An Empirical Evaluation of Automatic Conversion from Constituency to Dependency in Hungarian

Katalin Ilona Simkó[1], **Veronika Vincze**[1,2], **Zsolt Szántó**[1], **Richárd Farkas**[1]
[1]University of Szeged
Department of Informatics
[2]MTA-SZTE Research Group on Artificial Intelligence
kata.simko@gmail.com
{vinczev,szantozs,rfarkas}@inf.u-szeged.hu

Abstract

In this paper, we investigate the differences between Hungarian sentence parses based on automatically converted and manually annotated dependency trees. We also train constituency parsers on the manually annotated constituency treebank and then convert their output to dependency trees. We argue for the importance of training on gold standard corpora, and we also demonstrate that although the results obtained by training on the constituency treebank and converting the output to dependency format and those obtained by training on the automatically converted dependency treebank are similar in terms of accuracy scores, the typical errors made by different systems differ from each other.

1 Introduction

Nowadays, two popular approaches to data-driven syntactic parsing are based on constituency grammar on the one hand and dependency grammar on the other hand. There exist constituency-based treebanks for many languages and dependency treebanks for most of these languages are converted automatically from constituent trees with the help of conversion rules, which is the case for e.g. the languages used in the SPMRL-2013 Shared Task (Seddah et al., 2013) with the exception of Basque, where constituency trees are converted from manually annotated dependency trees (Aduriz et al., 2003), and Hungarian, where both treebanks are manually annotated (Csendes et al., 2005; Vincze et al., 2010). However, the quality of automatic dependency conversion is hardly investigated.

Hungarian is one of those rare examples where there exist manual annotations for both constituency and dependency syntax on the same bunch of texts, the Szeged (Dependency) Treebank (Csendes et al., 2005; Vincze et al., 2010), which makes it possible to evaluate the quality of a rule-based automatic conversion from constituency to dependency trees, to compare the two sets of manual annotations and also the output of constituency and dependency parsers trained on converted and gold standard dependency trees.

We investigate the effect of automatic conversions related to the two parsing paradigms as well. It is well known that for English, the automatic conversion of a constituency parser's output to dependency format can achieve competitive unlabeled attachment scores (ULA) to a dependency parser's output trained on automatically converted trees[1] (cf. Petrov et al. (2010)). One of the possible explanations for this is that English is a configurational language, hence constituency parsers have advantages over dependency parsers here. We check whether this hypothesis holds for Hungarian too, which is the prototype of free word order languages.

In this paper, we compare three pairs of dependency analyses in order to evaluate the usefulness of converted trees. First, we examine the errors of the conversion itself by comparing the converted dependency trees with the manually annotated gold standard ones. Second, we argue for the importance of training parsers on gold standard trees by looking at the typical differences between the outputs of

[1] However, it has been pointed out that errors in the conversion script may significantly influence the results of parsing, see e.g. Petrov and McDonald (2012) and Pitler (2012)

Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 1392–1401, Dublin, Ireland, August 23-29 2014.

dependency parsers trained on converted (silver standard) trees, parsers trained on gold standard trees and the manual annotation itself. Third, we demonstrate that similar to English, training on a constituency treebank and converting the results to dependency format can achieve similar results in terms of ULA to the dependency parser trained on the automatically converted treebank, but the typical errors they make differ in both cases.

2 Parsing Hungarian on the Szeged Treebank

Hungarian is a morphologically rich language, where word order encodes information structure, which makes its syntactic analysis very different from English's as the arguments in a sentence cannot be determined by their position but by their suffixes, cf. É. Kiss (2002). Words' grammatical functions are signified by case suffixes and verbs are marked for the number and person of their subject and the definiteness of their object, thus these arguments may be often omitted from the sentence: *Látlak* (see-1SG2OBJ) "I see you". Due to word order reasons, words that form one syntactic phrase may not be adjacent (long-distance dependencies), which is true for the possessive construction as well: the possessor and the possessed may be situated in two distant positions: *A fiúnak elvette a kalapját* (the boy-DAT take-PAST-3SGOBJ the hat-POSS3SG-ACC) "He took the boy's hat". Verbless clauses are also common in Hungarian, as the copula in third person singular present tense indicative form is phonologically empty, while it is present in all other moods and tenses: *A kalap piros* (the hat red) "The hat is red", but *A kalap piros volt* (the hat red was) "The hat was red".

