From speech signal to syntactic structure: A computational implementation

Tina Bögel and Tianyi Zhao
University of Konstanz

ABSTRACT

This paper presents a new computational implementation bridging several modules of grammar from phonetics to phonology to syntax. The system takes as input a speech signal annotated with syllables, interprets the phonetic data in phonological/prosodic terms, matches the data against a lexicon and makes the results available to a linguistically deep computational grammar. The system is showcased by means of syntactically ambiguous structures in German which can be disambiguated based on prosodic constituency information. A system evaluation with the German data showed good results for this new combination of automatic speech signal analysis and computational grammars, which takes a significant step towards a linguistically fine-grained computational analysis and hence towards real automatic speech understanding.

Keywords: Automatic speech understanding, syntactic ambiguities, prosodic disambiguation, LFG, German

INTRODUCTION 1

Spoken language is notoriously difficult for linguistic analyses in general and for computational implementations in particular. Various acoustic features such as duration, pitch contours, or voice quality contribute to the overall interpretation of an utterance, but are gradient in nature and subject to variation between and within speakers. This

makes it very challenging for computationally deep linguistic grammars to use information signalled by prosodic structure. As a consequence, linguistically relevant information is often lost during analysis. Consider, for example, the following statement with contrastive focus on *red*.

(1) Amra ate the RED apple.

The contrastive focus in example (1) can be acoustically signalled by a strong tonal accent with a steep rise on *red* (e.g., Xu and Xu 2005; Gussenhoven 2008) which also has implications for the meaning interpretation of the clause: Not only did Amra eat a red apple, but she ate (for example) neither the green nor the yellow apple. These types of foci often correct wrong assumptions in the interlocutors' common ground and are thus highly relevant for analyses concerned with discourse or information structure (Krifka 2008; Rooth 2016).

Another common issue is the determination of prosodic constituency in the context of syntactic ambiguities as in example (2) where *flat* can be either associated with the preceding phrase (2a) or the following phrase (2b).

(2) a. When the cake was dropped flat || plants stuck to its underside.
b. When the cake was dropped || flat plants stuck to its underside.

There are two possible syntactic analyses: a resultative structure as in example (2a) (... *drop the cake flat* ...), or a modifying structure as in example (2b) (... *flat plants* ...). Depending on whether the prosodic phrase boundary (||) precedes or follows the adjective *flat*, one of the interpretations becomes more likely (Bögel and Turk 2019). Such structures frequently appear in a variety of languages and it has been shown that many can be disambiguated by prosody (Lehiste *et al.* 1976; Price *et al.* 1991). Consequently, access to this information prevents overgeneration and supports meaning interpretation.

These are just two cases where prosodic information plays a crucial role in linguistic analyses, but numerous other examples can be found in a variety of linguistic structures across languages, e.g., the distinction between polar and constituent questions in Urdu by means of tonal accents (Butt *et al.* 2020), the second position placement of oblique pronoun clitics in Vafsi (Bögel *et al.* 2018), or the signalling

of a rhetorical question by means of pitch contour, constituent duration, and voice quality in German (Braun *et al.* 2019). This shows that access to information from the speech signal, e.g., concerning pitch distribution and prosodic constituency, benefits speech recognition and interpretation and is thus very desirable for linguistically deep computational grammars.

However, an integration of prosodic information with existing grammars is rarely pursued, although several approaches supporting automatic speech recognition and the determination of prosodic events are available and are widely used in phonetic and prosodic research. The Munich automatic segmentation system MAUS (Kisler *et al.* 2017; Schiel 1999), for example, is frequently utilized to automatically annotate segments and words in more than 20 languages such as English, German, French, and Finnish, but does not include the calculation of pitch accents or prosodic constituency. By contrast, ProsodyPro (Xu 2013) is used to analyze speech prosody with both discrete and continuous data as output, with a focus on time-normalized pitch contours and F_0 velocity. F_0 contours and other acoustic cues can be averaged across repetitions and speakers, which enables a direct statistical comparison. However, the system does not provide any categorical information, e.g., in terms of accents, and calculates the data without the consideration of sentence, word, or syllable structures which makes it difficult to (re-)associate the output with, e.g., syntactic constituents.

There are several approaches to the automatic annotation of prosodic events with relation to corpora (often with a focus on future speech synthesis) that go beyond the sole interpretation of acoustic cues and include basic morphosyntactic information as well, e.g., in form of part-of-speech (POS) tags. The *Prosodizer* (Braunschweiler 2003, 2006) can assign pitch accents and boundary tones during speech recognition in American English and German speech corpora following the ToBI labelling conventions (Silverman *et al.* 1992). The method relies on acoustic features as well as syntactic boundary labels and POS tags which are part of the corpus annotations. An evaluation showed more than 70% accuracy in pitch accent and boundary tone detection with major difficulties at the level of intermediate phrase boundaries. The multilingual prosody module of the Verbmobil system

integrates a word-based annotation and classification of boundaries, phrase accents, and sentence mood for German, English, and Japanese dialogues (Batliner *et al.* 2000, 2001; Wahlster 2013). Schweitzer and Möbius (2009) went beyond the word base and trained a number of classifiers on acoustic, phonological, and basic morphosyntactic attributes of German using the WEKA machine learning software (Witten and Frank 2005), reaching recognition accuracy rates of up to ~86% for the occurrence of accents, and ~93% for the occurrence of larger boundaries.

All of these approaches allow for the recognition and depiction of prosodic events in form of boundaries and accents, but none of them allow for real communication between prosodic structure and other modules of grammar. If (morpho)syntactic information in a given corpus is included in the system, it is used to facilitate prosodic annotation, but not vice versa, i.e., prosodic information is not used to determine (morpho)syntactic structure. None of the approaches are designed to allow for the prosodic disambiguation of syntactic structure or for signalling focus structures in order to enhance linguistic analyses by computational grammars.

Current large-scale grammar development projects which provide deep linguistic analyses include the Parallel Grammar project (ParGram, Butt *et al.* 2002; Sulger *et al.* 2013) based on Lexical-Functional Grammar (LFG; Kaplan and Bresnan 1982) and the DELPH-IN project in combination with the LinGO (Linguistic Grammars Online) Matrix effort (Bender *et al.* 2002; Copestake 2002) based on Head-driven Phrase Structure Grammar (HPSG; Pollard and Sag 1994). Other major grammar development efforts are based on CCG (Steedman 2000; Clark and Curran 2007) and TAG (Joshi 2003; Duchier *et al.* 2004; Gardent and Parmentier 2005).

So far, these grammar development approaches have focussed on the syntactic and semantic representation of language. There are no detailed implementations of p-structure (including prosody and (post)lexical phonology), although some initial attempts restricted to specific phonological phenomena have been made across frameworks (see, for example, Butt and King 1998, Bird 1992, Bird and Klein 1994, Klein 2000). Computational approaches to specific/isolated phonological phenomena without integration into a large-scale grammar have also been developed in frameworks based on constraint rankings

(as in Optimality Theory (Prince and Smolensky 2004); see, e.g., Tesar and Smolensky 1998; Becker *et al.* 2007; Yu 2018) and constraint weighting (as in Harmonic Grammar (Legendre *et al.* 1990); see, e.g., Potts *et al.* 2010). Penn and Carpenter (1999) combine two smaller-scale HPSG grammars of English and German with off-the-shelf speech recognition and TTS systems to allow for automatic translation and generation of spoken language. However, their system only includes spoken language in a detached manner in that a speech signal is first converted into a simple text string (which is then further processed by the grammar) and vice versa. To date, a real integration of spoken language into a large-scale computational grammar to enable deep automatic speech understanding has not been accomplished.

This paper uses the computational grammars developed in the spirit of Lexical-Functional Grammar (LFG), which have long been established as part of the ParGram project and have been used for a multitude of purposes with a strong focus on syntactic and semantic processing (a.o., Butt *et al.* 1999, 2002; Bobrow *et al.* 2007; Sulger *et al.* 2013; Crouch *et al.* 2017; Meßmer and Zymla 2018; Dalrymple *et al.* 2019). The input to all of these grammars is the s(yntactic)-string, which consists of a string of words that make up a written sentence (or a fragment thereof). In a standard computational LFG grammar, this string is tokenized into single words whose lexical morphosyntactic information is accessed and made available for further processing of the string in c(onstituent)- and f(unctional)-structures as well as semantic representations. This basic structure (including variations or extensions thereof) has been the established core structure of all computational LFG grammars. Grammars can be built via XLE, a state-of-the-art grammar development platform (a.o., Butt *et al.* 1999; Crouch *et al.* 2017), which allows researchers to build industrial-strength computational grammars for a wide range of languages and can be integrated with industrial-strength finite-state morphologies (Beesley and Karttunen 2003; Kaplan *et al.* 2004; Bögel *et al.* 2007).[1]

[1] See the XLE-Web interface which features a number of different computational LFG grammars that can be used interactively: https://clarino.uib.no/iness/xle-web.

While these grammars are well-established for syntactic and semantic analyses of texts, they are as of yet unable to process spoken language. As a consequence, linguistic phenomena whose analysis requires prosodic information (as demonstrated in examples (1) and (2)) cannot be interpreted by the traditional computational LFG grammars, although the combination of automatic speech recognition with linguistically deep computational grammars would be highly desirable and benefit both automatic speech understanding and speech synthesis.

This paper introduces a new system which bridges this gap between the automatic recognition of prosodic events and their linguistically deep analysis by computational LFG grammars, taking the prosodic disambiguation of syntactically ambiguous structures as a demonstration example. The implementation includes a representation of the speech signal in phonetic and phonological/prosodic terms, where the categorical representation of the latter enables the computational grammars to prosodically disambiguate syntactically ambiguous structures. This not only reduces overgeneration in the case at hand, but makes a linguistically fine-grained representation of prosodic categories (accents and boundaries) available for other modules of grammar, thus taking a huge step towards real automatic speech understanding.

The paper is structured as follows: Section 2 introduces the syntactically ambiguous data and briefly reports on a production experiment that establishes the relevant acoustic features for a prosodic disambiguation. Section 3 first gives a brief introduction to LFG and then describes the theoretical foundations behind the approach to the prosody-syntax interface proposed in this paper. Section 4 describes in detail all aspects of the computational implementation, from the interpretation of the speech signal to the disambiguation of syntactically ambiguous structures. This is followed by an evaluation of the system in Section 5. Section 6 concludes the paper.

