

# Dependency conversion and parsing of the BulTreeBank

## Abstract

Recently dependency parsing is gaining popularity. It is broadly accepted that dependency representations are more suitable for free word order languages. Statistical dependency parsers are easy to port from one language to another, if there are dependency treebanks for learning a grammar for the particular language. However, many treebanks are based on constituency and have to be converted to dependency representations prior to learning statistical dependency parsers. In this report we investigate the issues of the conversion of the BulTreeBank (Simov et al., 2002) from Head-driven Phrase Structure Grammar (HPSG) format to dependency-based format and its parsing. We have performed three different conversions to three different dependency formats. For two of the conversions we used head tables and dependency tables which were stated explicitly, as in (Xia, 2001). For the other conversion the tables were implicitly implemented by rules. Our choice of rules for the tables was guided by decisions rooted in different linguistic theories. We have parsed the converted treebank with the Malt parser (Nivre et al., 2004) for ‘evaluating’ our conversions. Then we made error analysis to find advantages and pitfalls of each conversion strategy.

## 1. Introduction

Generally porting existing algorithms for statistical Natural Language Processing from language to language can be done with limited effort. Statistical dependency parsers are not an exception from the rule. Nevertheless, for training statistical methods we need language resources which are often annotated according to different linguistic theories and annotation schemes.

Most of the current NLP technologies were firstly developed for English and some of them were then ported to other languages. In parsing, state-of-the-art statistical parsers like those reported in (Collins, 1997) and (Charniak, 2000) were ported to Czech<sup>1</sup> (the porting of the Collins’ parser was documented in (Collins et al., 1999)). Another porting of state-of-the-art parsers, this time from English to Italian, was described in (Corazza et al., 2004). In most of the cases parsers have to be adapted to the annotation scheme of the treebank for the new language.

The most famous treebank for parser evaluation for English is undoubtedly the Penn Treebank (Marcus et al., 1993). It is constituency-based but information about heads of the phrases can be found in the most recent version of the treebank. While information about heads seems to be very useful for learning and parsing English, this kind of information is crucial for free word order languages.

Constituency notion does not seem to be very convenient for free word order languages. It can be successfully substituted by fully dependency-based approach as in (Hajič, 1998), or richly extended with head information as in HPSG-based treebanks like the BulTreeBank (Simov et al., 2002). Pure constituency treebanks can be converted to dependency, if we want to benefit from dependency representations for free word order languages.

There are several studies for parsing Bulgarian. A shallow parsing module has been used in the annotation of the BulTreeBank. Chunks have been identified with a manually constructed grammar. We should also mention the work on

constituency parsing within a larger system for text analysis for Bulgarian (Tanev, 2001) which was not evaluated on a treebank because there was not a broadly available treebank for Bulgarian at that time.

Another study (Krushkov and Chaney, 2005) reports full constituency parsing of simple sentences in Bulgarian with a grammar extracted from a small syntactically annotated collection of sentences. Evaluation was done manually and this makes the results biased and not reliable.

The first experiments on dependency parsing of Bulgarian were performed by (Marinov and Nivre, 2005). They report 84.2% unlabelled and 78.0% labelled precision on a limited subset of the BulTreeBank converted to dependency graphs. The conversion tables from that experiment were used in one of the experiments that this report describes. However, they were extended with more rules in order to cover the larger amount of data that we had.

Besides the head and dependency tables for Bulgarian that were first introduced in (Marinov and Nivre, 2005), another table and another conversion method are reported in this report. All the methods for conversion that we used are evaluated on a small set of gold standard annotation. Finally an inductive dependency parser (Nivre et al., 2004), (Nivre, 2005) is used on the converted versions of our treebank to find out which conversion has been learned and parsed best. The report is structured in the following way: In Section 2. the treebank that we used in our experiments is presented. Section 3. discusses the ‘head tables’ that we have implemented. In section 4. we present our ‘dependency tables’ and argue about the shortcomings for each of the approaches that we followed. Section 5. reports error analysis of our conversions. In the next 2 sections (6. and 7.) we briefly present the parser that we use and show our parsing results. In section 8. we conclude our work.

## 2. The BulTreeBank Annotation Scheme

Treebank annotation is of great importance to a successful cross-theoretic portability. Since for different tasks and applications various types of information are needed, it is practical for a treebank to have not entirely constituency or dependency encodings, but rather some combination of

<sup>1</sup>The interested reader can find information about the performance of various parsers for Czech on: <http://ufal.mff.cuni.cz/czech-parsing/>

both. They might be presented with different degrees of explicitness. It is important the appropriate information to be easily derivable. Having all this in mind, we pursued hybrid annotation in BulTreeBank. HPSG language model was explored. It views the linguistic data as a set of constituent structures with head-dependant markings.

