v, hw)$$

However, due to sparseness of data, the probability is not computed using overall maximal likelihood estimation, but linearly approximated from partial conditional probabilities instead. The partial conditional probabilities are estimated using maximal likelihood, e.g.:

$$P(role|pt, t) = \frac{\#(role, pt, v)}{\#(pt, v)}$$

for probability of role assignment depending on the phrase type and voice.

Authors achieved 65 % precision and 61 % recall in the task of segmenting constituents and identifying their semantic roles. On pre-segmented constituents, they achieved accuracy of 82 %.

The authors of (0), building on the previously described word ((0)), discuss the necessity of parsing for predicate argument recognition. They argue that the *Path in the parse tree* feature in the model is most useful as a way of finding arguments in an unknown boundary condition. However, they show that omitting the feature results in only a small decrease in performance when using pre-segmented sentences.

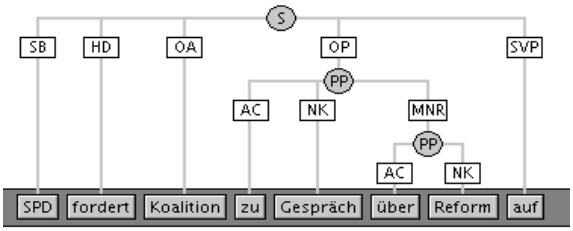
Automatic labeling (shallow semantic parsing) of the PropBank corpus is described in (0) which is again based on (0). The authors use the same set of features – path, phrase type, position, voice, head word, and add a feature predicate, and sub-categorization (for the phrase structure rule expanding the predicate's parent node in the parse tree). For training the probability, authors make use of Support Vector Machines (SVM). They train an individual SVM for each class (*Arg0*, *Arg1*, *Arg2*, *Arg3*, *Arg4*, *Arg5*, *ArgM*, *NULL*) to discriminate between this class and all the others. In a later step, they filter out overlapping combinations. They also employ verb classes to improve efficiency on unseen verbs, and name entities for constituents.[7] The authors reported 82 % precision and 73 % recall using all arguments, and 85 % precision and 77 % recall leaving out the *ArgM*s argument.

5 The SALSA Project

SALSA (Saarbrücken Lexical Semantics Annotation and Analysis)(0) developed at Saarland University creates a large annotated corpus for the frame semantics. It builds on top of the TIGER corpus [8] (0) which is a relatively flat syntactically annotated corpus of German newspaper, containing over 1.5 millions of words (80000 sentences). In the TIGER corpus, individual words are labeled with different layers of tags that include POS tags, and morphological information. The syntactic structure of sentences is described by phrase structure based trees using grammatical functions labels (e.g. *SB* for subject, *HD* for head), and syntactic categories (e.g. *S*, *NP*, *PP*). The syntactic trees allow for crossing edges in order to capture word order phenomena like long distance dependencies, right extraposition, and allow for secondary edges that mark the reuse of material in ellipses and coordinations. Figure 6 shows an example of TIGER sentence annotation.

[7] They use seven name entities: PERSON, ORGANIZATION, LOCATION, PERCENT, MONEY, TIME, and DATE; however, they do not go into detail how to assign entities to constituents.

[8] The TIGER corpus is a successor of the NEGRA corpus(0).

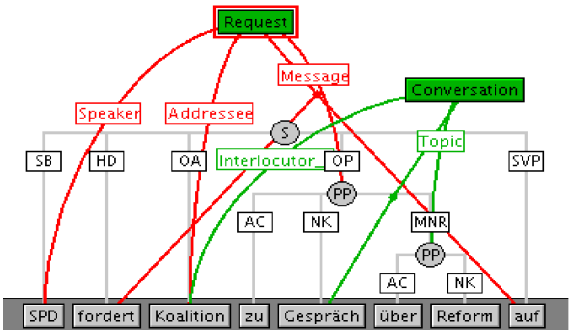


SPD request that Coalition talk about reform.

Figure 6: TIGER annotation of a sentence

On top of the TIGER treebank, SALSA adds flat semantic annotation using the FrameNet definitions (cf. 1). As the FrameNet is defined only for English, SALSA reuses as many as possible of its semantic frame description, and omits the syntactic part of the database. The semantic annotation in SALSA consists of annotation of individual frames. For each frame, the frame evoking element and some of its frame elements are associated to (TIGER) syntactic constituents.

Figure 7 shows annotation of two frames — the REQUEST frame with composed frame evoking element *fordert . . . auf* and frame elements SPEAKER (*SPD*), ADDRESSEE (*Koalition*) and MESSAGE (*zu Gespräch über Reform*), and CONVERSATION frame with frame evoking element *Gespräch* and semantic roles INTERLOCUTOR_1 (*Koalition*) and TOPIC (*über Reform*).



SPD request that Koalition talk about reform.

Figure 7: SALSA annotation on top of the TIGER annotation

In the sequel, we use the abbreviation **FEE** for *frame evoking element*, and **FE** for *frame element*. Moreover, we use the term **sub-corpus** for a set of sentences of the SALSA corpus that corresponds to one particular FEE.

The process of annotation is done FEE-wise, i.e. annotators obtain all sentences of the sub-corpus in one go. Annotators choose an appropriate frame for each frame evoking element and assign its specific frame elements that are realized in the sentence.

5.1 Underspecification

The SALSA annotation scheme allow for underspecification to represent unresolved word sense ambiguities or optionality. In a given context an FEE can evoke two different frames. For example, the verb

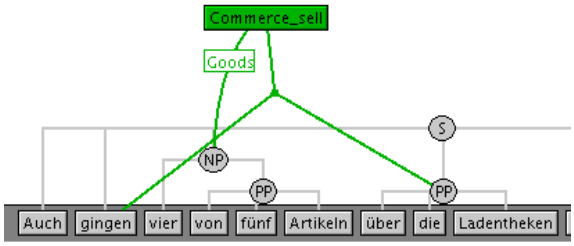
verlagen (demand) may evoke both, the REQUEST and the COMMERCIAL TRANSACTION frame. In this case the FEE is annotated with two alternative frames within one underspecification group.

An FE can also be marked as underspecified. For example, the FE *Antrag* (motion) in the REQUEST frame could have both, MEDIUM, and SPEAKER semantic roles.

Moreover, a syntactic constituent might be marked as an optional FE in case it may or need not be present in the frame.

5.2 Multipart Frame Evoking Elements and Frame Elements

In the SALSA annotation, a single FEE or FE could be composed of more words. In the case of FEE, this is used for the treatment of idiomatic or support constructions (multiword expressions). An example of the treatment of an idiomatic expression is displayed in Figure 8. Here the whole phrase *Über die Ladentheke gehen* ("go over the counter") is marked as a frame evoking element of the frame COMMERCE SELL, as its idiomatic meaning is "sell".

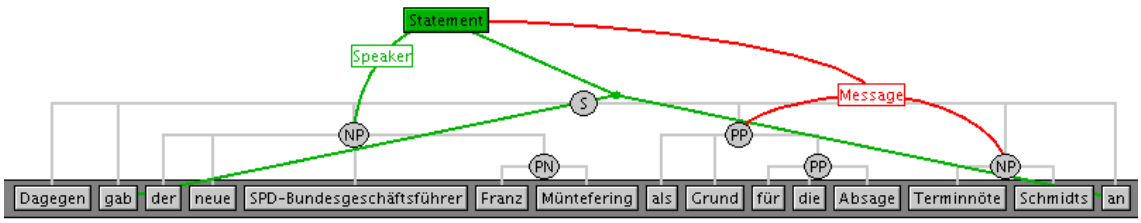


Also went four of five articles over the counter. *(literally)*

Figure 8: Example of a multipart frame evoking element

In the case of verbs with separable prefixes and reflexive verbs in the FEE position, the FEE may also consist of two parts, the main verb and the prefix (or the reflexive particle respectively), as shown in Figure 7: the FEE of the REQUEST frame consists of the main verb "*fordern*" and the prefix "*auf*".

