Learning a Lexicalized Grammar for German

Sandra Kübler
Computational Linguistics
Gerhard-Mercator Universität Duisburg, Germany

s.kuebler@uni-duisburg.de

Abstract

In syntax, the trend nowadays is towards lexicalized grammar formalisms. It is now widely accepted that dividing words into wordclasses may serve as a labor-saving mechanism - but at the same time, it discards all detailed information on the idiosyncratic behavior of words. And that is exactly the type of information that may be necessary in order to parse a sentence. For learning approaches, however, lexicalized grammars represent a challenge for the very reason that they include so much detailed and specific information, which is difficult to learn. This paper will present an algorithm for learning a link grammar of German. The problem of data sparseness is tackled by using all the available information from partial parses as well as from an existing grammar fragment and a tagger. This is a report about work in progress so there are no representative results available yet.

1. Introduction

When looking at the most recent advances in syntax theory, one will notice a definite tendency towards lexicalized approaches. Simple context-free grammar formalisms may be easy to handle but they lack the descriptive power to model idiosyncrasies in the syntactic behavior of single words.

In the natural language learning community, probabilistic approaches play a dominant role. Yet probabilistic learning has its strength in finding major trends in the training data. An idiosyncratic behavior of a single word is very likely to go unnoticed for lack of data. This divergence in interest might be the reason why hardly any attempt was made to have a lexicalized grammar learned.

In this paper, I will describe an approach to learning a link grammar. Link grammar (Sleator & Temperley 1991) is highly lexicalized, and therefore the problem of data sparseness will be immense. As a consequence, I have chosen a fuzzy representation. The fuzziness in this case models uncertainty rather than vagueness inherent in the language. The learning algorithm tries to extract as much information as possible from a grammar fragment, partial parses provided by this grammar, and wordclass information (for unknown words or to corroborate decision made by the system).

2. Link Grammar

Link grammar (Grinberg, Lafferty & Sleator, 1995; Sleator & Temperley 1991) is a highly lexical, context-free formalism that does not rely on constituent structure. Instead, it models connections between word pairs without building a hierarchical structure.

The link grammar formalism is best explained with an example of a linkage (i.e. a link grammar parse): Figure 1 shows a linkage for an English sentence. A linkage is a graph in which the vertices, representing the words, are connected by labeled arcs. These arcs are called links. For a grammatically correct sentence, the linkage must fulfill the following requirements: the links do not cross (= planarity), the graph is connected, and at most one arc connects a pair of words. If there is no linkage for a sequence of words, the sentence is not in the language modeled by the grammar.

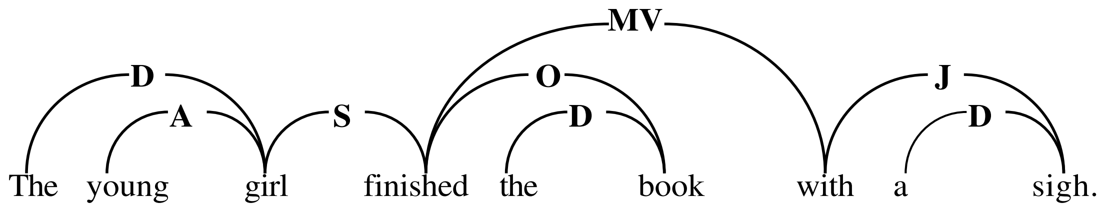


Figure 1: A link grammar parse

Sandra Kübler (1998) **Learning a Lexicalised Grammar for German.** In D.M.W. Powers (ed.) *NeMLaP3/CoNLL98: New Methods in Language Processing and Computational Natural Language Learning,* ACL, pp 11-18.

The labels on the arcs denote the syntactic relations or constituent relationships of the connected words. In figure 1, the link labeled S connects the subject noun to the finite verb, D connects determiners to their nouns, MV connects the verb to the following prepositional phrase, etc.

The grammar itself consists of a wordlist in which each word is paired up with all potential linking requirements. Each linking requirement models one usage of the word. A linking requirement, also called a disjunct, is a formal specification of the different connectors, which link with a matching connector of another word, including their direction and order. It is usually represented as a pair of ordered lists: the left list, containing connectors that link to the left of the word, and the right list, containing connectors that link to the right. For example, the linking requirement of the word "girl" in figure 1 is characterized by the formula ((D, A), (S)), for "finished" the formula is ((S) (O, MV)), and for "young" ((), (A)). In a more sophisticated version of the grammar, the labels are annotated by features, e.g. to ensure agreement between subject and verb.