Currently the treebank comprises 214000 tokens, a little more than 15000 sentences. Each token is annotated with elaborate morphosyntactic information. Additionally the Named Entities are annotated with ontological classes as person, organization, location, and other. The HPSG-based annotation scheme defines a number of phrase types which reflect both – the *constituent structure* and the *head-dependent* relation. Thus we have phrase labels with the explication of the dependent types like VPC (verbal head complement phrase), VPS (verbal head subject phrase), VPA (verbal head adjunct phrase), NPA (nominal head adjunct phrase) etc. We consider coordinations as non-headed phrases, where the grammatical function overrides the syntactic labels (Simov and Osenova, 2003). This fact causes problems if some head is always needed within the dependency relation. However, modelling coordination still remains one of the ‘tough nuts’ in all frameworks.

Behind the constituent structures and the head-dependent relations the treebank also represents phenomena like ellipsis, pro-dropness, word order, secondary predication, control. As an important mechanism for dealing with these phenomena we are using co-reference relations.

The treebank is in XML format, hence the restrictions over the language relations of dominance are encoded in a DTD. In most cases the head within phrases can be uniquely derived. For example, under the phrase VPC the head is the verb, while the complement is a nominal or a clause. Only in some combinations more specific rules are needed. For example, in NP phrases of the type NN. The head might be the former or the latter NP depending on the semantics of the phrase. In such cases manual annotation of the head is necessary.

### 3. Head Tables

We have performed three different conversions of the BulTreeBank from HPSG-based to dependency-based format. From now on we will refer to them in the following way: conversion 1 – the conversion of Svetoslav Marinov for the first ever experiments on dependency parsing of the BulTreeBank, with an extended head table by Atanas Chanev; conversion 2 – the conversion of Atanas Chanev and conversion 3 – the conversion of Kiril Simov and Petya Osenova for the CoNLL-X shared task<sup>2</sup>.

In two of our conversions from constituency to dependency representation, head tables (Xia, 2001) were used to determine the head of each constituent. For conversion 3, rules for identification of the head were applied, then all non-head daughters were made to point to the head daughter of the constituent. Once the head of each phrase of the sentence is known, the conversion approach can vary from recursively top-down as in (Daum et al., 2004) to iteratively bottom-up as in all the conversions described in this study.

Conversion procedures from constituency-based to dependency-based representation can be traced back to (Gaifman, 1965). He showed that if one knew the head daughter of each constituent in a sentence, the unlabelled dependency graph of that sentence could be easily retrieved. Information about heads is kept in a table that is known in the literature as ‘head table’. In addition to head tables, (Xia, 2001) introduced dependency tables which are needed for adding labels to the unlabelled dependency arcs of the sentence graph.

Besides in (Xia, 2001), conversions from constituency to dependency performed for English on the WSJ part of the Penn treebank have been reported in (Collins, 1997), (Yamada and Matsumoto, 2003) and (Nivre and Scholz, 2004). All these conversions benefit from a head table that consists of records containing the constituent that can have daughters, the direction of searching for the head constituent and a list of possible head constituents ordered by priority.

There are several studies in which German treebanks have been converted to dependency. Conversions have been reported in (Kübler and Telljohann, 2002) and (Ule and Kübler, 2004) for the TüBa-D treebank (Hinrichs et al., 2000). However, information about some dependencies is explicitly annotated in TüBa-D and only a few treebank specific issues have to be addressed for a successful conversion to dependency format.

The German NEGRA treebank (Skut et al., 1997) has been converted using the script DEPSY in (Daum et al., 2004). DEPSY is based on (Magerman, 1994) and implements a top-down recursive algorithm. However, the script can convert treebanks in only two formats: Penn Treebank and NEGRA treebank.

There are studies on conversion from constituency to the Prague Dependency Treebank (PDT) format. One of them is about the conversion of an English treebank (Žabokrtský and Kučerová, 2002). And another is about the conversion of an Arabic treebank (Žabokrtský and Smrž, 2003). The conversion algorithm used in these studies has been supplemented with a procedure for removing the traces from the treebanks.