Multipart FEs occur in cases where two (or more) distinct syntactic constituents are annotated as an instantiation of a single semantic role. In Figure 9, the PP *als Grund für Absage* (*as a reason for calling off*) and the NP *Terminnöte Schmidts* (*Schmidts' time conflict*) are both annotated as a MESSAGE of STATEMENT frame, since they jointly convey its content.



Therefore the new director . . . mentioned [. . . Schmidts' time conflicts] [as a reason for calling off (the appointment)].

Figure 9: Example of a multipart frame element

6 The LFG Parsing Architecture

Lexical Functional Grammar (LFG) ((0) and (0)) is a linguistic formalism that provides devices for describing both the common properties of nature languages and particular properties of individual languages. It assumes multiple levels of a representation of the sentence. Most prominent levels are the constituent structure (c-structure) and the functional structure (f-structure). Other levels of sentence representation are the semantic structure (s-structure), the anaphoric structure (a-structure), the discourse structure (d-structure), etc. They are, however, not all usually implemented in a grammar.

The lexical entries (stored in a lexicon) include information about the arguments of lexical items, and their grammatical functions (e.g. subject, object, adjunct).

For example, the lexical entry for the intransitive verb *sell* includes two grammatical functions, namely subject (SUBJ), and object (OBJ):

$$\text{'sell } \langle \text{SUBJ, OBJ} \rangle \text{'}$$

Moreover, the lexical entry of wordforms contain information about relevant morphological categories (e.g. person, number).

C-structure representation encodes the surface word order, and it consists of (projective) phrasal context-free trees, which are defined by context-free rules.

F-structure representation abstracts from surface word-order, and describes the predicate-argument structure of sentences. F-structures are encoded as an Attribute-value Matrix.

> **Attribute-value Matrix** (AVM)[9] is a way of describing complex linguistic structures. An AVM is a set of attributes to which values are assigned. Value could be either a canonical value, an AVM, or a set of AVMs. Formally, AVM can be represented as directed acyclic graph with labeled edges, where AVMs correspond to nodes and attributes correspond to labels of edges. The graph cannot contain directed cycles, but can contain undirected cycles. Undirected cycles indicate *reentrant* nodes (i.e. *attributes sharing the same value*).

Figure 10 shows the f-structure representation of a sentence as both, an AVM and a directed graph.

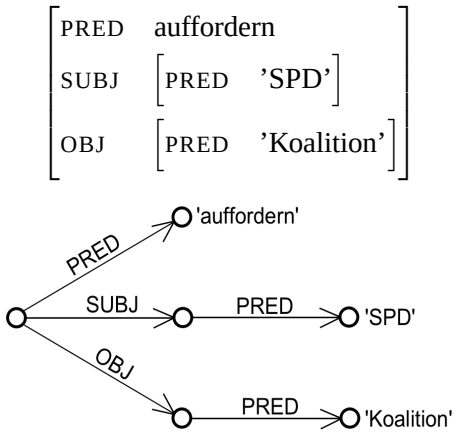


Figure 10: The f-structure displayed as an AVM and as a directed graph.

F-structures are designed using annotations accompanying the context-free (c-structure) rules. Each node of the (c-structure) context-free tree is projected to an f-structure node (via φ projection) and the

[9] For formal explanation of AVM see (0).

annotation of a context-free rule defines a relation among f-structure nodes. Similar rules could be also defined for other levels of representation (s-structure, etc.).

C-structures depend on the particular language, whereas f-structures seem to be more language-independent.

Figure 11 gives an example of context-free rules with annotation assigned to them.

$$\begin{array}{lcccc} S & \rightarrow & NP & & VP \\ & & \uparrow SUBJ = \downarrow & & \uparrow = \downarrow \end{array}$$

$$\begin{array}{lcccc} VP & \rightarrow & V & & NP \\ & & \uparrow = \downarrow & & \uparrow OBJ = \downarrow \end{array}$$

Figure 11: LFG context-free rules with functional annotation

The ↓(down arrow) symbol is a variable assigned to the f-structure node projected from the c-structure under which the equation is stated. The ↑(up arrow) symbol is a variable assigned to the f-structure node projected from the parent c-structure node written on the left side of the rule.

In the first rule, **S** is transformed to **NP** and **VP**. The equation under the **NP** (↑ SUBJ = ↓) indicates that the f-structure node (φ−) projected from the **NP** is the value of the **SUBJ** attribute of the f-structure node projected from the **S**. The ↑=↓equation beneath the **VP** (↑=↓) indicates that the **S** and **VP** nodes are both projected to the same f-structure node.

In the second rule, the **VP** and **V** are projected to the same f-structure node, whereas the **NP** is projected to the value of attribute **OBJ** of the projection of the **VP**.

The induced c-structure and f-structure of a sentence “John sees Marry” is given in Figure 12.

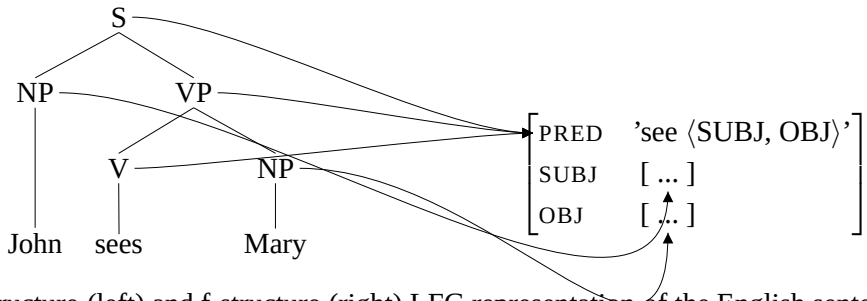


Figure 12: c-structure (left) and f-structure (right) LFG representation of the English sentence *John sees Mary*.

Part II
Automatic Frame Assignment

In this part we describe a method for automatic frame assignment using corpus based induction(0). We induce an LFG syntax-semantics interface for frame processing in a computational LFG parsing architecture.

First, we show how to model Frame Semantics in the LFG parsing architecture (sections 7–8). Second, we describe experiments we did with porting frame semantic annotation from the SALSA corpus to the LFG architecture using "parallel" LFG corpus (section 9). Third, we show extraction of rules for automatic frame assignment (section 10). Finally, we describe an application of these rules to the LFG parser output, and present the achieved results (section 10.9).

7 Modeling Frame Semantics in the LFG framework

We model the frame semantics as projection (σ_f) from the f-structure level of representation to the s-structure (semantic structure) level of representation. S-structure, much like the f-structure, is represented using attribute-value matrices (AVMs).

We define the σ_f projection to introduce a frame structure corresponding to the given sentence. AVM representing a frame contains attributes FRAME, FEE, and one attribute for each semantic role.

Figure 13 shows a semantic projection of sentence *SPD fordert Koalition zu Gespräch über Reform auf*. The verb *auffordern* (*fordert . . . auf*) maps to the (s-structure) AVM of the frame REQUEST with the attributes **FRAME** (holding the frame name) and **FEE** (holding the name of the predicate of the FEE).

The **SUBJ** (*SPD*) of the verb maps to the AVM that is the value of the frame's attribute SPEAKER, **OBJ** (*Koalition*) maps to the AVM that is the value of the frame's attribute ADDRESSEE, and the **OBJ** of **OBL** (*Gespräch über Reform*) maps to the AVM that is the value of the frame's attribute MESSAGE.

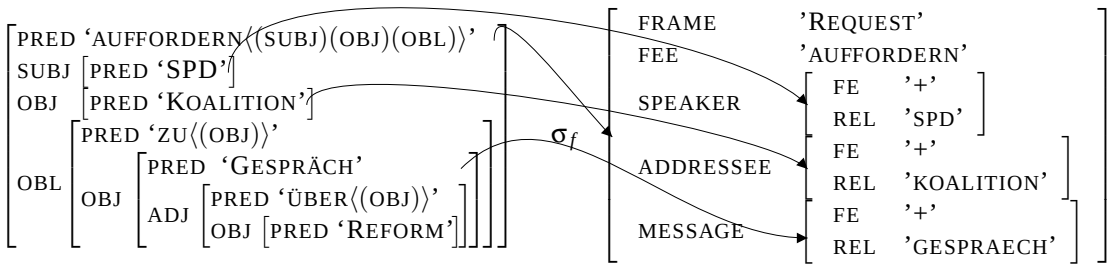


Figure 13: An example of the LFG projection architecture for Frame Annotation

Semantic structures corresponding to frame elements contain two attributes: the attribute FE with the value "+" indicating that this node is a frame element and the attribute REL with a value equal to the its predicate.