2 THE DATA: SYNTACTICALLY AMBIGUOUS STRUCTURES

The following German example (3) has two possible interpretations:

(3) Sie sahen, dass
They saw that

[der Partner]$_{NP1}$ [der Freundin]$_{NP2}$ fehlte
the.MASC.NOM partner the.FEM.GEN/DAT friend was.missing

a) "They saw that the friend's partner was missing."

b) "They saw that the friend missed the partner."

There are two sources of ambiguity in this example: the syncretism of the determiner *der* 'the' and the verb's valency. The determiner is ambiguous in this position as it can be interpreted either as feminine dative or feminine genitive (Table 1), which makes the complete

case	masc	fem	neut
gen	des	*der*	des
dat	dem	*der*	dem

Table 1:
The German determiner system for the singular genitive and dative

second NP *der Freundin* 'the friend' be interpreted as either dative or genitive. Adding to this local ambiguity is the valency of the verb *fehlen* 'missing', which can be used in either intransitive or transitive constructions, the latter requiring a dative object. As a result, the second NP can either be interpreted as a dative object to the verb or as a possessor phrase to the first NP *der Partner*, as indicated by the two translations given in example (3). Such syntactically ambiguous structures result in overgeneration, i.e., the (computational) grammar returns several possible solutions as illustrated in Figure 1. Previous research has shown that syntactically ambiguous structures can often be disambiguated by means of prosody (Price *et al.* 1991) and several studies have demonstrated this for a number of German structures as well (Żygis *et al.* 2019; Gollrad *et al.* 2010).

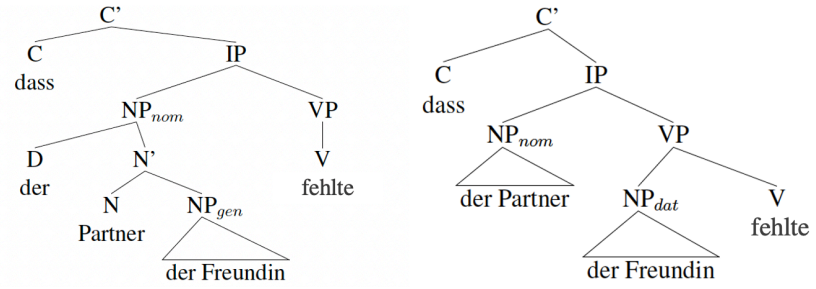


Figure 1: Two syntactic interpretations for example (3): genitive structure on the left, dative structure on the right

For structures as in example (3), current theories of the syntax-prosody interface would predict a prosodic phrase boundary to occur between the two NPs in the dative construction, but not in the genitive. Table 2 illustrates the predictions made by Selkirk's (2011) MATCH THEORY, which posits a phonological phrase (PhP/φ) for every syntactic XP (NP, PP, ...), in combination with Truckenbrodt's (1999) WRAP constraint, which assumes that a recursive XP/PhP is merged ('wrapped') into a single PhP.

For the syntactic structures given in Figure 1 and the string *der Partner der Freundin*, MATCH THEORY predicts a PhP boundary for every NP, resulting in two PhPs for the dative structure, and one nested PhP in the genitive structure. WRAP then assumes that the nested PhP in the genitive is wrapped into a single PhP. The algorithm thus assigns a PhP boundary after the first NP in the dative, but not in the genitive structure, as illustrated in Table 2.

Table 2: Prosodic phrasing predictions for the syntactic structures in Figure 1

Dative	Syntax		[der Partner]$_{NP}$ [der Freundin]$_{NP}$	
	Prosody	MATCH	$_{\varphi}$(der Partner)$_{\varphi}$(der Freundin)$_{\varphi}$	↕
		WRAP	$_{\varphi}$**(der Partner)$_{\varphi}$(der Freundin)$_{\varphi}$**	
Genitive	Syntax		[der Partner [der Freundin]$_{NP}$]$_{NP}$	
	Prosody	MATCH	$_{\varphi}$(der Partner $_{\varphi}$(der Freundin)$_{\varphi}$)$_{\varphi}$	↕
		WRAP	$_{\varphi}$**(der Partner der Freundin)$_{\varphi}$**	

In a production experiment, Bögel (2020) confirmed the theoretical predictions in Table 2. The stimuli consisted of nine fully ambiguous structures similar to example (3), where the first NP was always masculine and the second one feminine, followed by a verb with an ambiguous valency. All nouns had a disyllabic, trochaic foot structure

(i.e., the first syllable carried lexical stress and the second one was unstressed (x -)).

The participants were fifteen female native speakers of German.[2] Each participant was presented with a context and a target sentence. Participants were asked to read the context silently and to 'mentally understand' the sentence before producing it as naturally as possible. Each participant produced 18 sentences (9 genitive and 9 dative constructions), resulting in a total of 270 sentences.

A linear mixed effects regression model (lmer) with items and subjects as random factors yielded the following results:

- A significantly steeper **drop in the fundamental frequency (F_0)** ('Reset') between NP1 and NP2 (as measured at the final syllable of NP1 and the determiner of NP2) in the dative as compared to the genitive condition ($\beta = -9.31$, SE = 2.64, t = −3.53, $p < 0.01$).
- A **pause**[3] between the first and the second NP in the dative condition: ($\beta = -2.35$, SE = 0.92, t = −2.55, $p < 0.05$).
- The **duration** of the last syllable of the first NP was significantly longer in the dative condition than in the genitive condition ($\beta = -2.8$, SE = 0.79, t = −3.58, $p < 0.01$).

These findings confirm the placement of a prosodic phrase boundary after the first NP in the dative, and provide detailed information on the relevant acoustic indicators of a prosodic phrase boundary, namely duration, F_0 movement, and pauses.

While the experimental results are in line with the predictions in Table 2, the question remains how these findings can be used to prosodically disambiguate syntactically ambiguous structures in LFG.

---

[2] The main goal of the original production experiment was to find the prosodic cues that disambiguate the syntactic structures. In order to reduce variation with respect to pitch evaluation, the decision was made to only record female participants. For the computational implementation described below this has no effect, since the implementation normalizes pitch by means of semitones.

[3] Following the MAUS conventions, a pause is defined as a silence interval which lasts more than 100 ms. See https://clarin.phonetik.uni-muenchen.de/BASWebServices/help/help_faq#help_faq.

3 THEORETICAL ANALYSIS IN LFG

After a brief introduction to LFG, this section discusses the architectural assumptions made with respect to the interface between syntax and prosody from a theoretical perspective which in turn forms the basis for the computational implementation in Section 4.

The generative, non-transformational LFG framework (Kaplan and Bresnan 1982; Bresnan *et al.* 2016; Börjars *et al.* 2019; Dalrymple *et al.* 2019; Dalrymple 2023) has a modular architecture with parallel representative structures for separate linguistic aspects which constrain each other through mathematically well-defined functions. Different types of linguistic information are encoded in suitable representation structures. For example, the original core structures c(onstituent)-structure and f(unctional)-structure both represent different aspects of syntactic structure: While c-structure depicts linear order and syntactic constituency by means of tree diagrams as in Figure 1, f-structure captures key dependency relations like grammatical functions (e.g., subject and object) as well as other functional information such as tense/aspect or case. F-structures are represented in Attribute-Value-Matrices (AVMs) and are largely invariant across languages. These two structures are linked via the projection function ϕ to allow for communication between syntactic constituency and related functional information. A number of additional structures have been proposed over the years, including a(rgument)-structure, i(nformation)-structure, and m(orphological)-structure, each of which represents the linguistic information associated with that aspect of grammar. Correspondence between these structures is again ensured via well-defined projection functions (see Dalrymple 2023 for a general introduction to LFG).

Several proposals have also been made for p(rosodic)-structure (see Bögel 2023 for an overview). This paper follows the proposal made in Bögel 2015. It distinguishes between *comprehension* ('parsing' in computational terms), which describes the processing and subsequent understanding of the speech signal by a listener, and *production* ('generation' in computational terms), which describes the process from the initial concept to the actual form of an utterance. The present paper focuses on comprehension: It discusses the process of

going from a speech signal to a linguistic analysis, i.e., from phonetics to prosody to syntax.

In the proposal made by Bögel (2015), information at the prosody-syntax interface is exchanged on two levels: a) the *transfer of vocabulary* (ρ/π), which exchanges phonological and morphosyntactic information of lexical elements via a multidimensional lexicon, and b) the *transfer of structure* ($\natural$), which exchanges information on syntactic and prosodic phrasing, and on intonation. Figure 2 illustrates this interaction in LFG where syntactic constituent structure is represented by c-structure, prosodic/phonological information by p-structure, and the s(yntactic)-string is placed between them. Mathematically well-defined projection functions (here: $\natural$, ρ, π) allow for the correspondence between these modules.

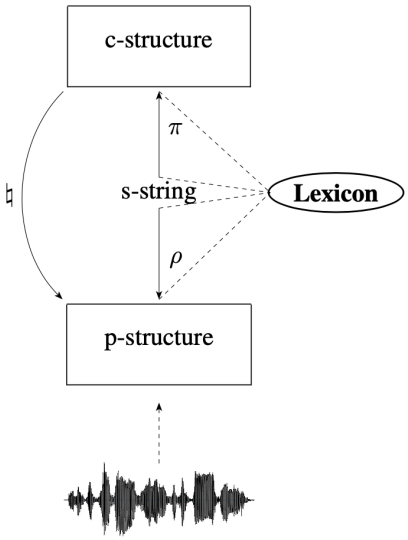


Figure 2:
The underlying architectural assumptions for the interface between syntax (c-structure) and prosody (p-structure)

P-structure 3.1

P-structure is represented via the p-diagram, a linear syllable-based representation of the speech signal over time (Figure 3). During comprehension, acoustic information from the speech signal feeds into p-structure and is stored at the *signal level*. Each syllable in the signal

...	...	...	...	...	...	...	signal
DURATION	0.15	0.25	0.25	0.13	0.31	0.19	↓
FUND. FREQ.	192	181	269	209	188	218	
SEGMENTS	[de:6]	[pa6t]	[n6]	[de:6]	[fROYn]	[dIn]	
VECTORINDEX	S_1	S_2	S_3	S_4	S_5	S_6	

Figure 3: The signal level of the p-diagram for *der Partner der Freundin*

receives a vector (S_n) which contains information, e.g., on the segments,[4] the duration, or the mean fundamental frequency (F_0) of that syllable.[5] Figure 3 shows the p-diagram fragment for the six syllables of *der Partner der Freundin*. The 'raw' signal information given in Figure 3 encodes patterns which can be interpreted in categorical terms at the *interpretation level*. For example, a strong rise in F_0, a following drop (from S_2 to S_4) and a comparatively long duration on the last (unstressed) syllable of *Partner* (as seen at S_3: [n6]) are strong indicators for a phonological phrase boundary. As a result, PHRASING = $)_\varphi$ is added to the syllable's vector at the interpretation level (Figure 4). Further possibilities at the interpretation level include, for instance,

...	...	...	...	...	...		interpretation
PHRASING	-	-	$)_\varphi$	$(_\varphi$	-	-	↓
SEMIT_DIFF	...	-1	6.8	-4.3	-1.9	2.6	
GToBI	-	L*	+H H-	-	L*	+H	
DURATION	0.15	0.25	0.25	0.13	0.31	0.19	signal
FUND. FREQ.	192	181	269	209	188	218	↓
SEGMENTS	[de:6]	[pa6t]	[n6]	[de:6]	[fROYn]	[dIn]	
VECTORINDEX	S_1	S_2	S_3	S_4	S_5	S_6	

Figure 4: The interpretation level of the p-diagram for *der Partner der Freundin*

[4] Segments are represented in SAMPA, a computer-readable phonetic alphabet (Wells 1997).