In all the transformations mentioned above, except in the transformation of the TüBa-D treebank, the conversion has been performed similarly, usually in a recursive top-down fashion together with processing of treebank annotation specific constructions. Having a constituency treebank and a head table, if the mentioned algorithms are used, the resulting dependency treebank should always be the same, except in the case of the TüBa-D conversion where the whole process is strongly dependent on the treebank.

Two of our conversion methods (namely conversion 1 and 2) are very similar to the conversion method for the WSJ part of the Penn treebank. The difference in the head tables is that there is not an option for right to left search for the head among the daughters of the constituent. However, this is not a big disadvantage, because in most of the cases there is very little ambiguity which daughter to be the head. In conversion 3 the head table was substituted by rules which allow even more precise specifications for the choice of the head than the method described in (Xia, 2001). The ‘head table’ was encoded in 250 rules. Several constructions were

---

<sup>2</sup><http://nextens.uvt.nl/~conll/>

Table 1: An extract from the collection of rules used in conversion 3.

Rule	Head
AdvPA -> Adv Adv	Adv[2]
AdvPC -> Adv Adv	Adv[1]
NPA -> NPA NPA	NPA[1]
NPA -> (N NPA) (N NPA)	*[1]
NPA -> (N H) (H N)	*[1]
NPA -> N N	N[1]
NPA -> CoordP PP	CoordP
Nomin -> *	*

Table 2: An extract from the head table for conversion 1.

Constituent	Head daughter
AdvPC	<Adv> <Gerund>
Ako	<Ako> <C>
AkoP	<Ako> <C>
APA	<A> <Participle> <Pron> <Adv> <M> <Prep>
APC	<A> <Participle>
C	<C> <Prep>
CLCHE	<C>
CLDA	<T>

converted by hand.

Our head tables have 44 records for conversion 1 and 38 records for conversion 2. The differences in the head tables for the two conversions are due to different treatment of several linguistic structures, e.g. clauses. Table 4 shows the percentage of the heads from each of the conversions (the rows) that were found in every conversion and the gold standard data (the columns).

The gold standard data (last column) that we annotated ourselves is used for evaluation. It consists of 60 sentences (976 tokens). For all the other columns the training part of the BulTreeBank was used. It consists of 10911 sentences (159394 tokens). All the punctuation marks were skipped in the evaluation.

In Table 1 we give a few rules that were used in conversion 3. The first column contains the rule from the grammar used in the annotation of the BulTreeBank and its left-hand side corresponds to the constituent whose head daughter should be selected. The constituents on the right hand side of the rule are its daughters among which the head should be chosen. Relying on the second column of the table the choice can be made.

For example, the head of the mother constituent given in the first record of Table 1 is the second Adv daughter constituent. Wildcards are used with the meaning ‘no matter which constituent’ and in some cases the meaning ‘or no constituent’ can be added. This approach is different from the approach of conversion 1 and 2 in its richer possibilities of specification which daughter to be the head of the constituent. The rule from the first row of Table 1 cannot be

Table 3: An extract from the head table for conversion 2.

Constituent	Head daughter
AdvPC	<Adv> <AdvPA> <AdvPC> <Gerund> <CoordP>
Verbalised	<T> <I>
Subst	<Pron> <M> <A> <Participle>
APA	<A> <APA> <APC> <Participle> <CoordP> <Pron>
APC	<A> <APC> <APA> <Participle> <CoordP>
C	<C>
CLCHE	<V> <VPA> <VPS> <VPC> <VPF> <Participle> <CLDA> <CLCHE> <CoordP>
CLDA	<V> <VPA> <VPS> <VPC> <VPF> <CoordP>

encoded using the head tables from conversions 1 and 2.

In addition to the rules from conversion 3 we give extracts from the head tables of conversion 1 (Table 2) and 2 (Table 3). Each rule from these tables starts with a mother constituent and then the possible head daughters are given ordered by priority.

All the records from the head tables from conversion 1 and 2 can be encoded with rules like those used in conversion 3. We can do that using rules of the type <Const1> -> \* <Const2> \* which are very common in conversion 3. A record from conversions 1 and 2 has the form Const1 <Const2> <Const3> ...<ConstN> meaning that Const2 is the head daughter of Const1 and if it is not present, then Const3 is, etc. This record can be translated to the rules: <Const1> -> \* <Const2> \* – the head is Const2, <Const1> -> \* <Const3> \* – the head is Const3, ..., <Const1> -> \* <ConstN> \* – the head is ConstN.