7.1 Frames in Context

The projection of frames in a context of other frames can lead to a structure of connected frames. Adding the CONVERSATION frame in the way it is used in Figure 7 to the sentence from Figure 13, we obtain a structure (displayed in Figure 14) where the value of the attribute MESSAGE of the REQUEST frame is the CONVERSATION frame, and the f-structure node corresponding to the predicate *Gespräch* is σ-projected

to it. Moreover, the value of the attribute ADDRESSEE of the frame REQUEST is identical with that of the attribute INTERLOCUTOR_1 of the frame CONVERSATION (the attributes are reentrant).

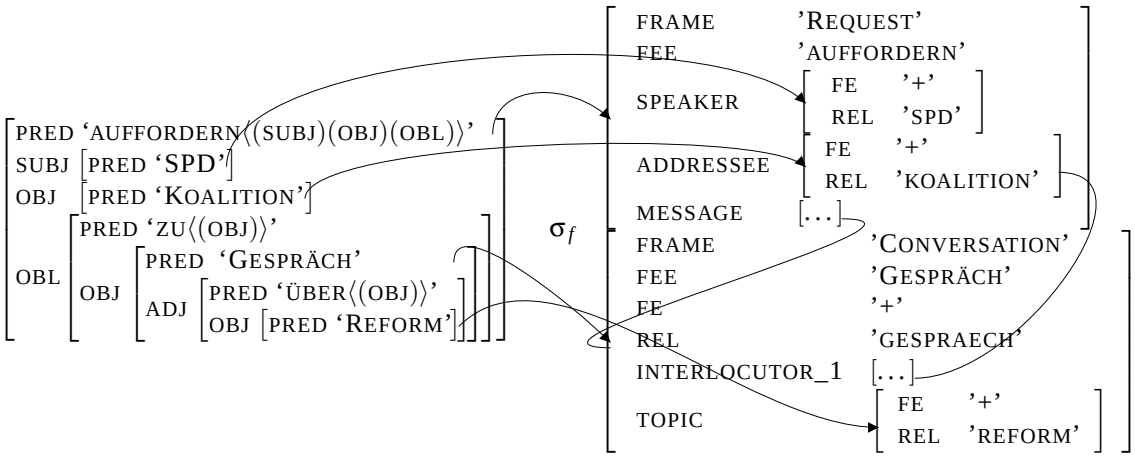
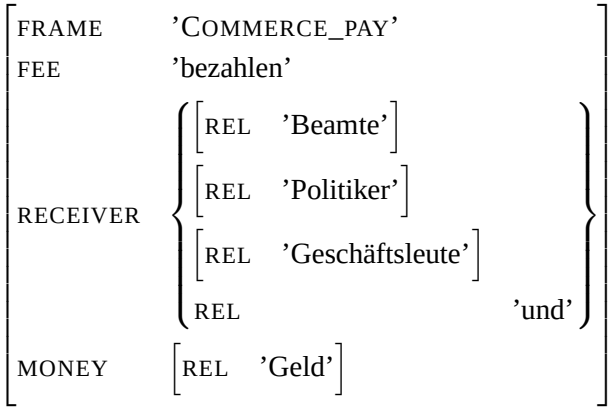


Figure 14: Frames in Context: An example of LFG projection architecture for Frame Annotation

7.2 Coordination

Frame elements that correspond to coordinated constituents need to capture all the constituents in a single frame element. We model coordination as a set in AVM. The value of coordinated frame element contain a set of AVMs, each for one coordinated element. The attribute REL of the frame element contains the coordination predicate (conjunction) in order to capture the relation between coordinated elements, and they are mapped (via σ_f) to the elements of the set.

Figure 15 shows an s-structure representation of a coordinated FE.



Beamten, Politikern und Geschäftsleuten wird Schmiergeld bezahlt
Clerks, politicians and businessmen are paid bribes

Figure 15: Frame with a coordinated RECEIVER FE

7.3 Underspecification

As already mentioned in 5.1, SALSA annotation scheme allows for underspecification. In the LFG architecture, we model underspecification as disjunction, which is encoded by optional transfer rules that create alternative (ambiguous) contexts. Optionality, introduced in 5.1, is modeled by a single optional rule.

7.4 Multipart Expressions

Treatment of multipart frame elements and frame evoking elements (cf. 5.2) require a special treatment. In the case of multipart FEEs (e.g. idiomatic expressions), we define the attribute FEE which contains the predicate of the main element of the multipart expression.[10] Moreover, we define a set attribute *FEE-MWE* which contains the elements of the multipart FEE except for the main one.

Later on, when constructing rules, we condition the rule by the existence of all the elements instead of the main FEE only.

Figure 16 shows a projection of multipart FEE corresponding to the SALSA annotation from Figure 8.

$$\begin{bmatrix} \text{FRAME} & \text{'COMMERCE\_SELL'} \\ \text{FEE} & \text{'gehen'} \\ \text{FEE-MWE} & \left\{ \begin{bmatrix} \text{REL} & \text{'über'} \end{bmatrix}, \begin{bmatrix} \text{REL} & \text{'die'} \end{bmatrix}, \begin{bmatrix} \text{REL} & \text{'Ladentheke'} \end{bmatrix} \right\} \\ \text{GOODS} & \begin{bmatrix} \text{REL} & \text{'Artikel'} \end{bmatrix} \end{bmatrix}$$

Vier Artikel gingen über die Ladentheke.
Four items were sold.

Figure 16: Multiword expressions

Otherwise, idiomatic expressions are treated in the same way as an ordinary FEEs, they receive the frame corresponding to their idiomatic meaning (e.g. *Über die Ladentheke gehen* receive frame COMMERCE_SELL).

Multipart frame elements were introduced in 5.2. Projecting two distinct constituents to a single node in s-structure can lead to inconsistencies.[11] Therefore, when modeling semantics in the LFG, multipart FEs require a special treatment. In the s-structure, *asymmetric embedding* at the semantic level is a typical pattern for such double-constituent annotation. The following sentence is an example of such double annotation:

> Der Geschäftsführer gab [$_{PP-MO}$ als Grund für die Absage] [$_{NP-OBJ}$ Terminnöte Schmidts] an.
> The director mentioned [Schmidts' time conflicts] [as a reason for calling off (the meeting)].

Such multiple-constituent annotations arise in cases where frame annotations are partial since corpus annotation is proceeds FEE-wise.[12] We account for such cases by a simulation of *functional uncertainty* equations. They use a potentially embedded anonymous frame[13] within the other one in both possible ways. Later, we apply a transfer rule that embeds one (or the other) of the two constituent projections as an unknown frame, to be evoked by the respective 'dominating' node. We don't specify the name of the inner frame, and we introduce an attribute "ROLE*" for the anonymous role of the inner frame[14].

[10] The identification of the main element might be ambiguous.

[11] E.g. when both constituents are involved in other frames as well.

[12] I.e. in the example sentence the REASON frame may not have been treated yet.

[13] Frames without a specified name and with unlabeled semantic roles.

[14] Notice that from definition of the AVM, there cannot be more attributes with the same name (ROLE*), which, however, is not the case here. If it were, we would use a set representation instead.