The Szeged Treebank (Csendes et al., 2005) is a manually annotated constituency treebank for Hungarian consisting of 82,000 sentences. Besides the phrase structure, grammatical roles of the verbs' arguments and morphological information are also annotated. It incorporates texts from six different domains: short business news, newspaper, law, literature, compositions and informatics, however, in this paper, we just focus on the short business news domain.

The Szeged Dependency Treebank (Vincze et al., 2010) contains manual dependency syntax annotations for the same texts. Certain linguistic phenomena – such as discontinuous structures – are annotated in this treebank, but not in the constituency treebank. In the dependency treebank, the possessor is linked to the possession while this connection is not annotated in the constituency treebank. The two types of trees can be seen in Figure 1.

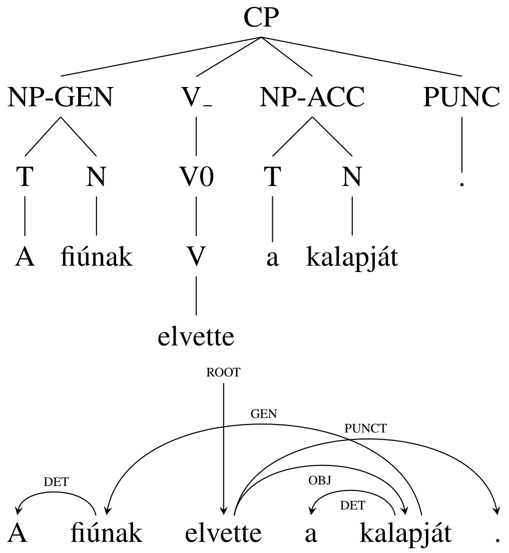

Figure 1: Discontinuous structure *A fiúnak elvette a kalapját* (the boy-DAT take-past3SGOBJ the hat-POSS3SG-ACC) "He took the boy's hat" in constituency and dependency analysis.

Another difference between the two treebanks is the way they represent different types of complex sentences, as can be seen in Figure 2. In the dependency treebank subordinations and coordinations are handled very similarly. The head of one of the clauses (the subordinated clause or the second clause in the case of coordination) is linked to the head of the other clause (the matrix clause of the subordination or the first clause of the coordination), only the type of relation between the two heads differs in the two structures, in the dependency tree in Figure 2, the heads of the three clauses (*átjött* “came over”, *megígérte* “promised” and *eljön* “come”) are linked to one another through their conjunctions with either an ATT relation in the case of subordination or COORD for coordination. In the constituency treebank these sentences are represented very differently: in the case of subordination, the subordinated clause is within the matrix clause: CP_3 is within CP_2 in the constituency tree in Figure 2. Coordinated clauses appear at the same level in the structure, in the same figure CP_1 and CP_2 are coordinated clauses.