The link grammar formalism is similar to dependency grammar (Mel'cuk 1988, Tesnière 1959) in that both of them model connections between single words. But link grammar connections are purely lexical: they do not intend to model valency or semantic aspects of words. An additional advantage of link grammar is that there exists an efficient parsing algorithm (Sleator & Temperley 1991, 1996) whereas there does not seem to exist one for dependency grammar.

2.1. Adaptations of the Formalism to Cover the German Language

Link grammar, like many other formalisms, seems to be especially suited for the English language. When trying to use this formalism for other languages, it seems wise to adapt the formalism to the needs of these languages, most of which are caused by a freer word order. In working with the German language, I have found the following changes immensely helpful:

Sleator and Temperley (1991) strongly prefer local links (i.e. links connecting words to their immediate neighbors), even if this is not supported by linguistics. As German uses agreement much more extensively than English, it is necessary to link words according to the agreement requirements rather than because of immediate neighborhood. This approach results in considerably more long distance links.

In English, the word order is rigidly determined for most parts of the sentence. Sleator and Temperley (1991) use different labels for links that can occur in more than one position (e.g. adverbs) depending on whether they are left or right links. In German, however, due to its freer word order, these phenomena are relatively common. In order to avoid using too many different labels describing the same kind of link but in different order, I have introduced the idea of control, or rather directionality of links. Each link is marked as either controller (§) or controlled (=). I can thus use the S-link for subjects preceding or following the finite verb, as shown in figure 2.

The principle of planarity states that links in a linkage must not cross. Sleator and Temperley (1991, 1) comment that most sentences of most languages adhere to that principle. Unfortunately, German is one of the languages in which this principle is violated in a number of cases. Some of them are caused by the free word order, some by phenomena like the splitting of the verb:

Ihnen	wird	vorgeworfen,	sie	hätten
to them	is	reproached,	they	had

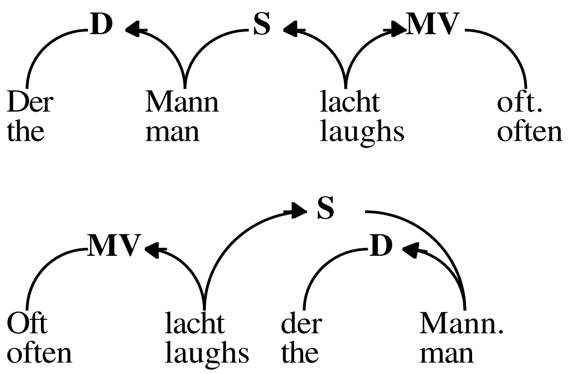
sich	in	Berlin	getroffen	.
each other	in	Berlin	met	.

They were reproached for having met in Berlin.

Ich	habe	den	Mann	gesehen,	der
I	have	the	man	seen,	who

das	Buch	besitzt	.
the	book	owns	.

I have seen the man who owns the book.



Grammar:

der	((), (=D))
Mann	((§D), (=S)), ((=S, §D), ())
lacht	((§S), (§MV)), ((§MV), (§S))
oft	((=MV), ()), ((), (=MV))

Figure 2: Controlled links

In the first example, the dative object "ihnen" links to "vorgeworfen" and the finite verb "wird" to the period. In the second example, "Mann" links to "der", the relative pronoun and the finite verb to the past participle "gesehen".

As crossing links are inevitable in German, there is a special marker for such links that may cross.

2.2. What Advantages Does Link Grammar Offer for Learning?

Link grammar offers at least two characteristics that will be of advantage in syntax learning:

Instead of relying on a hierarchical constituent structure, the link grammar formalisms is based on links on a single level. Therefore, they can be learned independently; there is no need for a top-down or bottom-up structuring. Thus errors in earlier steps of building the structure cannot have as disastrous effects as with constituent structures.

Another problem of constituent grammars, which may cause problems in learning, are long-distance dependencies. The information about a gap somewhere in the structure is usually passed on through several levels of the constituent tree. In link grammar, however, these distances are covered by a direct link, which means that these phenomena do not need any special attention during the learning process.