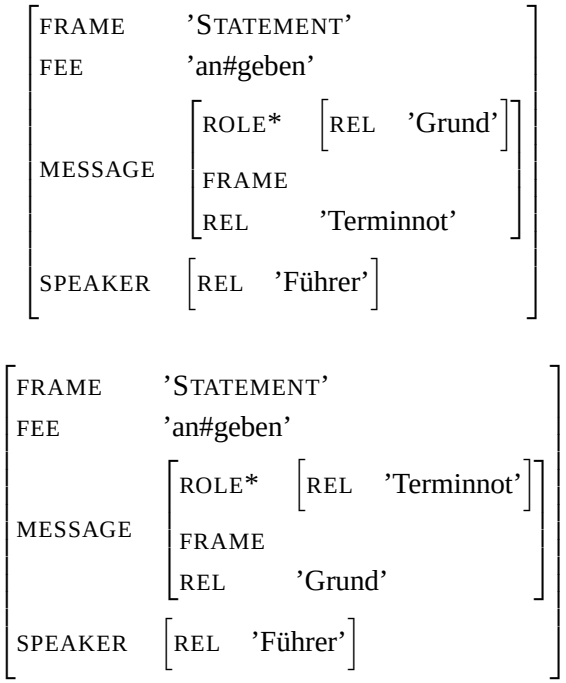


Figure 17: Multiword semantic roles
Two possible ways of constructing embedded frames structure.

Figure 17 shows alternative projections of multipart frame element for the example sentence, where the second one – with FRAME instantiated to REASON and ROLE* instantiated to CAUSE – corresponds to the actual reading. The MESSAGE of the STATEMENT frame points to the PP *as a reason for calling off*, which itself projects a frame REASON with FEs CAUSE for *Terminnöte* and EFFECT for *Absage*.

8 Definition of semantic projection

There are different approaches how s-structures can be incorporated into LFG (or into combination of c-structure and f-structure, to be exact). The most straightforward way is known as **co-description**. In the co-description, f-structure and s-structure levels are both described in the same way – in the lexicon and in annotations of the context free rules, as was described in section 6 for f-structure. They jointly determine a valid analysis of a given sentence. Analysis that do not satisfy both, f-structure and s-structure constraints, are inconsistent and they are ruled out.

An example of co-description definition for the frame from Figure 13 is stated in Figure 18 by the lexical entry of *auffordern* – the FEE for the frame REQUEST.

auffordern V,
$(\uparrow \text{PRED}) = \text{'AUFFORDERN}\langle(\uparrow \text{SUBJ})(\uparrow \text{OBJ})(\uparrow \text{OBL})\rangle\text{'}$
...
$(\sigma_f(\uparrow)\ \text{FRAME}) = \text{REQUEST}$
$(\sigma_f(\uparrow)\ \text{FEE}) = (\uparrow\ \text{PRED FN})$
$(\sigma_f(\uparrow)\ \text{SPEAKER}) = \sigma_f(\uparrow\ \text{SUBJ})$
$(\sigma_f(\uparrow)\ \text{ADDRESSEE}) = \sigma_f(\uparrow\ \text{OBJ})$
$(\sigma_f(\uparrow)\ \text{MESSAGE}) = \sigma_f(\uparrow\ \text{OBL OBJ})$

Figure 18: Lexical entry of the verb *auffordern* defining semantic projection by co-description

The σ_f is a function of f-structure nodes defining the semantic mapping to s-structure nodes. The f-structure of the verb *auffordern* (indicated by symbol $\uparrow$) projects to an s-structure node $\sigma_f(\uparrow)$ with attributes FRAME and FEE. The FEs – the attributes SPEAKER, ADDRESSEE and MESSAGE of the frame node – are defined as a σ_f-projection of the main predicate's SUBJ, OBJ and OBL OBJ functions, respectively.

An alternative to co-description is semantic construction via **description-by-analysis** (DBA) (0). In DBA, semantics is built on top of a fully resolved f-structure. DBA takes the f-structures as input (ignoring c-structure) and creates the semantics using defined rules. F-structures that are consistent with the constraints of the rule are enriched by the corresponding semantic projection, remaining f-structures are left untouched.

Both models are equally expressive – yet while co-description integrates the semantic projection into the grammar and parsing process, DBA keeps it as a separate module. For that reason, it can be developed separately from the grammar.

In our work we decided to use the DBA approach because the grammar was developed separately at the Stuttgart University. We implemented the DBA using the rewriting rules of the transfer system that is a part of the XLE[15] grammar processing platform. The system represents f-structure and s-structure (or another levels of representation) as a set of binary predicates which take variables or atomic values as arguments. Arguments stand for f-structure and s-structure nodes, and all predicates stand for AVM attributes or the projection between structures.

E.g. predicate *SUBJ(A, B)* represents the fact that *B* is the value of the SUBJ attribute of *A* which is equivalent to the following AVM:

$$\boxed{A}\left[\text{SUBJ}\quad \boxed{B}\ldots\right]$$

Transfer is defined as an ordered sequence of rules which are applied in cascade. The rule applies in all positions matching its conditions. If a rule applies to an **input set** of predicates, it defines a new **output set** of predicates. This output set is the input to the next rule in the cascade.

A rule applies if all terms (constraints) on its left-hand side match some (sub-)structure in the input. Then the terms on the right hand side are added to the input set. If the constraints do not match anything, the set of predicates remains unchanged. Once a rule is used and its output is generated, it is not used anymore in the sequence.

There are obligatory (marked by $==>$) and optional (marked by $?=>$) rules. Obligatory rules create only one output, whereas optional rules create two outputs – one is the result of application of the rule, the other is the original input set.

Figure 19 displays a transfer rule for the REQUEST frame from Figure 13, which corresponds to the co-description lexical entry in Figure 18.[16]

Arguments with the first letter in lower case are constants, whereas arguments with the first letter in capital are variables. An f-structure node matches the left-side constraints if it has an attribute PRED equal to *auffordern*, and it has attributes SUBJ, OBJ and OBL OBJ presented. For such an f-structure node, the rule defines a new semantic projection (σ_f) to s-structure representation of the frame with the frame information (attributes FRAME and FEE) and attributes for FEs projected from the appropriate f-structure nodes.

[15] Xerox Language Environment, http://www2.parc.com/istl/groups/nltt/xle/

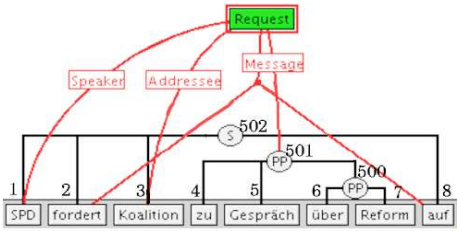
[16] 's::' is the XLE notation for σ-projection from the f-structure to the s-structure

```
pred(X,auffordern),
subj(X, A), obj(X, B), obl(X, C), obj(C, D)
==>
+'s::'(X, SemX), +frame(SemX, request),
                      +fee(X,auffordern),
+'s::'(A, SemA), +speaker(SemX, SemA),
+'s::'(B, SemB), +addressee(SemX, SemB),
+'s::'(D, SemD), +message(SemX, SemD).
```

Figure 19: Transfer rule for frame projection by DBA

9 Porting SALSA frame annotation to LFG

As a source of the semantic annotation for our work we use the annotation from the SALSA project (0), made on top of the syntactic TIGER corpus (0), which are encoded in an XML format (TIGER/SALSA XML) that extends the TIGER XML annotation scheme. Because we were using the XLE as the platform, we needed to port the semantic annotation to the XLE in the first step.



SPD requests that coalition talk about reform.

Figure 20: SALSA/TIGER representation of the sentence

The TIGER treebank had been converted to parallel LFG f-structure corpus (0) – LFG-TIGER corpus. For porting the semantics, we made use of the fact that the LFG-TIGER corpus preserves the original TIGER constituent identifiers[17], as f-structure feature named **TI-ID**. Figure 21 shows the LFG-TIGER representation of the sentence displayed in Figure 20 (with added constituent identifiers).