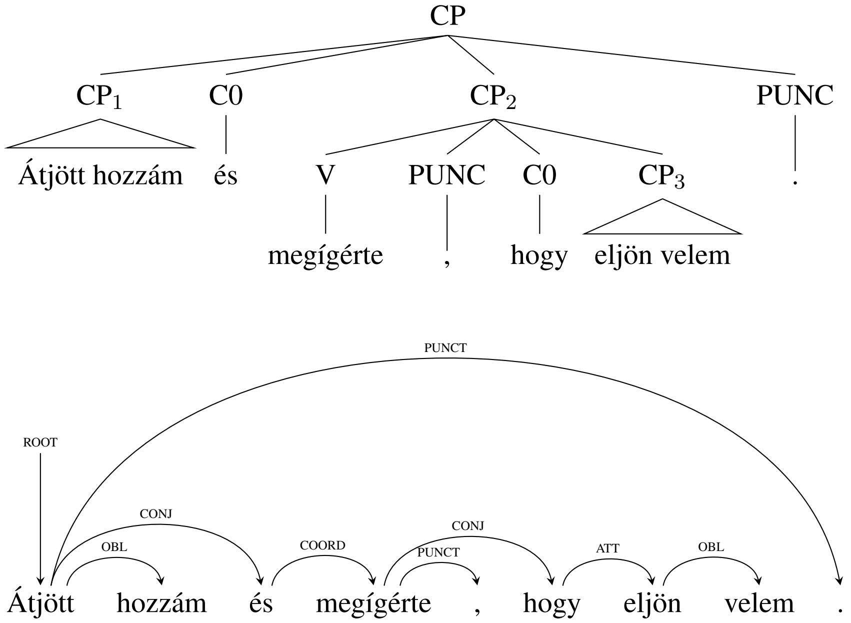

Figure 2: Constituency and dependency analysis of coordination and subordination in the sentence *Átjött hozzám és megígérte, hogy eljön velem* (through.come-PAST-3SG to.me and promise-PAST-3SG-OBJ that away.come-3SG with.me) “He came over and promised that he will come with me”.

The parallels of these two manually annotated treebanks make them suitable for testing our hypotheses about automatic dependency conversion. The differences between them originate from the characteristics of constituent and dependency syntax.

3 Converting Constituency Trees to Dependency Trees

In this section, we present our methods to convert constituency trees to dependency trees and we also discuss the most typical sources of errors during conversion.

3.1 Conversion rules

In order to convert constituency trees to dependency trees, we used a rule based system. Sentences with virtual dependency nodes were omitted, as they are not annotated in the constituent treebank and their treatment in dependency trees is also problematic (Farkas et al., 2012; Seeker et al., 2012). As a result, we worked with 7,372 sentences and 162,960 tokens.

First, we determined the head of each clause (CP) and the relations between CPs in complex sentences. In most cases the head of the CP is a finite verb, if the CP contains no finite verb, the head is the either an infinitive verb or a participle, if none of these are present in the CP, the head can be a nominal expression. The relations between the CP heads make up the base of the dependency structure using ROOT relation for the sentence’s main verb, COORD for coordination and ATT for subordination, as well as CONJ in the case of conjunctions between the CPs.

The arguments of verbs, infinitives and participles in the CP were linked to their governor and marked for their grammatical role in the Szeged Treebank. We used this information to construct the appropriate dependency relations between governors and their arguments. The main grammatical roles such as subject, object, dative have their own label in dependency syntax, while minor ones are assigned the oblique (OBL) relation. The argument's modifiers were then linked to the head or other modifiers based on the phrase structure with relations according to their morphological code.

Long distance dependencies, like the connection between a genitive case possessor and the possessed are not annotated in the constituency treebank. In these cases we used morphological information to link these elements together in the dependency tree. Figure 3 shows an example of converting a constituency tree to a dependency tree.

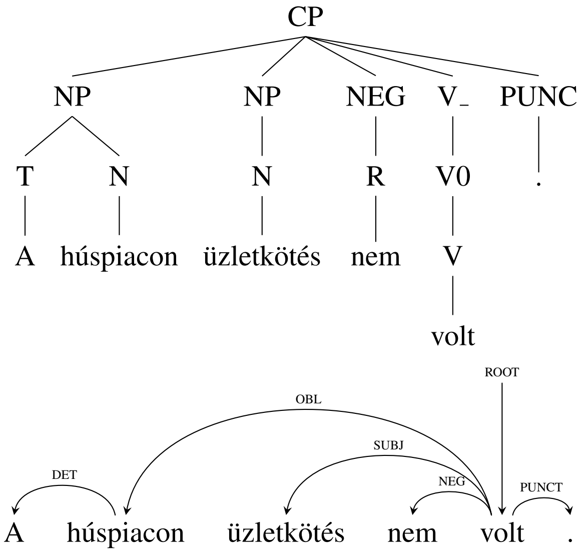

Figure 3: Conversion of the sentence *A húspiacon üzletkötés nem volt* (the meat.market-SUP transaction not was) "There were no transactions at the meat market." from constituency to dependency trees.