2.3. Former Approaches to Learning Link Grammar

There already exist two approaches to learning with a link grammar formalism (Della Pietra et al., 1994; Fong & Wu, 1995). In both cases, the probabilistic version of the grammar (Lafferty, Sleator & Temperley, 1992) are used and the word pairs plus their probabilities are inferred from a corpus by an EM-algorithm. The probabilistic model of link grammar restricts disjuncts in that only one left connector and at most two right connectors are allowed. At least for German, this formalism leads to a very unnatural and counterintuitive description.

Additionally, to reduce the amount of data to be processed, both approaches did not use the link type information but assumed only one type of link. This restriction may be very helpful concerning computing time yet thus valuable information is not taken into consideration.

3. A Fuzzy Relation for Representing the Link Grammar

Ever since Zadeh (1965) has introduced fuzzy sets, the interest in fuzzy modeling has increased steadily. In computational linguistics, fuzzy methods are mainly used in semantics to model vague meaning like the meaning of the concept "fast". A fuzzy set representing this concept would give gradually increasing grades of membership to the speed between 0 and 120 mph.

However, fuzzy methods cannot only be used for modeling vagueness, they are also useful in cases where the given information is either inexact or incomplete. Concerning grammar, and especially learning grammar, the latter case must be assumed.

A (complete) link grammar can be represented as a (crisp) relation G among the set W of all words and the set D of all potential disjuncts

$$G : W \times L \rightarrow \{0,1\}$$

with its characteristic function

$$\mu_G(w,d) = \begin{cases} 1 & if \quad < w,d > \quad is \quad grammatical \\ 0 & else \end{cases}$$

where an ordered pair $<w, d>$ is assigned the membership value 1 if d is a valid linkage for the word w .

Now if only a fragment of the grammar is known, the fuzzy relation G* is defined as

$$G^* : W \times L \rightarrow [0,1]$$

where the membership value does not indicate whether the ordered pair is in the grammar but whether the pair is known to be in the grammar or to what degree it is assumed to be in the grammar (for the characteristic function see section 4.1). Here the value 1 indicates that it is certain that the linkage is valid for the word in question, 0 indicates that there has never been any reason to assume that w takes d as a valid linkage.

4. Learning the Link Grammar

The system starts with a grammar fragment extracted from a small corpus of 50 annotated sentences. These sentences, as well as the test sentence used below, are taken from the TAZ, a German newspaper. At this stage, the grammar is crisp, i.e. the only membership values used are 1 for pairs of words and disjuncts found in the corpus and 0 otherwise. Then optional elements are marked, i.e. if a word is connected to two disjuncts d and d' of which d is equal to d' except that d has one ore more connector that are not in d', then these connectors are marked as optional.

The learning process itself is incremental: once a new sentence is presented to the system, the parsing component takes over. It attempts to parse the sentence with the crisp version of the grammar, i.e. with all pairs of words and disjuncts for which the relation G* gives the value 1. (At the moment, the parser still has to be implemented. The algorithm is described by Sleator and Temperley (1991,1996) yet it must be modified to account for the changes in the link grammar formalism necessary to describe German.) If the first attempt with the crisp grammar does not succeed, the threshold for G* is lowered from 1 to 0.3 and the attempt is repeated. In this case, less reliable information is used but if the parse succeeds, the validity of the disjuncts used in the parse is corroborated. Therefore their membership value is increased.

If the parser, however, does not succeed in parsing the sentence, the learning component is called:

• As a first step, every word in the sentence is tagged. (The formalism used for tagging will be Brill's (1993, 1995) transformation-based error-driven tagger.) Unlike other approaches to learning using constituent-based grammars, this system does not use the wordclass information to restrict the roles, a word can play in the parse. Rather it takes this information as a starting point in the search for potential disjuncts for unknown words. And if a new disjunct is found for a word already in the grammar, its credibility is tested by comparing the word's wordclass to the wordclass of the word with which the disjunct has the highest membership value in the grammar (cf. below). In both cases, the wordclass information is only used to corroborate decisions made in advance.

• After the wordclass information is provided, the systems looks for every potential conjugated verb in the sentence. For each of these verbs, a partial linkage is constructed, in which the verb is connected to the period by an Xp-link. This is an important step as the Xp-link cannot be crossed by any other link added later in the process.