[5] Mean F_0 is calculated based on the complete syllable and serves as a quick orientation for the researcher, not as a basis for the computational calculation discussed below.

a GToBI (Grice and Baumann 2002) analysis of the pitch in terms of high and low tones, or the differences between adjacent pitch values measured in normalized semitones (SEMIT_DIFF), which allow for an interpretation of the slopes leading to and from the accent (i.e., the scaling of the tones). While the p-diagram representation was developed with LFG in mind, it is an encapsulated, adaptable, and extendable representation that can be plugged into any modular framework.

The transfer of vocabulary 3.2

The transfer of vocabulary associates morphosyntactic and phonological information in lexical elements via the multidimensional lexicon. Following proposals made by, e.g., Levelt *et al.* (1999), the lexicon includes several dimensions (Table 3): The *s(yntactic)-form* contains the traditional morphosyntactic information associated with a particular lexical item (e.g., number, gender, or case), while the *p(honological)-form* contains information on the segments and the metrical frame of that entry: the number of syllables, the lexical stress pattern, and the prosodic status (e.g., whether the element is a clitic, underspecified, or a prosodic word). The lexicon in Table 3 shows the entries for the noun *Freundin*, which is feminine, singular, and a prosodic word with two syllables in a trochaic foot structure. The determiner *der* has ambiguous case information (genitive or dative) and consists of a single, prosodically underspecified syllable.[6] The lexicon is modular in that

s-form			p-form	
N	(↑ PRED)	= 'Freundin'	SEGMENTS	/f R OY n d I n/
	(↑ NUM)	= sg	METRICAL FRM	$('\sigma\sigma)_\omega$
	(↑ GEND)	= fem		
D	(↑ PRED)	= 'der'	SEGMENTS	/d e 6/
	(↑ NUM)	= sg	METRICAL FRM	σ
	(↑ GEND)	= fem		
	(↑ CASE)	= {gen | dat}		

Table 3: (Simplified) lexical entries for *der* and *Freundin*

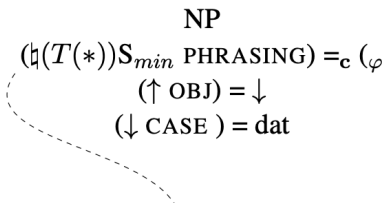
[6] The determiner 'der' can also be used in the nominative masculine. This option is omitted from Table 3 since it is not relevant for the data discussed in this paper.

there is a strict separation of module-related information: Each lexical dimension can only be accessed by the related module, i.e., p-structure can only access p-forms, and c-structure can only access s-forms. At the same time, the lexicon has a translating function: Once a dimension is triggered, the related dimensions can be accessed as well. During comprehension, if p-structure accesses a particular p-form, the related s-form becomes available and the morphosyntactic information is instantiated to syntactic structure. Conversely, during production, if c-structure accesses an s-form, the related p-form information becomes available to p-structure, ultimately forming the foundation for the phonetic utterance.

3.3 *The transfer of structure*

The transfer of structure exchanges information on prosodic and syntactic constituency via the projection function ♮. Figure 5 shows the annotation for an object nominal phrase (NP) which checks whether there is a (left) phonological phrase boundary associated with the left edge of the NP's corresponding prosodic unit in p-structure. The annotation can be read as follows: For all terminal nodes T (= {D/*der*, N/*Freundin*}) of the current node * (=NP), for the syllable with the smallest index (S_{min}) in this set of terminal nodes (i.e., the leftmost syllable), there must be ($=_c$) a (left) phonological phrase boundary ($(_\varphi$. If this is the case, an object with dative case is projected to f-structure: (↑ OBJ) = ↓ and (↓ CASE) = dat state that any material occurring under the current syntactic node (here: NP) is stored as part of the

Figure 5: The *transfer of structure*: prosodic and syntactic phrasing



PHRASING	...	...	((φ	...	...	...	...		
SEGMENTS	...	...	[de:6]	[fROYn]	[dIn]	...	...		
VECTORINDEX	...	...	**S4**	**S5**	**S6**	...	...		

grammatical function 'object' in f-structure, and that the related case is dative.[7] The annotation of the c-structure node NP thus combines two projection functions: First, the information concerning prosodic phrasing at p-structure is determined. If a prosodic phrase boundary is present, the current node is then interpreted as the object of the clause, effectively disambiguating the syntactically ambiguous structure in example (3)/Figure 1.

Figure 6 shows the complete analysis of a dative structure at the prosody-syntax interface during comprehension, where the transfer of

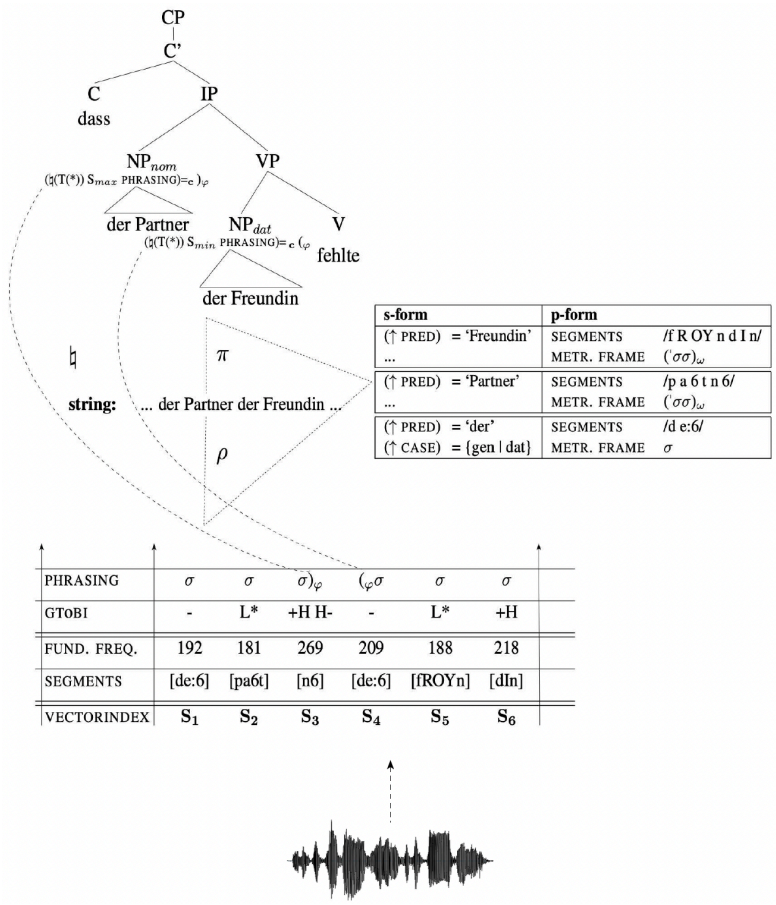


Figure 6: A dative structure at the prosody-syntax interface: comprehension

[7] For further explanations of the correspondence between c- and f-structure, the interested reader is referred to Dalrymple 2023.

vocabulary matches segmental strings against lexical items and the transfer of structure disambiguates the syntactically ambiguous structures based on larger prosodic constituents, in this case a phonological phrase boundary between the two NPs [*der Partner*] and [*der Freundin*].

This section provided the theoretical background for the prosodic disambiguation of syntactically ambiguous structures in LFG. The following section takes this theoretical analysis as a starting point and serves as a blueprint for an integration of prosodic structure into the existing computational LFG-grammars, thus enabling the grammars to include and process information from the speech signal as well.

4 COMPUTATIONAL IMPLEMENTATION

The computational implementation of the theoretical analysis presented in Section 3 is a new approach that includes the integration of spoken language. It categorizes the gradient information gained from the signal and organizes it within the p-diagram at p-structure. It then matches the information against a lexicon containing p-form and s-form information. The matching process leads to the creation of the s-string which is the linear concatenation of all matched s-forms and thus corresponds to the string that was originally used as input to the computational LFG grammars. The s-string (and the lexical morphosyntactic information associated with each word in the string) enables c- and f-structure to be parsed with XLE (Crouch *et al.* 2017), the grammar development platform used to create large-scale LFG grammars. In a final step, the implementation allows for the disambiguation of syntactic structures based on the automatically determined prosodic phrase boundaries at p-structure. The implementation is in Perl, with added scripts written in Praat (Boersma and Weenink 2021), xfst (Beesley and Karttunen 2003) and R (R Core Team 2016), all of which are open-source and commonly used software.[8]

[8] The source code for the computational implementation is available under https://github.com/ticle2/prosody-syntax-interface-in-LFG.