3.2 Error Analysis

We automatically converted the constituency treebank into dependency trees following the principles described above and detailed at our website (`http://www.inf.u-szeged.hu/rgai/SzegedTreebank`). For evaluation, we applied the metrics labeled attachment score (LAS) and unlabeled attachment score (ULA), without punctuation marks. The accuracy of the conversion was 96.51 (ULA) and 93.85 (LAS). The errors made during conversion were categorized manually in 200 sentences selected randomly from the short business news subcorpus of the Szeged Dependency Treebank, and the most typical ones are listed in Table 1, Column *convError*.

As it is shown, the most common source of error was when more than one modifier was within a phrase as the example in Figure 4 shows. In each figure, the gold standard parse can be seen on the left hand side while the erroneous one can be seen on the right hand side.

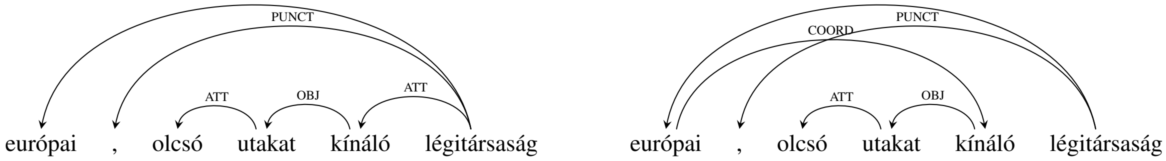

Figure 4: Multiple modifier error in *európai, olcsó utakat kínáló légitársaság* (European cheap trips-ACC offering airline) "European airline offering cheap trips".

Error type	convError		goldTrain		silverTrain		BerkeleyConv		convDep	
	#	%	#	%	#	%	#	%	#	%
Coordination	26	13.00	39	13.22	59	14.82	55	16.37	64	19.57
Multiple modifiers	26	13.00	30	10.17	49	12.31	52	15.48	47	14.37
Determiner	7	3.50	28	9.49	25	6.28	31	9.23	31	9.48
Conj./adverb attached	33	16.50	23	7.80	45	11.31	39	11.61	42	12.84
Arg. of verbal element	10	5.00	27	9.15	34	8.54	59	17.56	44	13.46
Sub- vs. coordination	7	3.50	9	3.05	12	3.02	–	–	–	–
Possessor	9	4.50	14	4.75	16	4.02	28	8.33	22	6.73
Wrong root	14	7.00	17	5.76	23	5.78	35	10.42	27	8.26
Consecutive nouns	4	2.00	11	3.73	14	3.52	13	3.87	15	4.59
Multiword NE	8	4.00	25	8.47	33	8.29	8	2.38	19	5.81
Wrong MOD label	25	12.50	26	8.81	34	8.54	–	–	–	–
Wrong other label	17	8.50	33	11.19	30	7.54	–	–	–	–
Other errors	14	7.00	13	4.41	24	6.03	16	4.76	16	4.89
Total	200	100	295	100	398	100	336	100	327	100

Table 1: Error Types. convError: errors made during converting constituency trees to dependency trees. goldTrain: errors in the output got by training the Bohnet parser on the gold standard data. silverTrain: errors in the output got by training the Bohnet parser on the silver standard data. BerkeleyConv: errors in the output got by training the Berkeley parser on the gold standard constituency data and converting the output into dependency format. convDep: errors in the output got by training the Bohnet parser without dependency labels on the silver standard data.

Coordination errors occurred when multiple members of a coordination were wrongly connected. On the other hand, the attachment of conjunctions and some adverbs was also problematic, for example in Figure 5 the conjunction *is* "also" is connected to the verb in the gold standard and to the noun in the converted version.

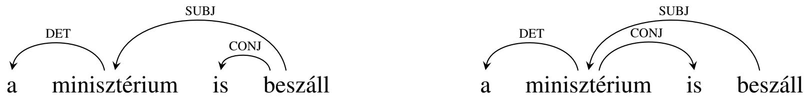

Figure 5: Conjunction attachment error in *a minisztérium is beszáll* (the ministry also steps.in) "the ministry also steps in".