• Then for all words listed in the grammar, the system retrieves all disjuncts which are connected to them. With these disjuncts, all potential partial linkages are constructed by linking all words which possess matching connectors. If word x, for example, possesses a disjunct with a connector =Jd-, it will be linked to word y possessing a disjunct with connector §Jd+. All these links must fulfill the conditions that they must not cross, that the order of connectors in the disjunct must not be changed, and that no two links can connect the same pair of words.

• In the next step, every disjunct in the partial parse which is activated (i.e. partially filled) attempts to fill the remaining connectors by linking them to neighboring words without violating the restrictions mentioned above. Like in the previous steps, all potential combinations are stored.

• After that, all words for which linking information is available but which are not yet connected to the partial parse are linked in any possible way.

• If the linkage is not connected at this stage, the words left out are either unknown or the disjunct needed for this sentence has not been recorded for them yet. Starting with an initial corpus of only 50 sentences, this will be the case for about 90% of the sentences. But even if the grammar fragment is increased considerably, it will be highly probable that most linkages are not connected at this stage. As the disjuncts needed to complete the linkage, or at least very similar ones, may already be included in the grammar, it is necessary to have an efficient retrieval function. In order to reduce the search space, the wordclass information is used to find entries with similar linking requirements. All the disjuncts found in this search are then given to the unknown word as potential disjuncts. They are then used to complete the linkage.

• At this stage in the process, the learner has aggregated a number of complete linkages. The next task must then be to evaluate them. This is done by the following method: First the membership value for each word and the disjunct used in the linkage is calculated (cf. section 4.1). This is not as trivial as it may seem as for many words, the disjuncts actually used in the linkage are different form those originally retrieved from the grammar. If connector could not be filled, they are dropped, while other connectors which originate from the linking requirements of another word are added. From these membership values of the single words, the overall value of the linkage is calculated as the arithmetic mean. This final figure is used as a measure of the quality of the linkage.

• The best parse then is given as the preferred parse for the input sentence, and all new pairs of words and disjuncts are added to the grammar with their calculated membership values. For pairs already in the fuzzy grammar, the membership value is increased.

• As a last step, for every new or modified word, optional elements are marked in the disjuncts.

4.1. Calculating the Membership Value

The following algorithm is used to calculate the membership value μ(w,d) for the pair <w, d>.

if (w $\in$ G*):
  if <w, d> $\in$ G*
  then $\mu(w,d) = \mu_{G*}(w,d)$
  else get the pair <w',d'> with wordclass(w) == wordclass(w') and minimal distance(d, d') and maximal $\mu_{G*}(w',d')$ then
  $\mu(w,d) = \mu_{G*}(w',d') - 0.1 - \text{distance}(d,d')$
if (w $\notin$ G*):
  if ((d $\in$ G*) $\wedge$ maximal $\mu_{G*}(w',d) \wedge$ (wordclass(w) == wordclass(w')))
  then $\mu(w,d) = \mu_{G*}(w',d) - 0.1$
  if ((d $\in$ G*) $\wedge$ maximal $\mu_{G*}(w',d') \wedge$ (wordclass(w) $\neq$ wordclass(w')))
  then $\mu(w,d) = \dfrac{\mu_{G*}(w',d)}{2}$
  if (d $\notin$ G*)
  then get the pair <w',d'> with (wordclass(w) == wordclass(w')) and minimal distance(d, d') and maximal $\mu_{G*}(w',d')$, then
  $\mu(w,d) = \mu_{G*}(w',d') - 0.1 - \text{distance}(d,d')$

Table 1: The grammar available for the example sentence

aber	((=E), ()), ((=CC, §Xk), (§Cd)), ((), (=E))
von	((), (§Jd, =MVp)), ((), (§Jd, =Yz, =MVp)), ((=MVp), (§Jdp)), ((=Mp), (§Jd)), ((MVpv), (§Jd))
einer	((), (=Dsfdn)), ((=Ons), (§GEp+))
Fehlernährung	
können	((§MVp), (§Sp1, §In, §Xk, =Coq)), ((§Sp1), (§In, §Xk, §COq)), ((§RSrp3), (§In))
wir	((=Sp1), ()), ((), (=Sp1))
heute	((), (=E))
schon	((), (§EBs)), ((), (=E))
sprechen	((§MVp, §E, §MVp), (=In))
.	((=Xp), ())

$$\mathrm{distance}(d,d') = \sum_{\forall c \in d,d'} \begin{cases} 0 & \text{if} \quad (c \in d) \wedge (C \in d') \\ 0.05 & \text{if} \quad \text{only features(d)} \neq \text{features(d')} \\ 0.1 & \text{if} \quad \text{control(d)} == \text{'§'} \\ 0.2 & \text{if} \quad \text{control(d)} == \text{'='} \end{cases}$$

Exception: Nothing is added if the connector c is the same as the preceding connector and the connector can be found in G* at least once marked for multiple occurrence.