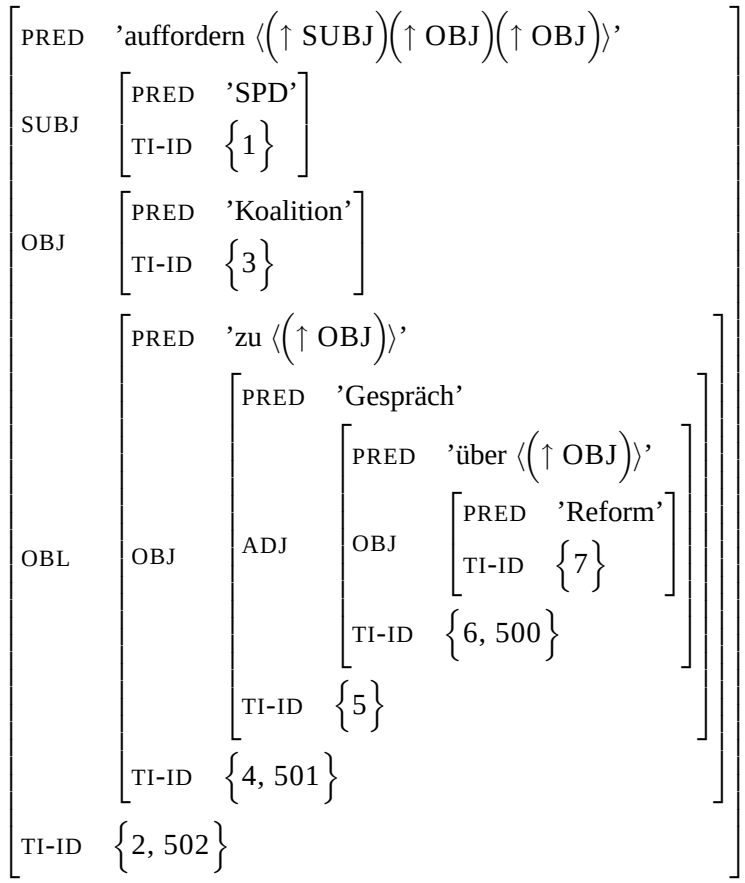
We used TI-ID attributes as anchors in transfer rules when porting the SALSA semantic annotation to the LFG-TIGER treebank in order to obtain an LFG corpus enriched by the semantic projection, as is displayed in Figure 22.

To implement the porting, we used the description-by-analysis (cf. section 8) via the XLE transfer system.

We extracted values of identifiers of the FEE and all FEs for each frame from the TIGER/SALSA annotation. By virtue of them we created transfer rules. Figure 23 shows identifiers extracted from the example sentence.

We generated a set of rules for the XLE transfer system for each frame annotation (one sentence could contain more frame annotations). The set of rules contains one rule for transferring the FEE (conditioned by the lexical value of its predicate), and five rules for each FE – one rule to assign a semantic projection to an f-structure node which does not have any, one rule to assign a semantic projection to an f-structure with already defined σ_f projection, and three rules to handle coordination. Moreover, there are more complex sets of rules that come into force in case of underspecification (cf. 5.1).

[17] Identifiers from 1 to 499 are assigned to terminals, identifiers from 500 on are assigned to nonterminals.



SPD requests that coalition talk about reform.

Figure 21: LFG representation of the sentence

We do not go into more details about implementing the rules in this work, and show only a model rule[18] in Figure 24.

We generated groups of rules for each frame separately and applied them to the f-structure of corresponding sentences in the LFG-TIGER corpus, so we obtained an LFG corpus with semantic annotation (in the way of Figure 22).

9.1 Overview of Data

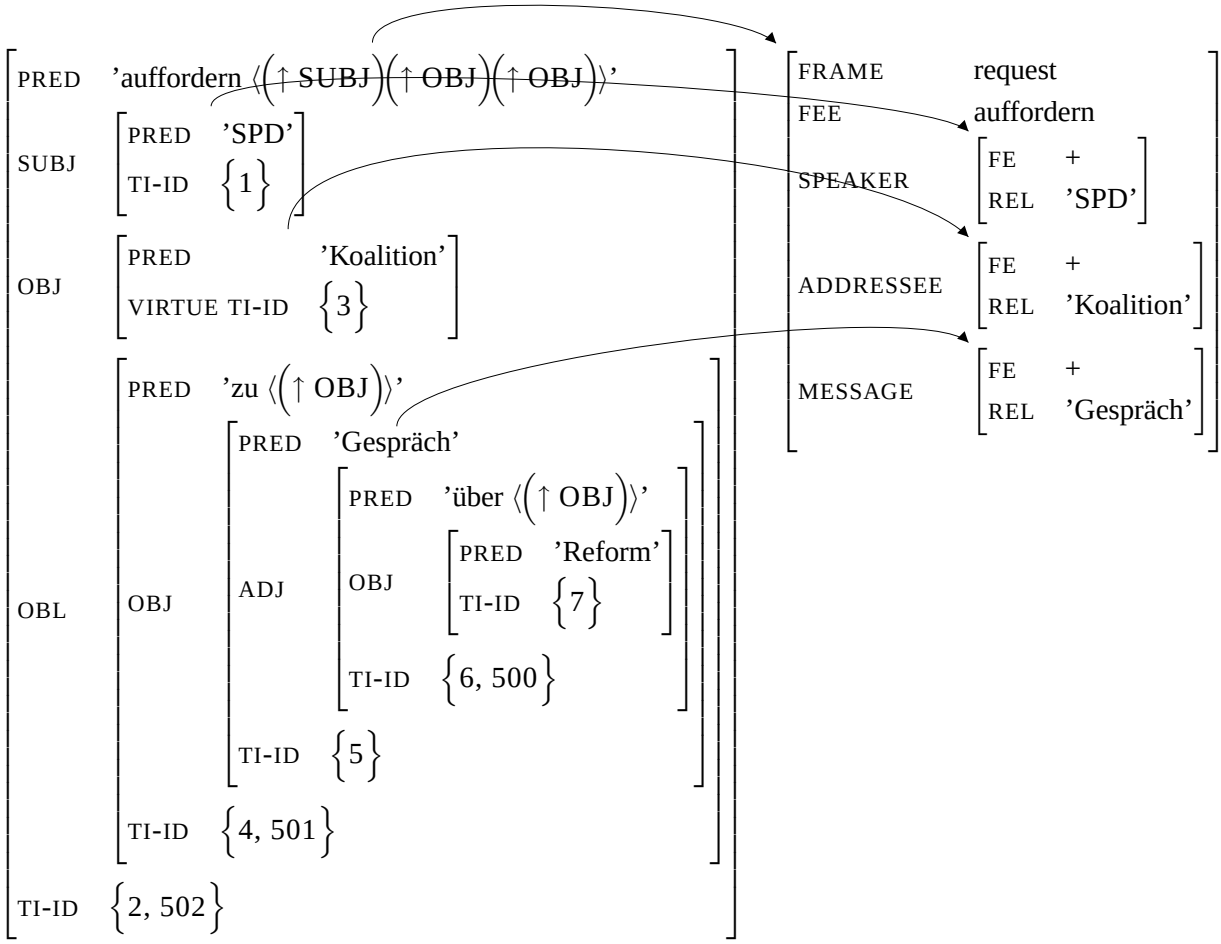
This paragraph summarizes results of the application of the rules.

We transformed 11 934 sentences for 342 different FEEs with 12 436 frames annotated. Table 1 gives the number of special phenomena we had to deal with in the data. The numbers denote number of sentences in which a given phenomenon occurred.

10 Extraction of Rules for Automatic Frame Assignment

In this section we describe the process of the extraction of lexical frame assignment rules from the semantically enriched LFG-TIGER corpus. Instead of anchoring those rules to the TIGER identifiers as in previous stage, we used here f-structure paths to identify constituents, and map them to frame

[18] For the sake of legibility we leave out all the technical details handling special linguistic phenomena described in previous chapters.



SPD requests that coalition talk about reform.

Figure 22: LFG representation of the sentence with semantic annotation

Frame:	REQUEST
FEE:	{2, 8}
SPEAKER	1
MESSAGE	3
ADDRESSEE	501

Figure 23: Constituent information extracted from the SALSA annotations

elements. These rules are independent of the particular sentence, and can be used to the f-structure output of new data (output of syntactic LFG parser) later.