Extracting and normalizing information from the speech signal 4.1

Figure 7 shows the input used for the computational implementation, a sound file annotated with SAMPA syllables. For the annotation, the data was first automatically annotated using the Munich Automatic Annotation System MAUS (Kisler *et al.* 2017; Schiel 1999), which aligns the speech signal with SAMPA segments (but not syllables) based on a given orthographic input. In order to obtain the syllabic annotation that serves as a base for the system described in this paper, the segmental annotation was matched against a lexicon created from the CELEX database for German words (Baayen *et al.* 1995). This database allows for the creation of different custom-tailored lexicons, in this case a lexicon containing the SAMPA-syllables for all the German words in the database. In a next step, the segmental MAUS annotation was matched against the syllable-based lexicon, keeping track of the start and end times of each syllable in the speech signal. Based on this information, a new Praat annotation tier was created containing only the SAMPA syllables. The syllable tier was then manually checked for alignment mistakes that regularly occur with forced

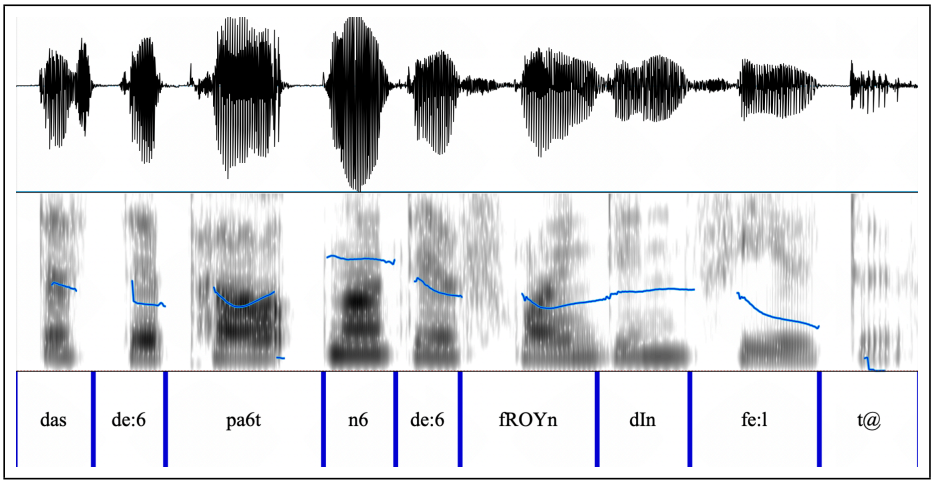


Figure 7: Input: a sound file annotated with SAMPA syllables in Praat, here for example (3)

aligners like MAUS (see, e.g., Gonzalez *et al.* 2020).[9] In a first step, a Praat script collects information from the speech signal. The script extracts the syllable segments, the duration of each syllable, and the mean F_0-values for each syllable for the signal level of the p-diagram (Figure 3). For a fine-grained analysis of the pitch during processing, the script furthermore divides each syllable into five even-spaced subintervals, takes the mean F_0-values of each subinterval and turns these values into semitones, thus effectively normalizing duration and pitch. In order to minimize the effect caused by incorrect pitch calculations by the Praat algorithm, the system checks for outliers among the semitones and — if present — excludes them from the following estimation of high and low tones.[10]

Each subinterval is tagged for position within the syllable, either as central, or as preceding or following a syllable boundary. This measure was implemented to allow for the determination of early or late pitch accents. For example, if a pitch accent unexpectedly occurs in an unstressed syllable preceding the stressed target syllable, the information that it occurs directly at the boundary to the target syllable would relate this accent to the target syllable as an 'early' accent.

4.2 *Interpreting the pitch*

In a second step, the raw values from the speech signal are interpreted in terms of categories that are 'meaningful' for other modules of grammar. Different measures are used for the interpretation of the pitch: In addition to the semitones and the differences between these semitones indicating falls and rises, the implementation also utilizes the residuals of a linear regression based on the pitch values of a given speech

[9] It would, of course, be desirable to have a system that provides a deep linguistic analysis from the raw speech signal to a syntactic structure. However, the fact that forced alignment of orthographic text to segmental annotation requires manual correction by a human annotator means that uncontrolled alignment (i.e., without the orthographic representation) would most likely result in increased inaccuracy. Since the main focus of this paper is on the implementation of the prosody-syntax interface, and not automatic speech recognition, the system starts with input files that are annotated with SAMPA syllables.

[10] Where an outlier is any data point above the 3rd quartile +1.5 Interquartile range (IQR) and below the 1st quartile −1.5 IQR (e.g., Winter 2019, 60).

signal. This measure was introduced to account for the lowering or rising of the pitch over time depending on the sentence type; e.g., in declaratives the pitch tends to get lower towards the end of the sentence (a.o., Ladd 1984; Xu 2005). This general tendency is reflected by the regression line. Residuals return the distance of each value from this line and are thus a good measure to describe deviations from the average, i.e., surprising values.

Both semitones (and their differences) and residual values are then used a) to determine the minimums (L) and maximums (H) in a given signal, and b) to determine the slopes between these categories, i.e., whether the rises/falls are strong or weak.

In order to mark both categories (i.e., type of accent and type of slope) in one representation, we devised the system in Table 4: Each level of L or H is characterized by a particular height and shape of the slope leading to it (*lead*) and following it (*tail*). Taken together,

Cat.	Max/Min	Lead	Tail
H4/L4	Max/Min	steep	steep
H3/L3	Max/Min	steep	flat
H2/L2	Max/Min	flat	steep
H1/L1	Max/Min	normal	flat

Table 4:
(Part of the) system of pitch accents and slopes in the computational implementation

semitones and residuals allow for the detection of deviations from the norm in the signal, i.e., maximums (H) and minimums (L). In order to exclude microprosodic effects (which might cause two tones to appear on one syllable), the distance between any Hs and Ls has to span at least one syllable. Slopes to and from a H/L tone are calculated based on the ratio between the semitones of adjacent Ls and Hs and the distance (the number of subintervals) between them. The resulting values indicate whether the associated slopes are steep or flat.H4 and L4 thus represent accents where the lead and the tail show a strong rise and fall respectively, while H1 and L1 have a relatively flat lead and tail. L2/L3 and H2/H3 are positioned between these two extremes, with each having a slightly different shape depending on the slopes. The following Figure 8 demonstrates some of the H and L tones discussed in Table 4 for a dative example as they would be assigned by the system. The tone values are stored as part of the *interpretation* level

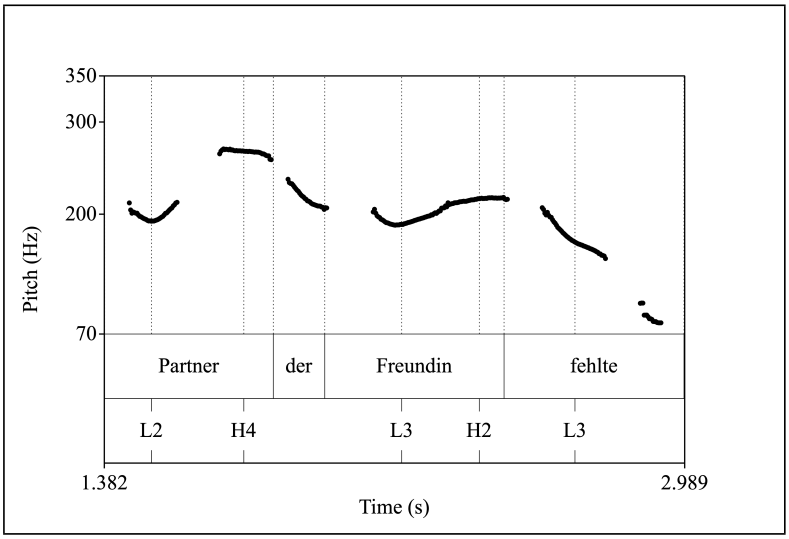


Figure 8: Tone values for a dative example as calculated by the system

of the p-diagram (Figure 4), where they replace the traditional GToBI values in order to facilitate (and simplify) the automatic interpretation by other modules of the grammar.

4.3 *Interpreting duration*

The categorization of a specific syllable as 'long' or 'short' is not a trivial process. Since the input to the system is always a single file, there is no direct way to compare the duration of one syllable to duration measures of syllables in similar positions in other input signals.[11] For the current analysis, this problem was resolved by creating a precompiled threshold for duration categorization. The compilation was based on the 270 utterances produced in the experiment described in

[11] There are two ways to deal with this problem: a) a database of all possible syllables in all possible (word) positions over many speakers in order to get an estimate of the expected syllable duration, or b) an estimation of syllable duration tailored to the dataset at hand. While a database would allow a more universal assessment of syllable duration, creating such a resource would be very time-consuming and the considerable size of such a database would be more of a hindrance to the system at hand. Since this paper is a proof of concept, we leave this work to further research, and show how option b), a tailored solution, can be realized.

Section 2, more precisely, on the stretch of data from the start of the subordinate clause to the end of the second noun; 7 syllables in total. Strictly speaking, the verb should have also been analyzed as part of this clausal stretch. However, it was disregarded for this particular calculation because different verbs show too much segmental variation. This, in turn, would have had an (undesired) effect on the duration measures.

Two values were used to classify syllables as long or short: speaker tempo and syllable duration. For the estimation of speaker tempo, the duration of each of the seven target syllables was added up for each single recording and then divided by 7. The resulting values for each signal produced by a single speaker were added up again and the mean over all values was calculated. This mean value was taken to represent the individual speech tempo for each speaker. The following Table 5 shows the distribution of speaker tempo values over all speakers. As we can see, the 'fastest' speaker has a rate of 0.150 seconds per syllable and the 'slowest' speaker has a rate of 0.225. The overall mean was 0.184. For the categorization into slow and fast speakers, the first and third quartile (0.170 and 0.196 respectively) were used as thresholds. Values below/above these thresholds can be deemed unexpected from a statistical perspective, so any speaker with a value below 0.170 could safely be considered as 'fast', and any speaker above 0.196 as 'slow'.

Minimum	1st Quartile	Mean	3rd Quartile	Maximum
0.150	0.170	0.184	0.196	0.225

Table 5: Distribution of speaker tempo values over all speakers in seconds per syllable

In addition to speaker tempo, we also determined the duration of each individual syllable in the target area in comparison to all syllables in the same position in the overall dataset, e.g., each first syllable in the first noun was compared to all other syllables that also occurred in the first position of the first noun. For these values, the mean duration of each syllable over all the speakers was taken; outliers were excluded.

For the fourth syllable in the target area (which corresponds to the second syllable of the first noun, e.g. [nɐ] in *partner*), we observed the distribution in Table 6. As discussed in Section 2, the fourth syllable

Table 6: Distribution of duration values for the fourth syllable in the target area over all speakers in seconds per syllable

Minimum	1st Quartile	Mean	3rd Quartile	Maximum
0.1579	0.1681	0.1783	0.1885	0.1987

is significantly longer in the dative condition than in the genitive, thus signalling a prosodic phrase boundary. Syllables with a duration above the 3rd quartile were interpreted as 'long' (= increased likelihood of boundary), and syllables below the 1st quartile as 'short' (= no boundary following).

While this estimation of expected and unexpected values of syllable duration is a good indication of a following prosodic phrase boundary, any duration value needs to be viewed with reference to speaker tempo. The reason is that a slow speaker will per se also produce slow syllables which will confound the calculation of a prosodic phrase boundary. To control for this particular factor, syllables were only categorized as slow if the speaker had a fast or normal speaking rate. For slow speakers producing slow syllables, the difference between the speaker's tempo and the overall mean speaker tempo was taken and subtracted from the duration value of the syllable in question. If this syllable could still be classified as slow, the value was retained.