The reason why the disjunct is punished harder for missing controlled links is that optional connectors usually are controlling.

4.2. Example

In this section, we will look at an example sentence. It will not be possible to give all the potential linkages but the gist of the argument should become clear.

The example sentence is:

Aber	von	einer	Fehlernährung	können	wir
but	of	a	malnutrition	can	we

heute	schon	sprechen	.
today	already	speak	.

Table 1 gives the information that can be extracted from the initial grammar G*. All the disjuncts listed for a word have the membership value 1 concerning this word. As can be seen in the table, there is only one unknown word in the sentence. However, only for the words "von", "einer", "wir", "heute", and "schon", the needed disjunct is listed. All words belonging to an open wordclass except "wir" give only partial or no information needed for this sentence.

1. step: The only wordclass information needed in the further process is that "Fehlernährung" is a noun, and "können" and "sprechen" are potential verbs.

2. step: As we know from step 1, both "können" and "sprechen" are verbs. So there are two ways to place the first link, linking each verb in turn to the period by an Xp-link.

3. step: For the information given in G*, see table 1. Three potential linkages are shown in figure 3. For each given linkage, there is another one differing only by linking "schon" instead of "heute" to "sprechen".

4. step: There are too many possibilities to link the remaining connectors of activated disjuncts to their neighbor. Figure 4 shows three of them, randomly chosen.

5. step: In figure 5, only two potential linkages are given after the remaining words are connected, the overall membership value for these linkages is calculated in step 7.

6. step: This step is not necessary because the linkage is complete.

7. step: The calculations for the linkages represented in figure 5 are given in table 2 and 3 respectively.

8. step: The disjuncts from table 2 for the words "aber", "Fehlernährung", "können", and "sprechen" with their membership values are added to the grammar.

9. step: There are two new disjuncts which can be marked for optional connectors: For the word "aber", the new disjunct is (({=CC}, {=Xk}), (§Cd)), and for "können" (({=Cd}, §MVp), (§Sp1, §In, {Xp})).

5. Future Work

There is still so much work to do that it is hard to decide what should be done first. The most important task is certainly the implementation of the algorithm and the parser. This will hopefully be finished for the presentation so that at least sample results can be given.

Another important task will be to increase the size of the corpus from which the grammar fragment is extracted. The more information is available to the learning component, the better the judgment on the best links will be. Another way to improve the choice and evaluation of new disjuncts will be to include co-occurrence information into the calculation of the membership value of a disjunct. If, for example, the connector §Xp+ is accompanied by an S-link in the majority of cases, a new disjunct including both connectors should be valued more confidently than one which does not.