If the rules of conversion 3 encode the same information as the head tables of conversion 1 or 2, the dependency arcs in the resulting dependency treebank will not differ from the arcs of the treebanks obtained by performing conversion 1 or 2 directly. With this we put an accent on the conversion table but not on the conversion algorithm. However, we will not prove here our assumption that the processing approach is irrelevant for a broader set of conversion algorithms, since it is beyond the scope of this report.

The differences in our conversions are not due to the different conversion methods but dependent on the different sets of head rules. Undoubtedly there are head rules that treat the same linguistic constructions differently in conversions 1, 2 and 3. This introduces different types of errors in the three converted dependency treebanks. We will discuss some of the interesting cases of erroneously converted graphs in Section 5.

Table 4: Comparison of the different conversions to one another as well as on the gold standard data.

Conv.	1	2	3	Gold
1	100%	82.18%	69.06%	62.94%
2	82.18%	100%	79.42%	74.49%
3	69.06%	79.42%	100%	70.76%

## 4. Dependency labels

### 4.1. Three sets of dependency labels

There had been three sets of dependency labels that we used in the dependency tables in conversions 1 and 2 and in the rules in conversion 3. The labels are taken from a Swedish treebank (Nilsson et al., 2005) in conversion 1 and where possible from an Italian treebank (the Turin University Treebank – TUT) (Bosco, 2004) in conversion 2. The labels in conversion 3 had been chosen according to linguistic principles more than taken from another treebank.

The labels used in conversion 1 are 14: ADV (adverbial modifier), APP (apposition), ATT (attribute), CC (coordination), DET (determiner), ID (non-first element of multi-word expression), IP (punctuation), OBJ (object), PR (complement of preposition), PRD (predicative complement), SUBJ (subject), UK (head-verb of subordinate clause dependent on complementizer), VC (verb chain), ROOT (dependent of a special root node). The labels used in conversion 1 were adapted from Swedish to Bulgarian without significant effort in (Marinov and Nivre, 2005).

The labels used in conversion 2 are generally following (Bosco, 2004) and more specifically a reduced version used in (Chanev, 2005). Although reflecting most of the basic principles of the TUT annotation scheme, the number of tags is greatly reduced to 14. The current tag set includes the tags: SUBJ (subject), OBJ (object), RMOD (adjectival or adverbial modifier, PP or relative clause), ARG (argument), INDCOMPL (locative or theme complement), EMPTYCOMPL (reflexive personal pronoun modifying verb), PREDCOMPL (predicative complement), INTERJECTION, APPPOSITION, COORDINATOR (coordinating conjunctions and arguments of coordination), CONTIN (part of an expression), TOP (root label), SEPARATOR (punctuation) and DEPENDENT (default label).

Although a reduced number of tags was used in conversion 2, it gave best results in some of the experiments. However, a more precise set of dependency tags should increase parsing accuracy<sup>3</sup>. Besides being somehow incomplete, the labels from conversion 2 were taken from an annotation scheme which is more semantically oriented and which was originally developed for Italian. Several changes were made in order the labels to represent syntactic more than semantic relations and fit the language (Bulgarian) better.

An example of such a change is using subjects and objects only in their shallow sense and not in prepositional phrases, for example<sup>4</sup>.

The dependency set from conversion 3 is more fine-grained than the dependency labels set of conversions 1 and 2. The number of labels is 16: subj (subject), obj (object), mod (modifier), indobj (indirect object), comp (complement), prepcomp (complement of preposition), adjunct, xcomp (clausal complement), xmod (clausal modifier), clitic (clitic form), xadjunct (clausal adjunct), marked (clauses introduced by a subordinator), pragadjunct (pragmatic adjunct), xsubj (clausal subject), xprepcomp (clausal complement of preposition), conj (coordinated conjunction), conjarg (argument of a coordinated construction), ROOT (root label), punc (punctuation).

Whereas there are labels with the same role in the three dependency label sets there are labels from each set with no strict analogues from the other two. Basic categories as root nodes, subjects, objects as well as punctuation are treated in the same way in all the dependency sets. Coordinations are treated differently. Conversion 3 has two different labels for coordinated constructions, namely ‘conj’ and ‘conjarg’. For the other sets there is only one (‘CC’ in conversion 1 and ‘COORDINATOR’ in conversion 2).

The variety of the other labels concerns mainly the detailness and different priorities of the relation encodings. For example, in conversions 1 and 3 ‘complement of preposition’ is set ‘PR’ and ‘prepcomp’, while in conversion 2 there is no such distinction. Then, in conversions 1 and 2 ‘predicative complement’ is set ‘PRD’ and ‘PRED-COMPL’, while in conversion 3 this kind of complement is a part of a broader label – ‘comp’.