Also, the constituency treebank did not mark all the grammatical relations (e.g. numerals and determiners were simply parts of an NP but had no distinct labeling, like *[NP az öt [ADJP fekete] kutya]* (the five black dog) "the five black dogs"), but it was necessary to assign them a dependency label and a parent node during conversion. However, in some cases it was not straightforward which modifier modifies which parent node: for instance, in *[NP nem [ADJP megfelelő] módszerek]* (not appropriate methods) "inappropriate methods", the negation word *nem* is erroneously attached to the noun instead of the adjective in the converted phrase. Determiner errors were those where the determiner was attached to the wrong noun in a NP with a noun modifier. In CPs with multiple verbal elements (both a finite verb and an infinitive or a participle in the CP) the arguments were sometimes linked to the wrong verb, as in Figure 6.

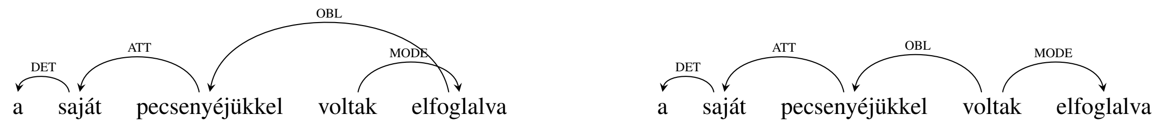
OBL
ATT
MODE
DET
a saját pecsenyéjükkel voltak elfoglalva
ATT
OBL
MODE
DET
a saját pecsenyéjükkel voltak elfoglalva

Figure 6: Verbal argument error in *a saját pecsenyéjükkel voltak elfoglalva* (the own roast-3PLPOSS-INS were busy) “they were busy with their own thing”.

Possessors are sometimes wrongly identified during conversion as long distance dependencies are not marked in the constituency treebank (see Figure 7).

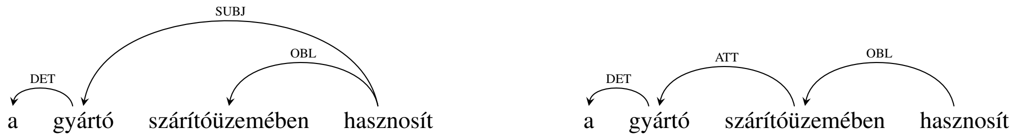
SUBJ
OBL
DET
a gyártó szárítóüzemében hasznosít
ATT
OBL
DET
a gyártó szárítóüzemében hasznosít

Figure 7: Possessor attachment error in *a gyártó szárítóüzemében hasznosít* (the manufacturer drying.plant-3SGPOSS-INE utilizes) “the manufacturer utilizes it in its drying plant”.

In CPs with more verbal element, sometimes the wrong word is selected as the root, as in Figure 8.

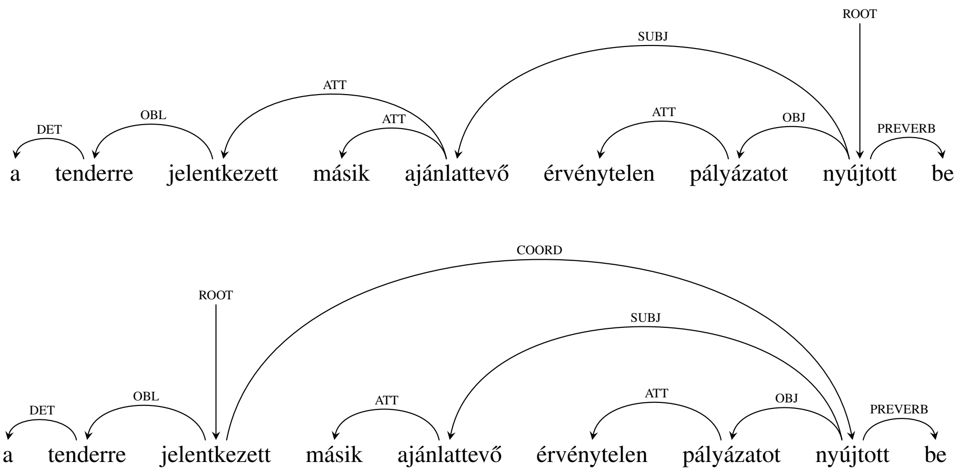
ROOT
SUBJ
ATT
DET
OBL
ATT
ATT
OBJ
PREVERB
a tenderre jelentkezett másik ajánlattevő érvénytelen pályázatot nyújtott be
COORD
ROOT
SUBJ
DET
OBL
ATT
ATT
OBJ
PREVERB
a tenderre jelentkezett másik ajánlattevő érvénytelen pályázatot nyújtott be