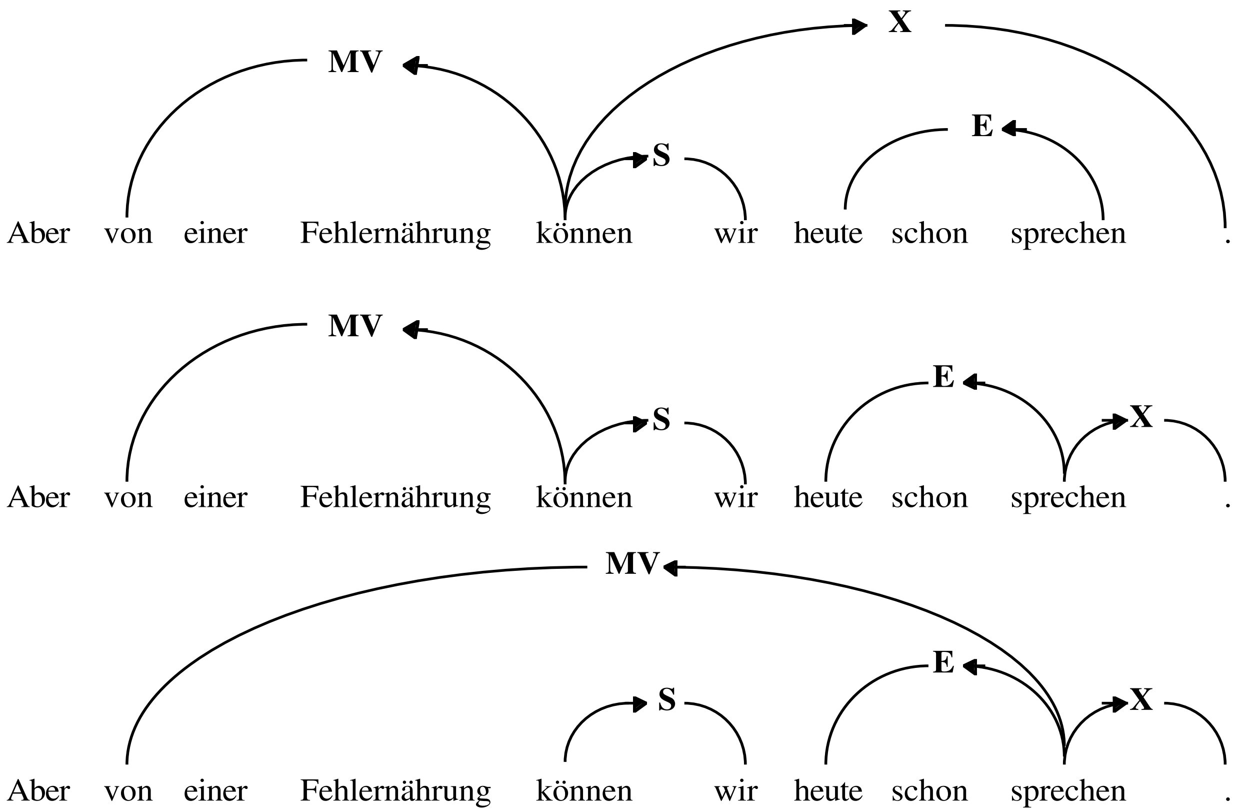
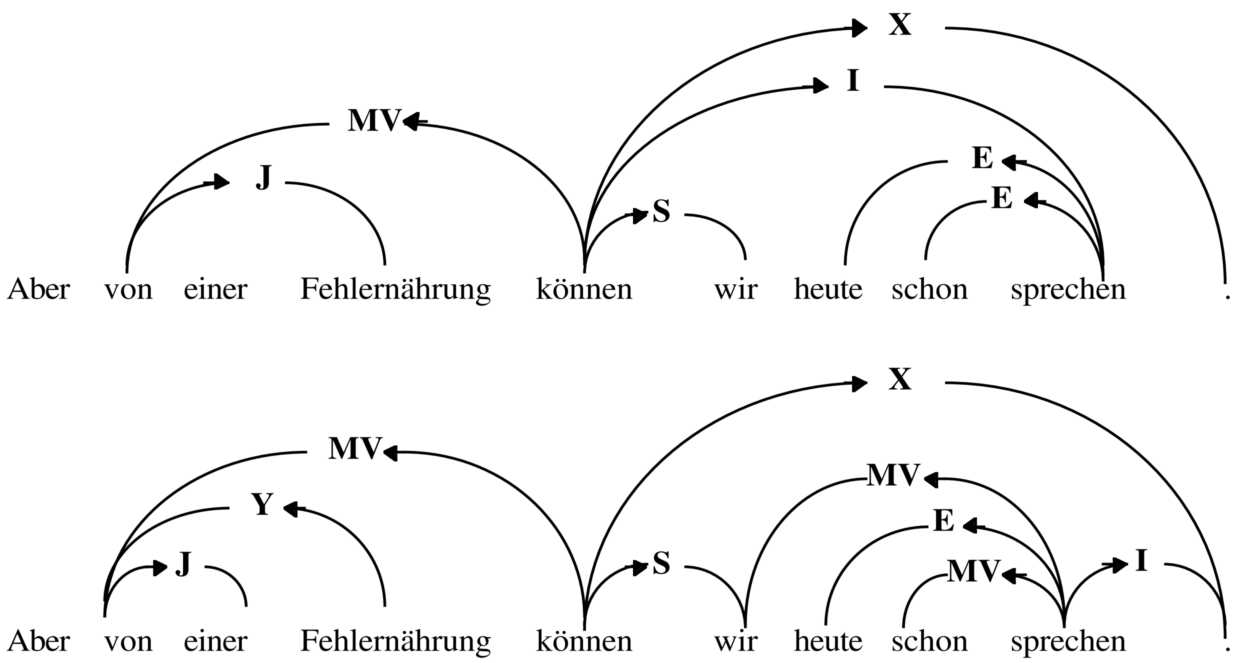