### 4.2. Problematic issues

The dependency table guides the choice of the appropriate dependency label for the arc that has already been found using the head table. Relying on two constituents above the word in the original treebank a dependency label should be chosen. This was the approach in conversion 2. In conversion 1, there were rules in which only one constituent above one of the words from the relation and two – above the other were enough to determine the dependency labels of some arcs in the graphs of some sentences.

Using one or two constituents above the words for determining the labels of each dependency relation might not be very appropriate for languages with free word order. In particular, if only two constituents are taken in mind when determining the label of the relation we may end up with errors like mistaken subjects and objects, especially if the structure of the trees in the treebank is too flat or it is too deep and both subject and object candidates have the same two constituents above them.

In languages like Bulgarian, where long-distance dependencies are common, it is difficult to keep the annotation scheme of the treebank uniform. A typical example for ‘breaking’ the boundaries of a constituent is in cases where a noun or pronoun subject is included in the verb phrase. Having such structures in our treebank it is harder to convert in a plausible way. The trees from the BulTreeBank are generally more deep than the trees from the Penn treebank.

<sup>3</sup>See Section 7.

<sup>4</sup>Besides being inconvenient to process, these constructions

cannot be converted using a common head table.

We consider one of the reasons for that to be the free-word-orderness of Bulgarian.

One of the problems with increasing the number of constituents above each word in the sentence tree to guide the decision which label to be used for each particular dependency arc is that the number of rules in the dependency table will increase. In some cases two constituents above the words from the tree are enough but in other cases more constituents should be considered. Using a dependency table that relies on two or three constituents above each word in the tree with the rules with three constituents having higher priority might be worth trying.

A common problem in conversion 2, for example, was mistaking the subject and the object in the sentence. In these cases two constituents above the words in the tree were not enough for selecting the appropriate label.

The following sentences will clarify this issue: ‘V dushata i se pojavi omraza sreshtu men’ (‘Hatred against me arose in her soul.’). Here ‘omraza’ (‘hatred’) is the subject of the verb ‘pojavi’ (‘arose’). The constituents above ‘omraza’ are N and NPA. The constituents above ‘pojavi’ are V and V. But in the sentence: ‘I shte napishat kritika za mene / pod formata na policejski akt’ (‘And they will write criticism about me / in the form of a policeman’s statement’) ‘kritika’ (‘criticism’) is the object of the verb ‘napishat’ (‘write’). The constituents above ‘kritika’ are N and NPA and the constituents above ‘napishat’ are V and V – the same as the constituents from the first example sentence.

## 5. Error analysis

Whereas in the previous section we described problems with incorrectly assigned dependency labels, if having a correctly retrieved dependency graph, this part of the report addresses cases of incorrectly attached dependents. We show where our conversions are wrong with respect to the gold standard. There might be at least two different reasons for incorrect attachment corresponding to two different types of errors: 1) relations which are errors according to any theory of dependency grammar and 2) relations where the errors are not errors according to some theory of dependency grammar, especially a theory that is consistent with the head tables of some of our conversions.

Inconsistencies between the conversions can be observed in the verb chain treatment and clauses. If an auxiliary verb is considered the head in a VP, sometimes arguments that normally should be attached to the main verb, like subject and object, will be attached by the conversion algorithm to the auxiliary verb instead. Having auxiliaries as heads can be syntactically more plausible, but we unwillingly neglect the valency of the verb, if not attaching its arguments immediately<sup>5</sup>.

According to (Tesnière, 1959) the main verb and the auxiliary form a nucleus and all the dependents of the verb should be attached to the nucleus. But we would not like to pass the boundary by introducing relations among words other than ‘dependent’. The auxiliary, if present, is chosen to be the head of the VP in conversions 1 and 3, contrary to

the solution implemented in conversion 2 where the main verb is chosen to be the head. We have agreed to use the auxiliary as the head in the gold standard.

The other major difference in the conversions are clauses. In conversions 1 and 3 subordinating conjunctions were chosen to be the heads of the clauses. However, the main verb of the clause was chosen to be the head of the construction in conversion 2. We have the subordinating conjunction being the head of the different types of clauses in the gold standard set. Similarly to the wrong attachment of arguments of the main verb in the auxiliary case, the conversion algorithm can attach the arguments of the main verb to the subordinating conjunction rather than the main verb and this is an error.