Both speaker tempo rates and individual syllable duration are stored as part of the system and are accessed during signal interpretation in order to facilitate boundary calculation.

4.4 *Lexical matching: the transfer of vocabulary*

During the transfer of vocabulary, the input from the speech signal is matched against the p-forms of the multidimensional lexicon, which then makes the associated s-forms available for syntactic parsing. In order to acquire the correct s-string, the p(honological)-string, which is created by concatenating the SAMPA syllables from the input speech signal (... *de:6.pa6t.n6.de:6.fROYN.dIn* ...), is matched exhaustively against a lexicon including phonological and morphosyntactic material as described in Section 3, Table 3. The lexicon is a finite-state

Input (p-string)	Lexicon		Output (s-string).
	p-form	**s-form**	
... de6.fROYn.dIn ... →	de:6	der	→ ... der Freundin ...
	fROYn.dIn	Freundin	
	...	...	

Table 7: The transfer of vocabulary: from p-string to s-string

transducer (xfst; Beesley and Karttunen 2003), where the upper side corresponds to the s-form, and the lower side to the p-form information associated with the lexical item. Matching the p-string against the lexicon results in the corresponding s-string (... *dass der Partner der Freundin* ...), which constitutes the input for the syntactic structure. Apart from making the s-string and the associated morphosyntactic information available to c- and f-structure, the matching of the p-string against the lexicon also makes the lexical p-form information (e.g., information on lexical stress or prosodic word/clitic status) available to the p-diagram.[12]

Prosodic phrase boundaries and the p-diagram 4.5

The previous sections described the different aspects relevant for the representation of a speech signal at p-structure. As a last step, prosodic phrase boundaries are calculated.

The production experiment reported in Section 2 elicited the acoustic factors which can be relevant for the determination of a PhP boundary: a rise in F_0 towards the boundary followed by a drop after the boundary, a pause, and a relatively long pre-boundary syllable.

[12] This information is especially relevant for production (not discussed here), because it allows the modelling of a prosodic baseline that can later be 'translated' into phonetic terms. But it is relevant for comprehension as well, in that it is generally assumed that pitch accents are only associated with lexically stressed syllables in German. Due to vowel quality differences and other reasons, however, the algorithm might also determine the local maximum or minimum to be on the previous or following syllable. Lexical stress (possibly in combination with positional information of the accent in the syllable, cf. Section 4.1) could in principle be used to shift the accents to the target syllable in the p-diagram representation.

Based on the pitch calculations in Section 4.2, the duration values in Section 4.3, and on the presence or absence of pauses, the implementation estimates the likelihood of a prosodic phrase boundary in the position at hand. If any of the following constraints are minimally met, a PhP boundary is included.

1. a H4 accent
2. a H3 accent in combination with a surprising residual value; only very high values (above 3 or below −3) are taken into account
3. a H3 accent with a long syllable
4. a pause

Figure 9 shows an automatically created p-diagram for the string *der Partner der Freundin* based on a speech signal with the dative construction. As discussed in Section 3.1, each vector includes the segments, the duration, and the mean F_0-value for the associated syllable. The p-diagram also contains the lexical p-form information by marking lexically stressed syllables with x and by adding the lexical prosodic unit information to the attribute PROS_PHRASE (prosodic phrasing). While each function word (*dass, der*) is represented by an underspecified syllable σ, the nouns' prosodic word status is indicated by a set of unmarked brackets. The automatically calculated PhP boundaries are marked by $_{pp}($ and $)_{pp}$. The p-diagram in Figure 9 shows that the system can give a fairly accurate categorical representation of the speech signal. The PhP boundary occurs after the first NP, thus indicating a dative structure. There are also several open questions, e.g., whether the low tone L2 associated with vector 2 (GToBI: L*), which occurs just before the syllable boundary, should be 'moved' to vector 3 where the syllable carries lexical stress, or whether an additional attribute for

pros_phrase	pp(σ	σ	(σ	σ))pp	pp(σ	(σ	σ)	(σ	σ))pp
pitch_tones		L2		H4		L3	H2	L3	
lex_stress	–	–	x	–	–	x	–	x	–
F0_mean	225.62	196.49	198.90	267.53	219.35	194.02	213.77	176.27	85.71
duration	0.17	0.16	0.33	0.18	0.14	0.30	0.20	0.28	0.22
syllables	das	de:6	part	n6	de:6	frOyn	dIn	fel	t@
Vector_index	1	2	3	4	5	6	7	8	9

Figure 9: P-diagram for a dative interpretation of the string *der Partner der Freundin* ('the partner of the friend')

'early' or 'late' L/H tones would be more useful. We leave questions like these to further research.

The information on prosodic phrase boundaries at p-structure is now available for further processing. However, in order to disambiguate the syntactic structure, c-structure has to recognize the ambiguity in the first place and be able to check for possible cues for a particular interpretation at p-structure.

Disambiguation and the fchart: the transfer of structure 4.6

This section describes how the overgeneration caused by syntactic ambiguities as in example (3) can be automatically disambiguated by intersecting a computational LFG-grammar for German with the p-structure created above.[13] In a first step, the syntactic string determined in Section 4.4 is parsed with a computational LFG grammar. In order to achieve this, the main Perl script creates an XLE-internal *xlerc* script (Crouch *et al.* 2017) which starts the computational grammar and parses the s-string. As expected, the grammar overgenerates and returns the two syntactic strings in Figure 1. The syntactic ambiguity leading to these parses can be made accessible by instructing the xlerc script to print out the so-called *fchart*, a Prolog representation of all choices, constraints, c-structure relations and more, in one file. The command in (4) will return a Prolog file *filename.pl*, which can be processed further by the main Perl script.

(4) `print-prolog-chart-graph filename.pl`

The following descriptions discuss only the relevant parts of the extensive fchart Prolog representation and the way they can be used to determine the actual linear position of the ambiguity (with the ultimate goal to check for prosodic phrase boundaries at that position).

The fact that there are two possible syntactic structures is encoded in the fchart section 'Choices' with the variables A1 and A2 (in this example, A1 corresponds to the dative option, and A2 to the genitive). This information alerts the script to the ambiguity of the parsed syntactic string.

[13] The following discussion describes this process in some technical detail; readers who are not familiar with XLE might want to continue with Section 5.

```
(5)   [
    choice([A1,A2], 1)
      ],
```

The next fchart section 'Constraints' indicates that the two choices A1 and A2 are based on the ambiguity in the verb's valency.

```
(6) % Constraints:
    [
  cf(A1,eq(var(3),semform('fehlen',4,[var(4),var(2)],[]))),
  cf(A2,eq(var(3),semform('fehlen',4,[var(4)],[]))),
    ],
```

As shown in (6), the verb *fehlen* 'miss' in choice A1 has two arguments (represented by abstract variables, var(4) and var(2)) and in choice A2 only one argument (var(4)). With respect to the linguistic example discussed in this paper, choice A1 thus refers to the (transitive) dative, i.e., to the two arguments [*der Partner*] and [*der Freundin*], and choice A2 to the (intransitive) genitive with one nested argument [*der Partner* [*der Freundin*]]. This difference in argument structure and the variable names of the arguments for each choice are then further tracked by the main script in order to ultimately relate these abstract variables to concrete surface forms.

In the fchart section 'C-structure' in (7), the *fspans* of the arguments (i.e., the s-forms over which the argument 'spans') are encoded with indexing numbers, where the first number indicates the start of the span, and the second number the end of the span. In example (7), the two arguments in choice A1 have the fspan from 17 to 28 for the first argument var(4) and the fspan from 29 to 41 for the second argument var(2). For choice A2, the single argument var(4) has an fspan from 17 to 41 (notably including the range of both arguments in choice A1).

```
(7) % C-Structure:
      [
    ...
    cf(A1,fspan(var(4),17,28)),
    cf(A1,fspan(var(2),29,41)),
    ...
    cf(A2,fspan(var(4),17,41)),
    ...
```

These numbers correspond to the surface forms (i.e., the s-forms or terminal nodes at c-structure). They indicate the start and the end of each of the arguments. In the next step, the script relates these index numbers from the fspans to the surface forms. Index number 17, the starting position of the first argument var(4) in both option A1 and A2 (cf. (7)), is associated with the start of the (first) determiner *der* of the first NP [*der Partner*] shown in the fchart excerpt in (8).

(8) `cf(1,surfaceform(9,`**`'der',17,20`**`))`
→ start of the first argument var(4) in both options

In choice A1, the span of the first argument var(4) is terminated with the indexing number 28, which also indicates the end of the surface form *Partner* in example (9). The first argument var(4) in choice A1 (but not A2) is thus the NP [*der Partner*].

(9) `cf(1,surfaceform(11,`**`'Partner',21,28`**`))`
→ end of the first argument var(4) in option A1 (subject in the dative construction)

As shown in (10), the surface form of the determiner of the second NP starts with index number 29. As seen in (7), this is also the start of the second argument var(2) in choice A1.

(10) `cf(1,surfaceform(13,`**`'der',29,32`**`))`
→ start of second argument var(2) in option A1

Finally, the surface form *Freundin* ends with index number 41, the terminating index number of the second argument (var(2)) of choice A1, and of the first and only argument (var(4)) of choice A2.

(11) `cf(1,surfaceform(15,`**`'Freundin',33,41`**`))`
→ end of second argument var(2) in option A1 (object of the dative)
→ end of first argument var(4) in option A2 (subject of the genitive)

For choice A1, the second argument thus stretches from the beginning of the second determiner (index 29, (10)) to the end of *Freundin* (index 41, (11)): [*der Freundin*]. In contrast, the argument for choice

A2 stretches all the way from the beginning of the first determiner (index 17, (8)) to the end of *Freundin*: [*der Partner der Freundin*].

By going through the fchart step by step, following each of its two choices A1 and A2, the script can pinpoint the position of the ambiguity in the syntactic string. In this case, this is the position at the end of the first NP [*der Partner*], where choice A1 concludes the first argument, and choice A2 does not.

Since the edges of syntactic NPs are associated with PhP boundaries (as established with the production experiment in Section 2), the algorithm now needs to check whether there is a PhP boundary after the last syllable of *Partner* in the p-diagram created in the last section. If this is the case, then choice A1 (the dative) should be selected, because we would expect a PhP boundary to be present between the two arguments. If there is no PhP boundary then choice A2 (the genitive) is more likely because the single argument should not be 'interrupted' by a PhP boundary. The selected option can be encoded in the Prolog file by automatically rewriting the fchart section 'Choices' which originally contained both choices (see (5)). The following example shows the 'Choices' section rewritten for choice A1 (i.e., the dative).