10.1 LFG paths

In terms of directed graph (c.f. section 6), we can describe relation of two f-structure nodes by a path in the graph (an **f-structure path**). In the f-structure path, the symbol ↑ (up-arrow) denotes the "point of origin". Symbols on the right side of the ↑ denote the sequence of the edges (i.e. attributes) to be followed in the graph.

```
% projection of frame evoking element
ti-id(X, 2), pred(X, X_pred) ==>
's::'(X, Sem_X), frame(Sem_X, 'request'), fee(Sem_X, X_pred).

% projection on the SPEAKER
ti-id(X, 2), 's::'(X, Sem_X), frame(Sem_X, 'request'), ti-id(Y, 1),
pred(Y, X_pred) ==> 's::'(Y, Sem_Y), speaker(Sem_X, Sem_Y).

% projection on the ADDRESSEE
ti-id(X, 2), 's::'(X, Sem_X), frame(Sem_X, 'request'), ti-id(Y, 3),
pred(Y, Y_pred) ==> 's::'(Y, Sem_Y), addressee(Sem_X, Sem_Y).

% projection on the MESSAGE
ti-id(X, 2), 's::'(X, Sem_X), frame(Sem_X, 'request'), ti-id(Y, 501),
pred(Y, Y_pred) ==> 's::'(Y, Sem_Y), message(Sem_X, Sem_Y).
```

Figure 24: Simplified LFG transfer rules for porting the REQUEST frame from the SALSA annotation to the LFG-TIGER corpus

Frames:	**12436 (100 %)**
Coordination:	467 (3.76 %)
Underspecification:	395 (3.18 %)
Multiword FEE:	1287 (10.34 %)
Asymmetric embedding:	421 (3.39 %)

Table 1: Overview of special annotation types

In Figure 25[19], going from the outmost AVM signed as 1, the path *(↑ OBL OBJ)* represents the node 2. Thus attribute names could be used as postfix left-associative unary operators transforming a node to another node (which is accessible from the first one by a single edge labeled by the name of the predicate).

We use also prefix variants of the unary right-associative operator with higher priority, which follow the edges in the graph in the reverse (backward) way. We refer to this notation as *inside-out* paths. Then going from 3, the path *(OBL ↑)* leads to node 1, and the path *((OBL ↑) OBJ)* leads to 4.

We call a path containing an inside-out subpart **non-local path**.[20] A path that does not contain any inside-out subpart is called **local path**.

10.2 Algorithm for Path Extraction

In this step, we created general rules, which were later applied to f-structures of new sentences. Those rule are anchored to the occurrence of the predicate of the FEE, and to the existence of f-structure paths of FEs relatively to the FEE. We designed a simple algorithm for extracting f-structure paths between f-structure nodes. The algorithm searches paths between two f-structure nodes (from *source node* to *destination node*), preferring local paths to non-local ones. Moreover, all inside-out (non-local) steps

[19] We assume here that ADJ contains a single AVM instead of a set.

[20] As it can lead to a node that is not "in the scope" of the source node (i.e. within the origin AVM).

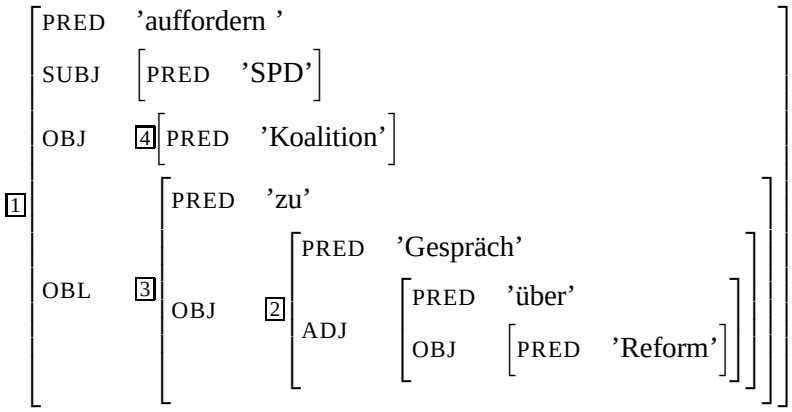


Figure 25: Simplified example of paths representation in f-structure

should precede all outside-in (local) steps. The algorithm consists of subsequent calls of **Depth-first Search** algorithm (DFS).[21]

The process starts from the *source node* (X_0) and tries to find the *destination node* locally using DFS for searching accessibility in directed acyclic graphs. There could be none or more different paths found. If there is any, the algorithm returns all of them and stops.

If there is none local path, the algorithm goes to all directly governing nodes[22] (X_1), and searches locally from them using DFS. If it finds the destination node in the X_1-context locally, it returns all of the found paths and stops. Such path (leading from the *source node* to the *destination node* via X_1) consists of exactly one inside-out step, which is the first step in the path, and a sequence of zero or more outside-in steps.

Until there is no paths found, the algorithms goes repeatedly one step outside in the terms of inside-out path and searches locally from the new point – the algorithm stops, when the destination node is found, or when it reaches the most outer f-structure and does not find the destination node there (in case the destination node is not present or the AVM is discontinuous).

The algorithm always stops, as the graph of AVM is acyclic.[23] The paths always contain first the inside-out (non-local) part, and then the outside-in (local) part. If there is some path leading from the *source node* to the *destination node*, the algorithm finds it. Proofs of the mentioned features of the algorithm are obvious.

In the example sentence showed in Figure 25, if the algorithm searches a path from 2 to 4, it first searches local content of 2, then local context of 3, and only in the local context of 1 it finds the node 4. The result path is *((OBL OBJ ↑)OBJ)*. For implementation reasons we used also "left-to-right" notation of the path where the sign "–" denotes an inside-out step and the sign "/" delimits individual steps:

–OBJ/–OBL/OBJ

10.3 Building Frame Assignment Rules

From the semantically enriched TIGER-LFG corpus we extracted path information for each frame. For the sentence from Figure 14 we extracted paths displayed in Table 2 for the REQUEST frame and paths displayed in Table 3 for the CONVERSATION frame.

[21] Cf. http://en.wikipedia.org/wiki/Depth_first_search for details.

[22] Due to reentrance, there could be more directly governing nodes.

[23] It follows only a limited number of backward links.

Frame element	Path in LFG notation	Path in "left-to-right" notation
SPEAKER	↑ SUBJ	SUBJ
ADDRESSEE	↑ OBJ	OBJ
MESSAGE	↑ OBL OBJ	OBL/OBJ

Table 2: Paths for the REQUEST frame

Frame element	Path in LFG notation	Path in "left-to-right" notation
INTERLOCUTOR_1	(OBL OBJ ↑) OBJ	–OBJ/–OBL/OBJ
TOPIC	↑ ADJ OBJ	ADJ/OBJ

Table 3: Paths for the CONVERSATION frame

We conditioned each rule by an existence of the FEE's predicate and by an existence of the f-structure nodes within the FE-paths relative to the FEE f-structure node. Rules are realized by rules of the transfer system that is a part of the XLE grammar processing platform (see section 8 for details).

We constructed two types of rules. In the first one, the **whole-frame rules**, we constructed exactly one rule for each frame in the corpus, which adds semantic projection of the whole frame as it is instantiated in the corpus. However, due to sparse data, we can find a lot of frame configurations in the real data that have not been seen in the training data. Therefore, we defined an alternative rule format, the **partial-frame rules**, in which we splitted frame assignment into separate rules for projection of the FEE and the individual FEs. This step allows semantic projection to be added even in cases where the f-structure does not satisfy the functional constraints for all FEs. It can improve robustness and account for syntactic variability when applied to new data.

Figure 26 displays an example of the whole-frame rules for the CONVERSATION frame from sentence in Figure 14, and Figure 27 displays example of the partial-frame rules for the same frame.

```
+pred(X, 'Gespräch'),
+obj(Y1_1, X), +obl(Y1_2, Y1_1), +obj(Y1_2, Y2)
+adj(X, Y2_1), +obj(Y2_1, Y2)
==>
's::'(X, Sem_X), fee(Sem_X, 'Gespräch'), frame(Sem_X, 'Conversation'),
's::'(Y1, Sem_Y1), 'INTERLOCUTOR_1'(Sem_X, Sem_Y1),
's::'(Y2, Sem_Y2), 'TOPIC'(Sem_X, Sem_Y2).
```

Figure 26: Whole-frame variant of frame assignment rule for the CONVERSATION frame.

The left-hand side of a rule (in front of "==>") defines condition and the right-hand side of the rule describes effect of the rules (added predicates). The "+" sign in the left-hand side of the predicate indicates that the predicate is not deleted from the f-structure.