Figure 8: Root error in *a tenderre jelentkezett másik ajánlattevő érvénytelen pályázatot nyújtott be* (the tender-SUB applied other bidder invalid application-ACC submit-PAST-3SG) “the other bidder applying to the tender submitted an invalid application”.

In some cases, consecutive (but separate) noun phrases were taken as one unit as if one noun modified the other, for example in Figure 9.

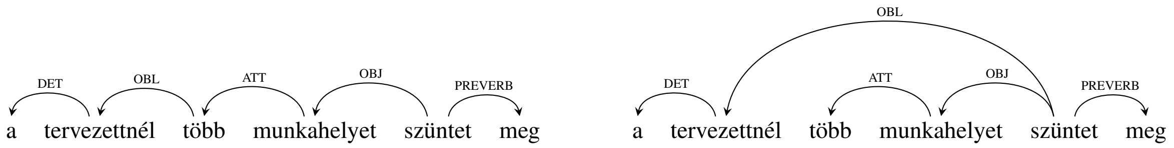
DET
OBL
ATT
OBJ
PREVERB
a tervezettnél több munkahelyet szüntet meg
OBL
DET
ATT
OBJ
PREVERB
a tervezettnél több munkahelyet szüntet meg

Figure 9: Consecutive noun error in *a tervezettnél több munkahelyet szüntet meg* (the planned-ADE more workplace-ACC terminates) “it terminates more workplaces than planned”.

Multiword NEs also caused some problems in the conversion, as in Figure 10.

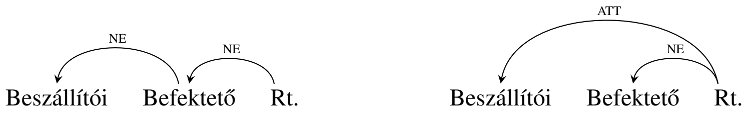

Figure 10: Multiword NE error in *Beszállítói Befektető Rt.* (a name of a company) .

In other cases, divergences between the gold standard and the converted trees are due to some erroneous annotations either in the constituency treebank or in the dependency treebank. A typical example of this is the wrong MOD (modifier) label. In the treebank, locative and temporal modifiers were classified according to the tridirectionality typical of Hungarian adverbs and case suffixes: *where*, *from where* and *to where* (or *when*, *from what time and till what time*) the action is taken place. Thus, there are six dependency relations dedicated to these aspects and all the other adverbials are grouped under the relation MOD. However, this distinction is rather semantic in nature and was sometimes erroneously annotated in the constituency treebank, which was later corrected in the dependency one and thus now resulted in conversion errors, as shown in Figure 11.

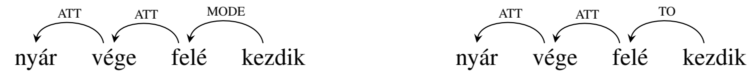

Figure 11: MOD label error in *nyár vége felé kezdik* (summer end-3SGPOSS around begin) "they begin around the end of the summer".

There were also some atypical errors that occurred too rarely to categorize them in a different class, like cases when an article or determiner got erroneously attached to a verb and so on, so they were lumped into the category of "other errors" in Table 1.

4 Training on Gold Standard and Silver Standard Trees

We also experimented with training the Bohnet dependency parser (Bohnet, 2010) on the manually annotated (gold standard) and the converted (silver standard) treebank. The Bohnet parser (Bohnet, 2010) is a state-of-the-art[2] graph-based parser, which employs online training with a perceptron. The parser contains a feature function for the first order factor, one for the sibling factor, and one for the grandchildren.