(12)
```
[
select(A1, 1)
   ]
```

In the last step, the main script starts an xlerc script containing the command in (13) which reparses the altered fchart. Since only one choice (A1) is given, the script only pays heed to the structures and constraints associated with that choice and ignores the others, thus effectively disambiguating syntactic structure by means of prosodic information.

(13) `read-prolog-chart-graph filename_new.pl`

Figure 10 shows XLE's c-structure output after the script reparsed the disambiguated fchart.

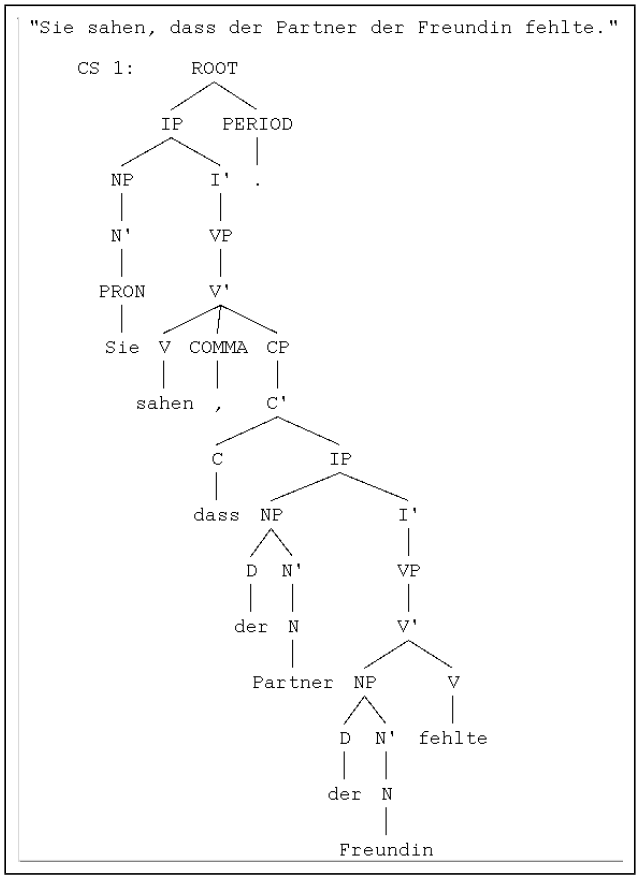


Figure 10:
A prosodically disambiguated dative c-structure

EVALUATION 5

For the evaluation, the recordings described in Section 2 were used to create a gold standard. Since spoken data has a lot of variation (with statistical analyses mostly only capturing tendencies), the data first needed to be sorted into representative and non-representative recordings for each case condition. To this end, a perception study was conducted in order to determine which of the recordings were most likely to be interpreted as datives or genitives by listeners. The motivation for this approach is to only evaluate the system on recordings that human listeners would be able to identify as well.

5.1 *Perception study*

In an online perception experiment, 32 native German speakers were asked to rate the 270 recordings from the production study described in Section 2. For the experiment, the recordings were randomized and assigned to different experimental lists. Each participant was asked to listen to nine genitive and nine dative recordings, and to indicate which meaning they thought was associated with the signal on a scale from one to five. On the scale, 1 (and to a lesser degree 2) represented a dative interpretation, 3 was considered 'undecidable', i.e., listeners did not show a clear tendency towards the case condition, and 5 (and to a lesser degree 4) represented the genitive interpretation. Each sentence was rated by two or three listeners (depending on the list), resulting in a total of 576 ratings. Only the sentences that were correctly rated at least twice (i.e., where the case of the produced sentence matched the case perceived by the listeners) were included in the gold standard and used for the evaluation.

Although datives and genitives were evenly distributed in the presented material, listeners were much more likely to mark a recording as dative. Figure 11 shows the distribution of listener responses over all recordings. A non-parametric two-tailed Wilcoxon rank sum test showed that the response values differed significantly from the actual case values ($W = 35765$, $p < 0.01$). Table 8 shows the 576 rating responses of the perception experiment where 32 listeners each rated 18 (9 dative and 9 genitive) randomized recordings. The results confirm that the mismatches between listener responses and actual case values were particularly high for the genitive recordings. This

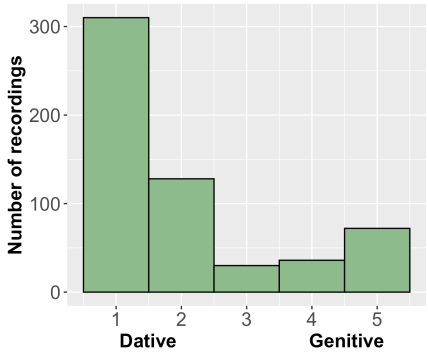


Figure 11:
Listener ratings of (equally distributed) dative and genitive recordings

	Dative	Genitive
Match	238	74
Mismatch	50	214

Table 8:
Matching and mismatching occurrences between listener ratings and actual case condition

mismatch is likely to be due to the general historic decline of the genitive in comparison to the dative (see, e.g., Scott 2011; Pittner 2014). As a consequence, the recordings that made up the gold standard were imbalanced between the two case conditions: From the original 270 recordings, 78 were categorized by at least two listeners as dative (1 or 2) and 17 as genitive (4 or 5). Note also that the recordings of one of the 15 speakers that took part in the production experiment described in Section 2 never received correct ratings by the participants of the perception study, i.e., this speaker did not use the prosodic cues that were necessary for the listeners to disambiguate the syntactic structure. For this reason, the following evaluation is only based on the recordings from 14 speakers.

Evaluation 5.2

In a next step, the gold standard recordings were semi-automatically annotated with SAMPA syllables following the process described in Section 4.1. Input to the system was a single wav-file with a corresponding TextGrid containing one Tier with SAMPA syllables (as illustrated in Figure 7). Each output by the system was checked for syntactic disambiguation and the placement of a correct prosodic phrase boundary in the target position in the p-diagram.

We present two types of evaluations. The ‘broad’ evaluation includes ratings that only show a tendency towards a particular interpretation: If two listeners rated a recording with a 2 or if there were mixed ratings (1 and 2), the recording was still classified as a dative even though the choice of rating showed some insecurity. By contrast, the ‘narrow’ evaluation only included recordings where all listeners were confident of the interpretation, i.e., all of them uniformly rated a dative with a 1 and a genitive with a 5. The reason for this distinction will become clear below.

5.2.1 Broad evaluation: results

The broad evaluation included 78 dative and 17 genitive recordings. The system was able to correctly interpret 68 of the 95 cases (71.5%). Figure 12 shows the results for each case condition and for both conditions taken together. Table 9 shows the system's performance measures for the broad evaluation. Since the data used for the broad evaluation still contained a level of insecurity (ratings 2 for a dative and 4 for a genitive), the evaluation was repeated including only the recordings where at least two speakers unanimously rated a dative as 1 or a genitive as 5.

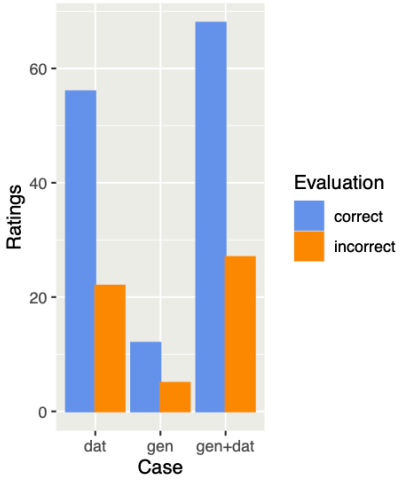


Figure 12: Correctly and incorrectly labelled input signals sorted by case condition; broad evaluation

Table 9: Precision, Recall, and F_1-score measures for the broad evaluation

	Precision	Recall	F_1-score
Dative	0.918	0.718	0.806
Genitive	0.353	0.706	0.471
Macro-average	0.636	0.714	0.639

5.2.2 Narrow evaluation: results

For the narrow evaluation, only recordings rated confidently as dative or genitive (i.e., 1 or 5) by at least two listeners were used. This resulted in 48 recordings for evaluation (38 dative, 10 genitive). The system was able to correctly determine 79% of the input (see Figure 13),

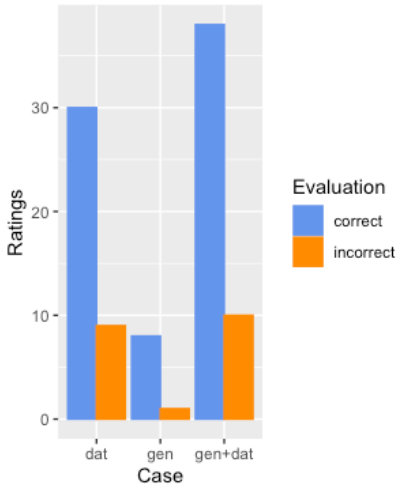


Figure 13:
Correctly and incorrectly labelled input signals sorted by case condition; narrow evaluation

	Precision	Recall	F_1-score
Dative	0.968	0.769	0.857
Genitive	0.471	0.889	0.616
Macro-average	0.72	0.829	0.737

Table 10:
Precision, Recall, and F_1-score measures for the narrow evaluation

i.e., the results are noticeably higher compared to the broad evaluation where tendencies (2 and 4) were included as well. The better performance of the system in the narrow evaluation is also reflected in the performance measures in Table 10. Although these values are promising, there are still a number of recordings which were correctly identified by the listeners, but not by the system. The following section discusses some additional findings and possible reasons for this difference.

Discussion 5.3

As discussed above in Section 5.1, the evaluation data is based on 14 out of 15 speakers who took part in the original production study, as none of the recordings by speaker no. 15 were correctly rated by the participants of the perception study. Furthermore, the data is very imbalanced, which reflects a more general preference by speakers to use the dative instead of the genitive in object constructions. However,

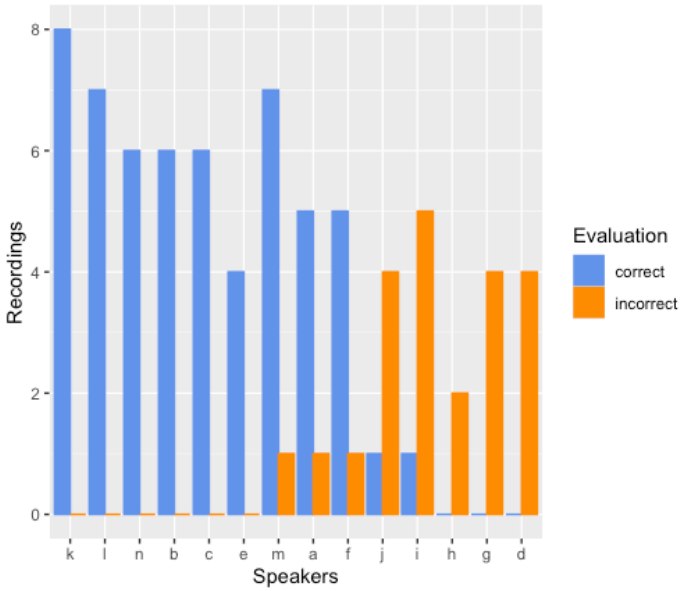


Figure 14: Results of the broad evaluation of the dative recordings sorted by speaker and correctness of the evaluation

as the system is not trained on corpus data, it is not affected by this preference for the dative construction.