10.4 Identifiers

In adition to the frame assignment rule we created a unique identifier for each frame annotation, which fully described the rule. We used those identifiers later for evaluation. The identifier begins with the percentage sign (**%**)[24] followed by a single space, and then several items separated by semicolon. The first item is a keyword **IDENT**, the second one is the name of the frame, and the third one is the lemma of the FEE's predicate. Two items for each FE follow – the FE's name and its relative path leading from

[24] **%** sign introduces comments in the XLE transfer system, therefore the identifiers can be added to the code of the rules without influence on its behavior.

+pred(X, ’Gespräch’), ==> ’s::’(X, Sem_X), fee(Sem_X, ’Gespräch’), frame(Sem_X, ’Conversation’).
+pred(X, ’Gespräch’), +’s::’(X, Sem_X), +frame(Sem_X, ’Conversation’), +obj(Y1_1, X), +obl(Y1_2, Y1_1), +obj(Y1_2, Y2) ==> ’s::’(Y1, Sem_Y1), ’INTERLOCUTOR_1’(Sem_X, Sem_Y1).
+pred(X, ’Gespräch’), +’s::’(X, Sem_X), +frame(Sem_X, ’Conversation’), +adj(X, Y1_1), +obj(Y1_1, Y1) ==> ’s::’(Y1, Sem_Y1), ’TOPIC’(Sem_X, Sem_Y1).

Figure 27: Partial-frame variant frame assignment rules for the CONVERSATION frame

the FEE. As there is no implicit order of the frame elements (AVMs are sets of attribute-value pairs), frame elements are sorted alphabetically according to their names.

The path consists of steps in the left-to-right notation, where the sign “$” denotes access of an element of the set ($\in$).

The identifier corresponding to the rule in Figure 26 is as follows:

```
% IDENT;Conversation;Gespräch;INTERLOCUTOR_1;-OBJ/-OBL/OBJ;TOPIC;ADJ/OBJ
```

The identifier corresponding to the rule for the REQUEST frame in Table 2 is as follows:

```
% IDENT;Request;auf#fordern;ADDRESSEE;OBJ;MESSAGE;OBL/OBJ;SPEAKER;SUBJ
```

10.5 Special phenomena

We added a special rule template for projecting coordination. When a path leads to coordinated constituents, each of them is projected to a semantic node and included into a set corresponding to the FE.

Multipart frame evoking elements also required a special treatment. All parts of the multipart element must have occurred in the sentence to let the rule apply.

For the f-structure containing a multipart FEE (Figure 28), the whole-frame variant of the rule will look as Figure 29 shows – predicate **in_set** indicates an element of a set (*in_set(A,B)* means $A \in B$). Then, it is projected to the structure shown in Figure 16.

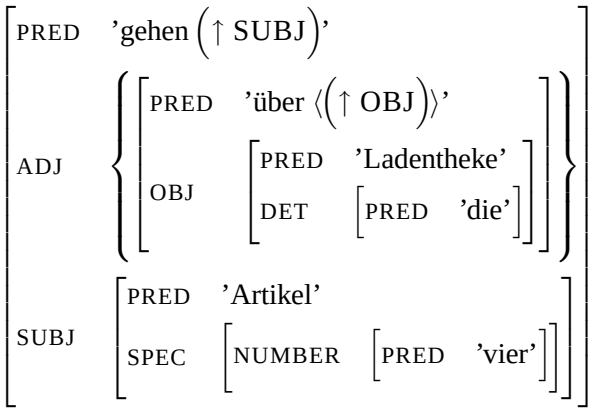
10.6 Adjunct Specification

When FE is in the position of ADJunct and OBLique, we project an element of a set. The fact that the set can contain more than one element increases the level of ambiguity in the output, as the rule applies for all predicates.[25]

For each ADJunct and OBLique, we conditioned the rule with its predicate. This additional condition is based on the assumption that semantic roles are usually bounded to a certain preposition.

For example, in Figure 30, if a frame element were assigned to the predicate “Gespräch”, we would condition the predicate of the element of the set with the value “*zu*”. The important part of the generated rule is shown in Figure 31

[25] Or for all possible combinations of predicates when more FEs are involved in the same set.



Vier Artikel ging über die Ladentheke.

Figure 28: F-structure of the sentence with a multipart frame evoking element

```
+pred(X, 'gehen'),
+adjunct(X, Z0_1), +in_set(Z0_2, Z0_1), +pred(Z0_2, 'über'),
+adjunct(X, Z1_1), +in_set(Z1_2, Z1_1), +obj(Z1_2, Z1_3),
        +spec(Z1_3, Z1_4), +det(Z1_4, Z1_5), +pred(Z1_5, 'die'),
+adjunct(X, Z2_1), +in_set(Z2_2, Z2_1), +obj(Z2_2, Z2_3), +pred(Z2_3, 'Ladentheke'),
+subj(X, Y0_1)
==>
's::'(X, Sem_X), fee(Sem_X, gehen), frame(Sem_X, 'Commerce_sell'),
fee_mwe(Sem_X, FEEMWE),
's::'(Z0_2, Sem_Z0), in_set(Sem_Z0, FEEMWE),
's::'(Z1_5, Sem_Z1), in_set(Sem_Z1, FEEMWE),
's::'(Z2_3, Sem_Z2), in_set(Sem_Z2, FEEMWE),
's::'(Y0_1, Sem_Y0), 'GOODS'(Sem_X, Sem_Y0), fe(Sem_Y0, +).
```

Figure 29: Frame assignment rules for a frame with multipart FEE (whole-frame variant)

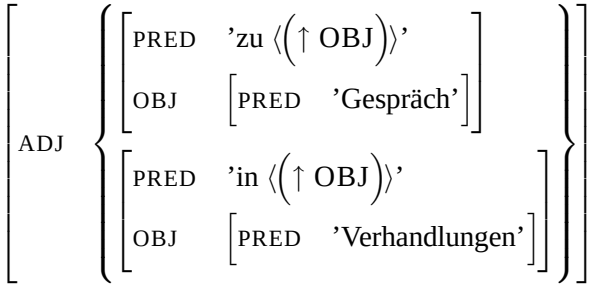


Figure 30: Adjunct specification f-structure

As for the identifier, we encoded the adjunct specification in the path of the FE using **@** (“at" sign) as delimiter:

```
% IDENT;...;ROLE;ADJUNCT/$@zu/OBJ;...
```

```
...
+adjunct(X, Y0_1), +in_set(Y0_2, Y0_1), +pred(Y0_2, 'zu'), +obj(Y0_2, Y0_3),
...
==>
...
's::'(Y0_3, Sem_Y0), 'ROLE'(Sem_X, Sem_Y0),
...
```

Figure 31: Adjunct specification rule

10.7 Compilation

In the compilation process, we unified all the trained rules of one FEE deleting duplicities, occurring several times in the training data. We exported rules into sets of rules (files in the format of XLE transfer system) with the same FEEs. Moreover we exported a big set of rules containing all rules – this set could be later applied to f-structures of new sentences.

10.8 Overview of Data

We extracted rules for frame assignment from the semantically enriched LFG-TIGER corpus. We compiled 9707 lexical frame assignment rules in the format of the XLE transfer system. The average number of distinct rules per one FEE was 8.83. Abstracting over FEEs, we obtained 7317 FRAME-specific rules with an average of 41.34 distinct rules per a frame.

Among the rules extracted from the enriched LFG corpus, 12.82 % were non-local (i.e. contained some non-local path) and 87.18 % were local.

For the partial-frame rules, we obtained 960 FEE assignment rules, and 8261 FEE-specific FE assignment rules. Abstracting over the FEE, this reduces to 4804 rules.

10.9 Evaluation

To check the quality of generated rules, we reapplied the induced frame assignment rules to the original LFG-TIGER corpus(0) and evaluated the generated frame annotations against the semantically enriched corpus (cf. 9).

We extracted the rule identifiers of the rules (cf. 10.4) for annotation of both, the **whole-frame rules**, and the **partial-frame rules** (cf. 10.3). Because of the reentrance, more rules could have been obtained from one frame.