In table 5 the three conversions of the sentence ‘I shte napishat kritika za mene / pod formata na policejski akt.’ (‘And they will write criticism about me / in the form of a policeman’s statement.’) are given together with the gold standard heads. Each row from the table contains a numbered word from the sentence together with information about its part-of-speech, the number of the head word, extracted from conversions 1, 2, and 3 as well as the gold standard head and the dependency labels from each of the conversions.

Conversions 1 and 2 of the sentence from table 5 are identical. The root of conversion 1 should be the same as the root of conversion 3 and the gold standard. This error is probably due to the rules from the head table of conversion 2 that were added to the head table of conversion 1.

Another serious problem reflecting on the conversion procedures is the presence of errors in the treebank. If a head rule is introduced just for the reason to deal with an error annotation, the quality of the conversion will decrease. Sometimes it is hard to distinguish erroneous annotations from proper ones. It is especially tricky to convert linguistic constructions which were not specified in the annotation guidelines of the treebank.

## 6. The parser

We used the Malt parser (Nivre et al., 2004) in our experiments. Malt parser is a data-driven dependency parser that uses a dependency treebank to learn the actions of a shift-reduce parsing algorithm. It had been tested on several languages, including Swedish, Italian and Bulgarian among others. It has proven to be easily portable from one language to another and is suitable for parsing the data that we have converted.

Malt parser is more generally a framework for construction of different parsers. Different features as part-of-speech tags, lexical units and dependency labels can be used for preparing feature models for learning. Some feature models had proven to be language independent to a large extent. For example, the model m4 is always better than the model m2. Using the m7 model (Marinov and Nivre, 2005) report very good results for Bulgarian. The model consists of six part-of-speech features, four lexical features and four dependency features.

Several learning methods are available as well as a few parsing algorithms within the framework of Malt parser. In this report, however, we report results that were obtained

<sup>5</sup>These clarifications were discussed in personal communication with Joakim Nivre.

W No	Word	PoS	Head 1	Head 2	Head 3	Head gold	Dep 1	Dep 2	Dep 3
1	I	Cp	3	3	2	2	-	COORDINATOR	conj
2	shte	Tx	3	3	0	0	VC	ARG	ROOT
3	napishat	Vpptf-r3p	0	0	2	2	ROOT	TOP	comp
4	kritika	Ncfsi	3	3	2	2	OBJ	SUBJ	obj
5	za	R	4	4	4	4	ATT	RMOD	mod
6	mene	Ppelas1	5	5	5	5	PR	ARG	prepcomp
7	/	pt	3	3	2	2	IP	SEPARATOR	punct
8	pod	R	3	3	2	2	ADV	INDCOMPL	adjunct
9	formata	Ncfsd	8	8	8	8	PR	ARG	prepcomp
10	na	R	9	9	9	9	ATT	RMOD	mod
11	policejski	Amsi	12	12	12	12	ATT	RMOD	mod
12	akt	Ncmsi	10	10	10	10	PR	ARG	prepcomp
13	.	pt	3	3	2	2	IP	SEPARATOR	punct

Table 5: Three different analyses and the gold standard for a sentence from the BulTreeBank.

Table 6: Parsing results for the BulTreeBank converted to dependency

Accuracy	Conv. 1	Conv. 2	Conv. 3
Unlabelled:	76.04%	86.00%	85.27%
Labelled:	20.69%	79.24%	79.48%

using only the arc-eager algorithm from (Nivre et al., 2004) together with memory based learning (Daelemans and den Bosch, 2005).

## 7. Parsing results

We annotated a small set of gold standard data on which to evaluate our conversions, because we did not have a large scale manually annotated dependency treebank. The reason for our decision to use the conversions for training and parsing with a statistical dependency parser was that this could give us evidence for dependency parsing performance of the BulTreeBank.

If the parsing results for the three conversions are different from each other, we can trace the inconsistencies and conclude good and bad conversion practices. However, parsing is not the best measure for the accuracy of our conversions, because a wrongly converted construction could be learned and parsed properly as well as a properly converted construction could be learned and parsed wrongly and the other two combinations are also possible. Although the parsing task is not entirely appropriate for evaluation of the accuracy of our conversions we can conjecture about their applicability.

All the results are obtained on the same training and test sets with the original gold standard part-of-speech tags of the BulTreeBank. The results are given in Table 6. The metric that we use for evaluation is labelled and unlabelled accuracy measured per word as defined by (Lin, 1998).