A closer look at the data reveals strong speaker variation as illustrated in Figure 14, where the correct and incorrect evaluations for the dative are displayed for each speaker separately. According to Figure 14, the speakers can be sorted into two categories: Speakers whose recordings were rated correctly by both the system and the listeners (blue), and those whose recordings were rated correctly by the listeners, but not the system (orange). While there are five speakers with both correct and incorrect system ratings, the speakers can still be clearly associated with one of the two groups. In experimental terms this means that there is a group of speakers who (predominantly) signal the dative by acoustic means which were not captured by the production experiment described in Section 2; i.e., these speakers do not use pitch movement, a pause, or duration as acoustic cues at the target position between the two NPs. As a consequence, the system cannot correctly distinguish between the two syntactic structures.

Since the experiment was designed specifically for this target position, it is at this point not possible to determine the strategy used by this subgroup (this has to be left for future research).

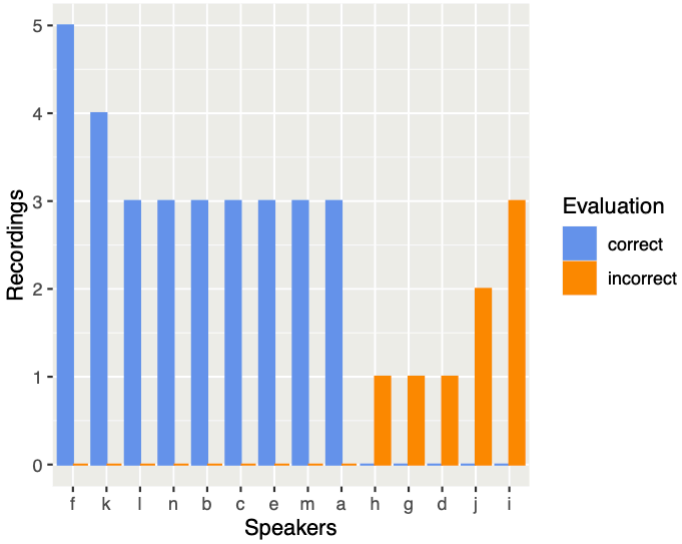


Figure 15:
Results of the narrow evaluation of the dative recordings sorted by speaker and correctness of the evaluation

As with the broad evaluation, speaker variation was clearly visible in the narrow evaluation as well: Figure 15 shows a precise division between two groups of speakers.[14] The subgroup {j, i, h, g, d} does not seem to use the acoustic cues to signal a prosodic boundary discussed in Section 2 and can thus not be correctly classified by the system.

CONCLUSION 6

This paper introduced a new end-to-end system, which takes a speech signal annotated with syllables as input, extracts the different acoustic cues, calculates pitch accents and prosodic phrase boundaries, and creates a representation of the data in form of a p-diagram. The information stored in the p-diagram is subsequently used by a computational LFG grammar to disambiguate the syntactically ambiguous structures in the input.

The implementation enables the traditionally text-based computational LFG grammars to process spoken language and to integrate

[14] One has to keep in mind that the data itself is greatly reduced here, with speakers h, g, d only contributing a single sentence.

the speech signal information into the analysis of linguistic phenomena, thus closing the gap between automatic speech recognition and linguistically deep computational grammars. In addition to syntactic and semantic analyses, the computational LFG grammars can now process and interpret any phenomena indicated by prosodic constituency or pitch accents. As such, they take a major step towards real automatic speech understanding.

An initial evaluation of the German system showed promising results and gave interesting insights into speaker variation. Challenges are manifold, and foremost is the problem that prosody is always gradient and includes a lot of variation (within and between speakers, but also within and between different language varieties, etc.). Syntax and semantics, in contrast, are less prone to variation and mostly rely on categorical information, which makes the communication between these modules and prosodic structure difficult. Nevertheless, the system introduced in this paper proves that an integration of spoken language into the existing computational grammars is possible and desirable in order to allow for a complete end-to-end analysis between form (the speech signal) and meaning (the semantic interpretation), and for an automatic analysis of linguistic phenomena from all relevant angles.

REFERENCES

R. Harald Baayen, Richard Piepenbrock, and Leon Gulikers (1995), The CELEX lexical database (CD-ROM), in *Linguistic Data Consortium*, University of Pennsylvania, Philadelphia.

Anton Batliner, Jan Buckow, Heinrich Niemann, Elmar Nöth, and Volker Warnke (2000), The prosody module, in Wolfgang Wahlster, editor, *Verbmobil: Foundations of speech-to-speech translation*, pp. 106–121, Springer, Berlin/Heidelberg.

Anton Batliner, Bernd Möbius, Gregor Möhler, Antje Schweitzer, and Elmar Nöth (2001), Prosodic models, automatic speech understanding, and speech synthesis: Towards the common ground, in *Proceedings of the European Conference on Speech Communication and Technology*, volume 4, pp. 2285–2288, Aalborg, Denmark.

Michael Becker, Joe Pater, and Christopher Potts (2007), OT-Help: Java tools for Optimality Theory, `http://web.linguist.umass.edu/~OTHelp/`.

Kenneth R. Beesley and Lauri Karttunen (2003), *Finite state morphology*, CSLI Publications, Stanford, CA.

Emily Bender, Dan Flickinger, and Stephan Oepen (2002), The Grammar Matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammar, in John Carroll, Nelleke Oostdijk, and Richard Sutcliffe, editors, *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, pp. 8–14.

Steven Bird (1992), Finite-state phonology in HPSG, in *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, pp. 74–80.

Steven Bird and Ewan Klein (1994), Phonological analysis in typed feature systems, *Computational Linguistics*, 20(3):455–491.

Daniel G. Bobrow, Bob Cheslow, Cleo Condoravdi, Lauri Karttunen, Tracy Holloway King, Rowan Nairn, Valeria de Paiva, Charlotte Price, and Annie Zaenen (2007), PARC's Bridge and Question Answering System, in *Proceedings of the Grammar Engineering Across Frameworks Workshop (GEAF 2007)*.

Paul Boersma and David Weenink (2021), Praat: doing phonetics by computer [computer program, Version 6.1.48], available at http://www.praat.org/.

Tina Bögel (2015), *The syntax-prosody interface in Lexical Functional Grammar*, Ph.D. thesis, University of Konstanz, Konstanz.

Tina Bögel (2020), German case ambiguities at the interface: Production and comprehension, in Gerrit Kentner and Joost Kremers, editors, *Prosody in syntactic encoding*, number 573 in Linguistische Arbeiten, pp. 51–84, De Gruyter, Berlin.

Tina Bögel (2023), Prosody and its interfaces, in Mary Dalrymple, editor, *Handbook of Lexical Functional Grammar*, Language Science Press, Berlin.

Tina Bögel, Miriam Butt, Annette Hautli, and Sebastian Sulger (2007), Developing a finite-state morphological analyzer for Urdu and Hindi, in *Proceedings of the Sixth International Workshop on Finite-State Methods and Natural Language Processing (FSMNLP)*, Potsdam, Germany.

Tina Bögel and Alice Turk (2019), Frequency effects and prosodic boundary strength, in Sasha Calhoun, Paola Escudero, Marija Tabain, and Paul Warren, editors, *Proceedings of the 19th International Congress of Phonetic Sciences (ICPhS), Melbourne, Australia*, pp. 1014–1018.

Tina Bögel, Saeed Reza Yousefi, and Mahinnaz Mirdehghan (2018), Vafsi oblique pronouns: stress-related placement patterns, in Miriam Butt and Tracy Holloway King, editors, *Proceedings of LFG18*, pp. 88–108.

Kersti Börjars, Rachel Nordlinger, and Louisa Sadler (2019), *Lexical-Functional Grammar: An introduction*, Cambridge University Press, Cambridge.

Bettina Braun, Nicole Dehé, Jana Neitsch, Daniela Wochner, and Katharina Zahner (2019), The prosody of rhetorical and information-seeking questions in German, *Language and Speech*, 62(4):779–807.

Norbert Braunschweiler (2003), *Automatic detection of prosodic cues*, Ph.D. thesis, University of Konstanz.

Norbert Braunschweiler (2006), The prosodizer – automatic prosodic annotations of speech synthesis databases, in *Proceedings of Speech Prosody*, Dresden, Germany, doi:10.21437/SpeechProsody.2006-136.

Joan Bresnan, Ash Asudeh, Ida Toivonen, and Stephen Wechsler (2016), *Lexical-Functional Syntax*, Wiley-Blackwell, Malden, MA, 2 edition.

Miriam Butt, Helge Dyvik, Tracy H. King, Hiroshi Masuichi, and Christian Rohrer (2002), The parallel grammar project, in *Proceedings of the COLING-2002 Workshop on Grammar Engineering and Evaluation*.

Miriam Butt, Farhat Jabeen, and Tina Bögel (2020), Ambiguity resolution via the syntax-prosody interface: The case of *kya* 'what' in Urdu/Hindi, in Gerrit Kentner and Joost Kremers, editors, *Prosody in Syntactic Encoding*, volume 573 of *Linguistische Arbeiten*, pp. 85–118, De Gruyter, Berlin.

Miriam Butt and Tracy Holloway King (1998), Interfacing phonology with LFG, in Miriam Butt and Tracy Holloway King, editors, *Proceedings of LFG98*, pp. 1–17, CSLI Publications, Stanford, CA.

Miriam Butt, Tracy Holloway King, María-Eugenia Niño, and Frédérique Segond (1999), *A grammar writer's cookbook*, CSLI Publications, Stanford, CA.

Stephen Clark and James R. Curran (2007), Wide-coverage efficient statistical parsing with CCG and log-linear models, *Computational Linguistics*, 33(4):494–552.