Figure 3: Potential partial linkages after step 3



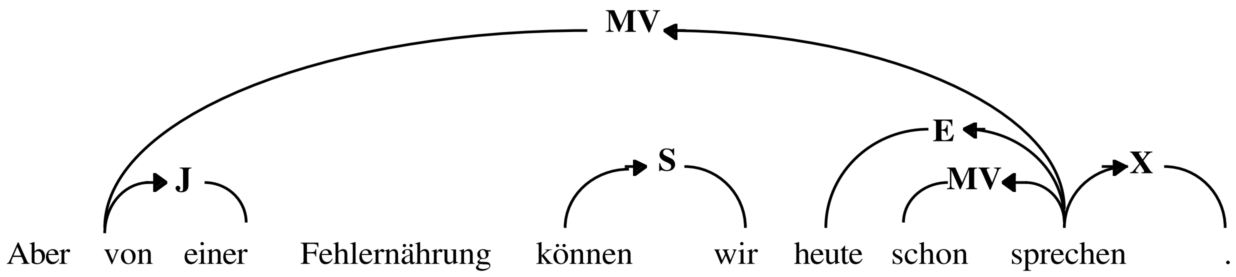


Figure 4: Potential partial linkages after step 4

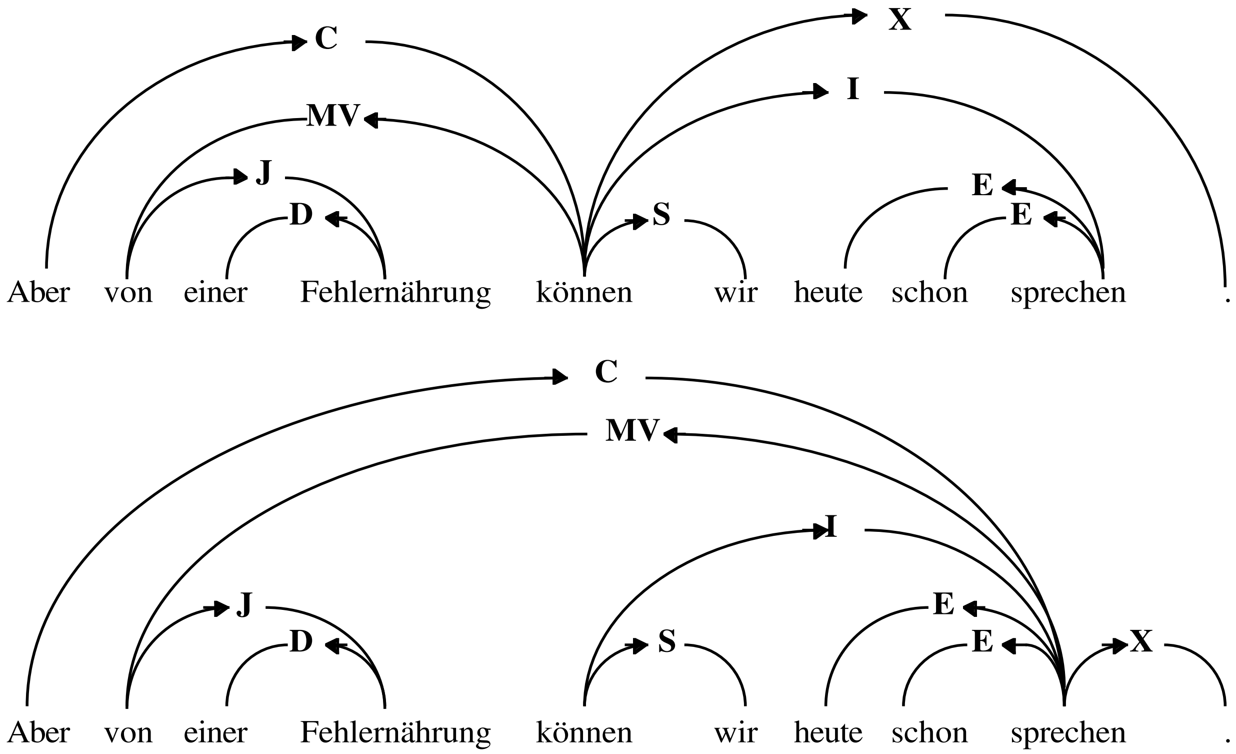


Figure 5: Potential linkages after step 5

Table 2: The evaluation of the disjuncts for the first linkage

word	disjunct	value	comment
aber	((), (§Cd))	0.9	((), (§Cd)) ∈ G*
von	((), (§Jd, =MVp))	1	
einer	((), (=Dsfdn))	1	
Fehlernährung	((=Jd, §Dsfdn), ())	0.9	((=Jd, §Dsfdn), ()) ∈ G*
können	((=Cd, §MVp), (§Sp1, §In, §Xp))	0.75	most similar disjunct in G*: ((§MVp), (§Ss3, §In, §Xp))
wir	((=Sp1), ())	1	
heute	((), (=E))	1	
schon	((), (=E))	1	
sprechen	((=In, §E, §E), ())	0.8	((=In, §E), ())∈ G*
.	((=Xp), ())	1	
	arithmetic mean =	0.93	

Table 3: The evaluation of the disjuncts for the second linkage

word	disjunct	value	comment
aber	((), (§Cd))	0.9	((), (§Cd)) ∈ G*
von	((), (§Jd, =MVp))	1	
einer	((), (=Dsfdn))	1	
Fehlernährung	((=Jd, §Dsfdn), ())	0.9	((=Jd, §Dsfdn), ()) ∈ G*
können	((), (§Sp1, §In))	0.7	most similar disjunct in G*: ((=Cd, §EF), (§Sp1, §In+))
wir	((=Sp1), ())	1	
heute	((), (=E))	1	
schon	((), (=E))	1	
sprechen	((=Cd, §MVp, =In, §E, §E), (§Xp))	0.4	most similar disjunct in G*: ((=Cd, §MVp), (§Ss1, §In, §Xp))
.	((=Xp), ())	1	
	arithmetic mean =	0.89	

6. References

Brill, E. (1993). A Corpus-Based Approach to Language Learning (Ph.D. thesis). Philadelphia: University of Pennsylvania, Department of Computer and Information Science.

Brill, E. (1995). Transformation-based tagger, version 1.14. ftp://blaze.cs.jhu.edu/pub/brill/Programs/RULE_BASED_TAGGER_V.1.14.tar.Z