If using memory based learning, conversion 2 gives the best unlabelled accuracy and labelled accuracy which is very close to the best. A defect of conversion 2 is the significant number of arcs in the training and test sets (around 4%) that

were given the default dependency label ‘DEPENDENT’. We believe that if we reduce that percent, parsing results will improve. The same statement is valid for conversion 1 where there are too many default labels. This is due to the rules from the head table of conversion 2 that we artificially added to the head table of conversion 1.

Finally we should mention some preliminary experiments with the Malt parser using another learner within the parsing framework – Support Vector Machines (Chang and Lin, 2005). We obtained better results for conversions 2 and 3. We haven’t performed tests on conversion 1.

## 8. Conclusions and future work

We can conclude that in general terms the head table is more important than the conversion algorithm. It should be easy to use different conversion algorithms with the same head table and obtain the same results in most of the cases (with minor corrections in the algorithms which have something to do with treebank specific phenomena).

We conclude that the choice of dependency labels for automatically converted constituency compatible treebanks should be linguistically motivated and specific for the language.

We showed that obtaining applicable dependency parsing results for Bulgarian is achievable if we use the BulTreeBank, even though it is not a dependency treebank. Our results can be used in areas like Question Answering and other tasks from NLP where the syntactic structure of the sentence can provide clues for better analysis and disambiguation.

Our work demonstrates that a treebank that combines the constituent and the dependency information is a valuable source for extraction of dependency treebanks with different inventory of dependency labels and with different granularity of specificity. The experiments with the Malt parser show that the quality of the parsing output depends on the information in treebank. This gives us area for future research how to extract the most appropriate treebank for a given task.

Having achieved state-of-the-art parsing for Bulgarian we can further improve our conversions in several directions. The first one is for the dependency tables of conversion 1

and 2. The table of conversion 2 is not large enough and there are still around 4% of the words in both the training and test data having the default ‘DEPENDENT’ label. The same problem can be observed to a greater extent in conversion 1.

The second direction for further research is to choose a unified representation for all the dependency structures, combining approaches from the three conversions. A first step in that direction could be to perform several further transformations in order to fix errors, e.g. in the clauses and VPs’ treatment. Combined with a unified approach to dependency representation this step could gain some parsing accuracy.

Optimizing the different options and feature models of the Malt parser for Bulgarian and using the SVM learner can improve further parsing results.

Besides the dependency parser we could try a good constituency parser on the BulTreeBank as well. Our intuition is that such a parser would not give better parsing accuracy, because of the free word order nature of the Bulgarian language. Nevertheless, evaluating and comparing a dependency and a constituency parser on the BulTreeBank can point some interesting directions for future research.

### Acknowledgements

We would like to thank Joakim Nivre for providing the Malt parser as well as hints about best performing feature models and issues concerning treebank conversion. We thank Alberto Lavelli for his valuable comments and suggestions on a previous version of this report. Also, we thank Svetoslav Marinov who made available to us his previous work on treebank conversion which is used in the report.

The author did this research in cooperation with Kiril Simov and Petya Osenova at the Institute for Parallel Processing, Bulgarian Academy of Sciences.

The work was partly funded by a grant provided within the project BIS-21++ at the Institute for Parallel Processing, Bulgarian Academy of Sciences. BIS-21++ is a project funded by the European Commission in FP6 INCO via contract no.: INCO-CT-2005-016639.

### 9. References

C. Bosco. 2004. *A grammatical annotation system for treebank annotation*. Ph.D. thesis, University of Torino.

A. Chanev. 2005. Portability of dependency parsing algorithms – an application for Italian. In *Proc. of the fourth workshop on Treebanks and Linguistic Theories (TLT)*, Barcelona.

C.-C. Chang and C.-J. Lin. 2005. LIBSVM: A library for Support Vector Machines. URL: <http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf>.

E. Charniak. 2000. A maximum-entropy-inspired parser. In *Proc. of the First Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Seattle.

M. Collins, J. Hajič, E. Brill, L. Ramshaw, and C. Tillmann. 1999. A statistical parser for Czech. In *Proc. of the 37th Meeting of the Association for Computational Linguistics (ACL)*, College Park.

M. Collins. 1997. Three generative, lexicalized models for statistical parsing. In *Proc. of the 35th Annual Meeting of the Association for Computational Linguistics (ACL)*, Madrid.

A. Corazza, A. Lavelli, G. Satta, and R. Zanoli. 2004. Analyzing an Italian treebank with state-of-the-art statistical parsers. In *Proc. of the 3rd workshop on Treebanks and Linguistic Theories (TLT 2004)*, Tübingen.