We obtained 93.98 % recall and 25.95 % precision for the whole-frame rules, and 94.98 % recall and 45.52 % precision for the partial-frame rules. In average, there were 8.46, resp. 7.83 applied rules (assigned frames) per annotation instance (ambiguity in the output). Table 4 summarizes our results.

	Whole-frame rules	Partial-frame rules
Precision	25.95 %	45.52 %
Recall	93.98 %	94.98 %
Frames per annotation instance	8.46	7.83

Table 4: Results of the frame assignment process on the TIGER-LFG corpus

Finally, we applied the frame assignment rules to the original LFG parses obtained from the German LFG grammar developed in the ParGram project.[26] The grammar produces f-structures that are compatible with the LFG-TIGER corpus to a certain extent, thus the syntactic constraints of the frame annotation rules could match the f-structure output of the parser. In contrast to the LFG-TIGER treebank, the grammar delivers f-structure for alternative syntactic analysis. We do not expect frame projections for all syntactic readings, but where rules apply, they create ambiguity in the semantics projection.

We applied the rules to the parses of 6032 corpus sentences. Compared to the LFG-TIGER corpus, we obtained lower recall and precision for both types of rules — 52.21 % recall and 6.93 % precision and 76.41 % recall and 18.32 % precision, respectively. In average, there were 13.35, resp. 9.00 applied rules per annotation instance (ambiguity in the output). Table 5 summarizes our results.

	Whole-frame rules	Partial-frame rules
Precision	6.93 %	18.32 %
Recall	52.21 %	76.41 %
Frames per annotation instance	13.35	9.00

Table 5: Results of the frame assignment process on LFG parses

Teh drop in the precision and the higher number of ambiguity rate might be due to the higher ambiguity in the input. Moreover, in the second experiment we applied the complete rule set to sentences. Thus the rules could have applied to unannotated instances, and therefore create more ambiguities. The drop in recall is mainly due to overgeneration in automatic lemmatisation of the LFG parser and overgeneration in functional assignments to PPs in the LFG-TIGER corpus[27], which are not all matched in the LFG parser output. Another explanation of the worse result of the second experiment can be that the f-structures of the LFG-TIGER corpus and the output of the parser are not fully compatible.

The relatively low precision in both data sets and the high ambiguity rate could be explained by the lack of statistical disambiguation of the results. For comparison, Gildea and Jurafsky in (0) achieve 65 % precision and 61 % recall making use of statistical methods for selecting only one alignment.

11 Summary and Future Work

We presented a method for a corpus-based induction of an LFG syntax-semantics interface for frame semantic processing. We transferred frame annotations from a manually annotated syntactic corpus to an LFG parsing architecture that allows processing of unparsed text. We showed how to model frame semantic annotations in an LFG projection architecture, including special phenomena that involve non-isomorphic mapping between two levels of representations.

As the semantic corpus is under construction, our results are restricted. Yet, we gave an exemplification for how to build a uniform computational semantics interface for frame assignment that can be used to process parsed corpora.

In future steps statistical disambiguation of the assigned semantic structure can be employed, and the rules could be applied on testing data in order to achieve results comparable to other ongoing works. This step is currently under development at Saarland University.

[26] http://www2.parc.com/istl/groups/nltt/pargram/

[27] PPs are assigned to ambiguous LFG attribute – e.g. ADJ, ADJ-LOC, ADJ-DIR, ...

We will use the experience acquired in this work also in our current work on the Prague Dependency Treebank, where we are aiming at an automatic assignment of tectogrammatical annotations based on resolved analytical structure. In the PDT, analytical layer corresponds to some extent to the LFG f-structures (reflects shallow syntactic structures), whereas tectogrammatical layer corresponds to the LFG s-structure (as it reflects the deep syntactic structure). We are also working on automatic disambiguation of verb frames on the tectogrammatical layer in which we can use similar verb characteristics as those used in the LFG frame assignment rules.

References

S. Brants, S. Dipper, S. Hansen, W. Lezius, and G. Smith. The TIGER Treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, Sozopol, Bulgaria, 2002.

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

M. Collins. Three Generative, Lexicalised Models for Statistical Parsing. In *Proceedings of the ACL '97*, pages 16–23, 1997.

K. Erk, A. Kowalski, S. Padó, and M. Pinkal. Towards a Resource for Lexical Semantics: A Large German Corpus with Extensive Semantic Annotation. In *Proceedings of the ACL '03*, pages 537–544, Sapporo, Japan, 2003.

C. J. Fillmore. Frame Semantics and the Nature of Language. In *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech*, volume 280, pages 20–32, 1976.

M. Forst. Treebank Conversion – Establishing a Testsuite for a Broad-coverage LFG from the TIGER Treebank. In A. Abeillé, S. Hansen, and H. Uszkoreit, editors, *Proceedings of the 4th International Workshop on Linguistically Interpreted Corpora (LINC '03)*, Budapest, Hungary, 2003.

A. Frank and J. Semecký. Corpus-based Induction of an LFG Syntax-Semantics Interface for Frame Semantic Processing. In *Proceedings of the 5th International Conference on Linguistically Interpreted Corpora, LINC '04*, Geneva, Switzerland, 2004.

D. Gildea and D. Jurafsky. Automatic Labeling of Semantic Roles. *Computational Linguistics*, 28(3):245–288, 2002.

D. Gildea and M. Palmer. The Necessity of Parsing for Predicate Argument Recognition. In *Proceedings of ACL'02*, Philadelphia, PA, 2002.

J. Hajič. Tectogrammatical Representation: Towards a Minimal Transfer in Machine Translation. In *6th International Workshop on Tree Adjoining Grammars and Related Frameworks*, pages 216–226, 20th–23rd May 2002.

P.-K. Halvorsen and R. M. Kaplan. Projections and Semantic Description in Lexical-Functional Grammar. In M. Dalrymple, R.M. Kaplan, J.T. Maxwell, and A. Zaenen, editors, *Formal Issues in Lexical-Functional Grammar*, pages 279–292. CSLI Lecture Notes, No.47, 1995.

R. Kaplan. The Formal Architecture of Lexical-Functional Grammar. In *Proceedings of ROCLING II*, pages 3–18, Taipei, Republic of China, 1989.

P. Kingsbury, M. Palmer, and M. Marcus. Adding Semantic Annotation to the Penn TreeBank. In *Proceedings of the HLT Conference '02*, San Diego, 2002.

J. Panevová. Valency Frames: Extension and Reexamination. In *Festschrift fuer Andrzej Boguslawski (eds. V. S. Chrakovskij, M. Grochowski, G. Hentschel), Studia Slavica Oldenburgensia 9*, pages 325–340. Bibliotheks- und Informationssystem, Oldenburg, 2001.

Sameer Pradhan, Kadri Hacioglu, Wayne Ward, James H. Martin, and Daniel Jurafsky. Semantic role parsing: Adding semantic structure to unstructured text. In *ICDM '03*, pages 629–632, Melbourne, Florida, USA, November 19–22 2003.

P. Sgall, E. Hajičová, and J. Panevová. The Meaning of the Sentence in its Semantic and Pragmatic Aspects. *Academia, Prague, Czech Republic/Reidel Publishing Company, Dordrecht, Netherlands*, 1986.

W. Skut, T. Brants, and H. Uszkoreit. A Linguistically Interpreted Corpus of German Newspaper Text. In *proceedings of the 10th European Summer School in Logic, Language and Information (ESSLLI'98). Workshop on Recent Advances in Corpus Annotation, August 17-28*, Saarbrücken, Germany, 1998.

Z. Žabokrtský. Automatic Functor Assignment in the Prague Dependency Treebank. In *TSD '00, Proceedings (eds. P. Sojka, I. Kopeček, K. Pala)*, pages 45–50. Lecture Notes in Artificial Intelligence vol. 1902, Springer, 2000.

Z. Žabokrtský and M. Lopatková. Valency Frames of Czech Verbs in VALLEX 1.0. In *HLT-NAACL '04 Workshop: Frontiers in Corpus Annotation*, pages 70–77. Association for Computational Linguistics, May 2–7 2004.