Ann Copestake (2002), *Implementing typed feature structure grammars*, CSLI Publications, Stanford, CA.

Richard Crouch, Mary Dalrymple, Ronald M. Kaplan, Tracy H. King, John T. Maxwell III, and Paula Newman (2017), *XLE documentation*, Palo Alto Research Center, Palo Alto, CA, online documentation.

Mary Dalrymple, editor (2023), *The handbook of Lexical Functional Grammar*, number 13 in Empirically Oriented Theoretical Morphology and Syntax, Language Science Press, Berlin, doi:10.5281/zenodo.10037797.

Mary Dalrymple, John J. Lowe, and Louise Mycock (2019), *The Oxford reference guide to Lexical Functional Grammar*, Oxford University Press, Oxford.

Denys Duchier, Joseph Le Roux, and Yannick Parmentier (2004), The metagrammar compiler: An NLP application with a multi-paradigm architecture, in *Proceedings of the 2nd Oz-Mozart conference, MOZ*, Charleroi.

Claire Gardent and Yannick Parmentier (2005), Large scale semantic construction for tree adjoining grammars, in Philippe Blache, Edward Stabler, Joan Busquets, and Richard Moot, editors, *Proceedings of Logical Aspects of Computational Linguistics*, pp. 131–146, Bordeaux.

Anja Gollrad, Esther Sommerfeld, and Frank Kügler (2010), Prosodic cue weighting in disambiguation: case ambiguity in German, in *Proceedings of Speech Prosody*, Chicago, doi:10.21437/SpeechProsody.2010-178.

Simon Gonzalez, James Grama, and Catherine E. Travis (2020), Comparing the performance of forced aligners used in sociophonetic research, *Linguistics Vanguard*, 6(1):1–13.

Martine Grice and Stefan Baumann (2002), Deutsche Intonation und GToBI, *Linguistische Berichte*, 191:267–298.

Carlos Gussenhoven (2008), Types of focus in English, in Chungmin Lee, Matthew Gordon, and Daniel Büring, editors, *Topic and focus*, volume 82 of *Studies in Linguistics and Philosophy*, pp. 83–100, Springer, Dordrecht.

Aravind Joshi (2003), Tree-adjoining grammar, in Ruslan Mitkov, editor, *Handbook of computational linguistics*, Oxford University Press, Oxford.

Ronald M. Kaplan and Joan Bresnan (1982), Lexical-Functional Grammar: A formal system for grammatical representation, in Joan Bresnan, editor, *The mental representation of grammatical relations*, pp. 173–281, MIT Press, Cambridge, MA.

Ronald M. Kaplan, John T. Maxwell III., Tracy Holloway King, and Richard Crouch (2004), Integrating finite-state technology with deep LFG grammars, in Erhard Hinrichs and Kiril Simov, editors, *Proceedings of the ESSLLI Workshop on Combining Shallow and Deep Processing for NLP*, pp. 11–20.

Thomas Kisler, Uwe Reichel, and Florian Schiel (2017), Multilingual processing of speech via web services, *Computer Speech and Language*, 45:326–347.

Ewan Klein (2000), A constraint-based approach to English prosodic constituents, in *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 217–224.

Manfred Krifka (2008), Basic notions of information structure, *Acta Linguistica Hungarica*, 55(3–4):243–276.

D. Robert Ladd (1984), Declination: A review and some hypotheses, *Phonology*, 1:53–74.

Géraldine LEGENDRE, Yoshiro MIYATA, and Paul SMOLENSKY (1990), Harmonic grammar – a formal multi-level connectionist theory of linguistic well-formedness: An application, in *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*, pp. 388–395, Lawrence Erlbaum, Cambridge, MA.

Ilse LEHISTE, Joseph P. OLIVE, and Lynn A. STREETER (1976), Role of duration in disambiguating syntactically ambiguous sentences, *The Journal of the Acoustical Society of America*, 60:1199–1202.

Willem J.M. LEVELT, Ardi ROELOFS, and Antje S. MEYER (1999), A theory of lexical access in speech production, *Behavioral and Brain Sciences*, 22:1–75.

Moritz MESSMER and Mark-Matthias ZYMLA (2018), The Glue Semantics work bench: A modular toolkit for exploring Linear Logic and Glue Semantics, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of LFG18*, pp. 268–282, CSLI Publications, Stanford, CA.

Gerald PENN and Bob CARPENTER (1999), ALE for speech: A translation prototype, in *Proceedings of the 6th European Conference on Speech Communication and Technology (EUROSPEECH'99)*, pp. 947–950, Budapest, Hungary.

Karin PITTNER (2014), Ist der Dativ dem Genitiv sein Tod? Funktionen und Konkurrenzformen von Genitiv-NPs im heutigen Deutsch (Is the dative the death of the genitive? The function and competition of Genitive-NPs in Modern German), in Corinna REUTER and Anne-Kathrin SCHLIEF, editors, *Linguistische und sprachdidaktische Aspekte germanistischer Forschung Chinesisch-Deutsch*, pp. 41–56, Peter Lang, Frankfurt am Main.

Carl J. POLLARD and Ivan A. SAG (1994), *Head-driven Phrase Structure Grammar*, University of Chicago Press, Chicago.

Chris POTTS, Joe PATER, Karen JESNEY, Rajesh BHATT, and Michael BECKER (2010), Harmonic grammar with linear programming: From linear systems to linguistic typology, *Phonology*, 27(1):77–117.

Patti PRICE, Mari OSTENDORF, Stefanie SHATTUCK-HUFNAGEL, and Cynthia FONG (1991), The use of prosody in syntactic disambiguation, *Journal of the Acoustical Society of America*, 90(6):2956–2970.

Alan PRINCE and Paul SMOLENSKY (2004), *Optimality Theory: Constraint interaction in generative grammar*, Blackwell, Oxford, Malden, MA.

R CORE TEAM (2016), *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria.

Mats ROOTH (2016), Alternative semantics, in Caroline FÉRY and Shinichiro ISHIHARA, editors, *Oxford handbook of information structure*, pp. 19–40, Oxford University Press, Oxford.

Florian Schiel (1999), Automatic phonetic transcription of non-prompted speech, in John J. Ohala, Yoko Hasegawa, Manjari Ohala, Daniel Granville, and Ashlee C. Bailey, editors, *Proceedings of the 14th International Congress of Phonetic Sciences (ICPhS), San Fracisco, CA, USA*, pp. 607–610, San Francisco.

Antje Schweitzer and Bernd Möbius (2009), Experiments on automatic prosodic labeling, in *Proceedings of INTERSPEECH*, pp. 2515–2518, Brighton, UK.

Alan K. Scott (2011), Everyday language in the spotlight: The decline of the genitive case, *German as a Foreign Language*, 1(1):53–70.

Elisabeth O. Selkirk (2011), The syntax-phonology interface, in John Goldsmith, Jason Riggle, and Alan Yu, editors, *The handbook of phonological theory*, pp. 435–484, Blackwell, Malden, MA.

Kim Silverman, Mary Beckman, John Pitrelli, Mari Ostendorf, Colin Wightman, Patti Price, Janet Pierrehumbert, and Julia Hirschberg (1992), ToBI: A standard for labeling English prosody, in *Proceedings of the 1992 International Conference on Spoken Language Processing*, pp. 867–870, Banff.

Mark Steedman (2000), *The syntactic process*, MIT Press, Cambridge, MA.

Sebastian Sulger, Miriam Butt, Tracy Holloway King, Paul Meurer, Tibor Laczkó, György Rákosi, Cheikh Bamba Dione, Helge Dyvik, Victoria Rosén, Koenraad De Smedt, Agnieszka Patejuk, Ozlem Cetinoglu, I Wayan Arka, and Meladel Mistica (2013), Pargrambank: The pargram parallel treebank, in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 550–560, Association for Computational Linguistics, Sofia, Bulgaria.

Bruce Tesar and Paul Smolensky (1998), Learning optimality-theoretic grammars, *Lingua*, 106:161–196.

Hubert Truckenbrodt (1999), On the relation between syntactic phrases and phonological phrases, *Linguistic Inquiry*, 30(2):219–255.

Wolfgang Wahlster (2013), *Verbmobil: Foundations of speech-to-speech translation*, Springer, Berlin/Heidelberg.

John C. Wells (1997), SAMPA computer readable phonetic alphabet, in Dafydd Gibbon, Roger Moore, and Richard Winski, editors, *Handbook of standards and resources for spoken language systems*, pp. 684–732, Mounton de Gruyter, Berlin, New York.

Bodo Winter (2019), *Statistics for linguists: An introduction using R*, Routledge, New York London.

Ian H. Witten and Eibe Frank (2005), *Data mining: Practical machine learning tools and techniques*, Morgan Kaufmann, San Francisco, USA, 2nd edition.

Yi Xu (2005), Speech melody as articulatorily implemented communicative functions, *Speech Communication*, 46:220–251.

Yi Xu (2013), ProsodyPro – A tool for large-scale systematic prosody analysis, in Brigitte Bigi and Daniel Hirst, editors, *Tools and resources for the analysis of speech prosody*, pp. 7–10, Laboratoire Parole et Langage, France.

Yi Xu and Ching X. Xu (2005), Phonetic realization of focus in English declarative intonation, *Journal of Phonetics*, 33:159–197.

Kristine M. Yu (2018), Advantages of constituency: Computational perspectives on word prosody in Samoan, in Annie Foret, Reinhard Muskens, and Sylvain Pogodalla, editors, *Formal Grammar (FG 2017)*, pp. 105–124, Springer, Berlin/Heidelberg.

Marzena Żygis, John M. Tomlinson, Caterina Petrone, and Dominik Pfütze (2019), Acoustic cues of prosodic boundaries in German at different speech rates, in Sasha Calhoun, Paola Escudero, Marija Tabain, and Paul Warren, editors, *Proceedings of the 19th International Congress of Phonetic Sciences (ICPhS), Melbourne, Australia*, https://hal.science/hal-02097707.

Tina Bögel
0000-0001-5644-3730
Tina.Boegel@uni-konstanz.de

Department of Linguistics
Universitätsstraße 10,
78464 Konstanz,
Germany

Tianyi Zhao
0009-0000-5258-4069
Tianyi.Zhao@uni-konstanz.de

Department of Linguistics
Universitätsstraße 10,
78464 Konstanz,
Germany

Tina Bögel and Tianyi Zhao (2025), *From speech signal to syntactic structure: A computational implementation*, Journal of Language Modelling, 13(1):1–42
https://dx.doi.org/10.15398/jlm.v13i1.397