W. Daelemans and A. Van den Bosch. 2005. *Memory-Based Language Processing*. Cambridge University Press.

M. Daum, K. A. Fith, and W. Menzel. 2004. Automatic transformation of phrase treebanks to dependency trees. In *Proc. 4th Int. Conf. on Language Resources and Evaluation, LREC-2004*, Lisbon.

H. Gaifman. 1965. Dependency systems and phrase-structure systems. *Information and control*, 8:304–337.

J. Hajič. 1998. Building a syntactically annotated corpus: The Prague Dependency Treebank. In *Issues of Valency and Meaning*, Prague. Karolinum.

E. Hinrichs, J. Bartels, Y. Kawata, V. Kordoni, and H. Telljohann. 2000. The Verbmobil treebanks. In *Proc. of 5. Konferenz zur Verarbeitung natürlicher Sprache*, Ilmenau.

H. Krushkov and A. Chanev. 2005. Automatic parsing: a probabilistic approach for Bulgarian. In *Proc. of Annual Spring Conference of the Union of Bulgarian Mathematicians (UBM)*, Borovetz.

S. Kübler and H. Telljohann. 2002. Towards a dependency-based evaluation for partial parsing. In *Proc. of the LREC-Workshop Beyond PARSEVAL – Towards Improved Evaluation Measures for Parsing Systems*, Las Palmas.

D. Lin. 1998. A dependency-based method for evaluating broad-coverage parsers. *Natural Language Engineering*, 4 (2):97–114.

D. M. Magerman. 1994. *Natural language parsing as statistical pattern recognition*. Ph.D. thesis, Stanford University.

M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19 (2):273–290.

S. Marinov and J. Nivre. 2005. A data-driven parser for Bulgarian. In *Proc. of the fourth workshop on Treebanks and Linguistic Theories (TLT)*, Barcelona.

J. Nilsson, J. Hall, and J. Nivre. 2005. MAMBA meets TIGER: Reconstructing a Swedish treebank from antiquity. In *Proc. from the special session on treebanks at NODALIDA 2005*, Joensuu.

J. Nivre and M. Scholz. 2004. Deterministic dependency parsing of English text. In *Proc. of 20th International Conference on Computational Linguistics (COLING 2004)*, Geneva.

J. Nivre, J. Hall, and J. Nilsson. 2004. Memory-based dependency parsing. In *Proc. of the Eighth Conference on Computational Natural Language Learning (CoNLL)*, Boston.

- J. Nivre. 2005. *Inductive Dependency Parsing of Natural Language Text*. Ph.D. thesis, University of Växjö.
- K. Simov and P. Osenova. 2003. A treatment of coordination in the bulgarian hpsg-based treebank. In *Proc. from FDSL-5*, Leipzig. in press.
- K. Simov, P. Osenova, S. Kolkovska, E. Balabanova, D. Doikoff, K. Ivanova, A. Simov, and M. Kouylekov. 2002. Building a linguistically interpreted corpus of Bulgarian: the BulTreeBank. In *Proc. of LREC 2002*, Canary Islands.
- W. Skut, T. Brants, B. Krenn, and H. Uszkoreit. 1997. Annotating unrestricted German text. In *Proc. of 6. Fachtagung der Section Computerlinguistic der Deutschen Gesellschaft für Sprachwissenschaft*, Heidelberg.
- H. Tanev. 2001. *Automatic Text Analysis and Ambiguities Resolution in Bulgarian*. Ph.D. thesis, University of Plovdiv.
- L. Tesnière. 1959. *Éléments de syntaxe structurale*. Editions Klincksieck.
- T. Ule and S. Kübler. 2004. From phrase-structure to dependencies, and back. In *Proc. of the International Conference on Linguistic Evidence*, Tübingen.
- Z. Žabokrtský and I. Kučerová. 2002. Transforming Penn Treebank phrase trees into (Praguian) tectogrammatical dependency trees. *Prague Bulletin of Mathematical Linguistics*, 78:77–94.
- Z. Žabokrtský and O. Smrž. 2003. Arabic syntactic trees: from constituency to dependency. In *Proc. 10th Conference of the European Chapter of the Association of Computational Linguistics, EACL 2003*, Budapest.
- F. Xia. 2001. *Automatic Grammar Generation from Two Different Perspectives*. Ph.D. thesis, University of Pennsylvania.
- H. Yamada and Y. Matsumoto. 2003. Statistical dependency analysis with Support Vector Machines. In *Proc. of IWPT*, Nancy.