Della Pietra, S. & Della Pietra, V. & Gillett, J. & Lafferty, J. & Printz, H. & Ures, L. (1994). Inference and Estimation of a Long-Range Trigram Model. In R. Carrasco & J. Oncina (Eds.), *Grammatical Inference and Applications*: *Proceedings of the Second International Colloquium, ICGI-94, Alicante, Spain* (pp. 78-92). Berlin: Springer.

Fong, E. & Wu, D. (1995). Learning Restricted Probabilistic Link Grammars. *IJCAI-95 on New Approaches to Learning for Natural Language Processing*, *Montreal, Canada*. (pp. 49-56).

Grinberg, D. & Lafferty, J. & Sleator, D. (1995*) A robust parsing algorithm for link grammars* (Tech. rep. CMU-CS-95-125). Pittsburgh, PA: Carnegie Mellon University, School of Computer Science.

Lafferty, J. & Sleator, D. & Temperley, D. (1992). Grammatical trigrams: a probabilistic model of link grammar. *Proceedings of the AAAI Conference on Probabilistic Approaches to Natural Language*. Cambridge, MA.

Mel'cuk, I. (1988). *Dependency syntax: theory and practice*. State University of New York.

Sleator, D. & Temperley, D. (1991). *Parsing English with a link grammar* (Tech. Rep. CMU-CS-91-196). Pittsburgh, PA: Carnegie Mellon University, School of Computer Science.

Sleator, D. & Temperley, D. (1996). Link grammar parser, version 2.1. ftp://ftp.cs.cmu.edu/user/sleator/link-grammar/system-2.1.tar.gz

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

Zadeh, L. (1965). Fuzzy sets. *Information and Control* 8, pp. 338-353.