From the corpus, 5,892 sentences (130,211 tokens) were used in the training dataset and the remaining 1,480 sentences (32,749 tokens) in the test dataset. For evaluation, we again applied the metrics LAS and ULA. Results are shown in Table 2, Rows *goldTrain* and *silverTrain*.

As the numbers show, better results can be achieved when the gold standard data are used as training database than when the parser is trained on the silver standard data, the differences being 1.6% (ULA) and 3.16% (LAS). Besides evaluation scores, we also compared the outputs of the two scenarios: we used the same set of randomly selected sentences as when investigating conversion errors and carried out a manual error analysis against the gold standard data in each case: see Table 1, Columns *goldTrain* and *silverTrain*.

There are some common error types that seem to cause problems for both ways of parsing. For instance, coordination and multiple modifiers are among the most frequent sources of errors in both cases as for the error rates are concerned. However, with regard to the absolute numbers, we can see that both error types are reduced when the gold standard dataset is used for training. On the other hand, finding the parent node of a conjunction or an adverb seems to improve significantly when the parser is trained on gold standard data. This is probably due to the fact that they are not marked in the constituency treebank and thus training data for these grammatical phenomena are very noisy in the silver standard treebank. All in all, we argue that there are some grammatical phenomena – e.g. the attachment of

[2] For a comparative evaluation with other dependency parsers on the same treebank see Farkas et al. (2012). According to their results, the Bohnet parser achieved the best scores on the treebank hence we also used this parser in our experiments.

Setting	**LAS**	**ULA**
Conversion	93.85	96.51
goldTrain	93.48	95.17
silverTrain	90.32	93.57
BerkeleyConv	–	92.78
convDep	–	93.23

Table 2: Results of the experiments. Conversion: converting constituency trees to dependency trees. goldTrain: training the Bohnet parser on the gold standard data. silverTrain: training the Bohnet parser on the silver standard data. BerkeleyConv: training the Berkeley parser on the gold standard constituency data and converting the output into dependency format. convDep: training the Bohnet parser without dependency labels on the silver standard data.

conjunctions or adverbs – that require manual checking even if automatic conversion from constituency to dependency is applied.

5 Pre- or Post Conversion?

It is well known that for English, converting a constituency parser's output to dependency format (post conversion) can achieve competitive ULA scores to a dependency parser's output trained on automatically converted trees (pre conversion) (Petrov et al., 2010; Farkas and Bohnet, 2012). One of the possible reasons for this may be that English is a configurational language, hence constituency parsers are expected to perform better here. In this paper, we investigate whether this is true for Hungarian, which is the prototype of morphologically rich languages with free word order.

We employed the product-of-grammars procedure (Petrov, 2010) of the Berkeleyparser (Petrov et al., 2006), where grammars are trained on the same dataset but with different initialization setups, which leads to different grammars. We trained 8 grammars and used tree-level inference. The output of the parser was then automatically converted to dependency format, based on the rules described in Section 3 (*BerkeleyConv*). Second, we used the silver standard dependency treebank for training the Bohnet parser (*convDep*). Since our constituency parser did not produce grammatical functions for the nodes, we trained the Bohnet parser on unlabeled dependency trees in order to ensure a fair comparison here (that is the difference between the columns *BerkeleyConv* and *convDep* in Table 1).

As the numbers show, competitive results can be obtained with both methods, yielding an ULA score of 92.78 and 93.23, respectively. This means that the same holds for Hungarian as for English and the surprisingly good results of post conversion are not related to the configurational level of the language.

Manually analysing the errors on the same set of sentences as before, there are again some error categories that occur frequently in both cases such as coordination, the attachment of conjunctions, modifiers and determiners. On the other hand, training on constituency trees seems to have some specific sources of errors. First, the possessor in possessive constructions is less frequently attached to its possessed, which may be due to the fact that the genitive possessor is not linked to the possessed in the constituency treebank and thus the parser is not able to learn this relationship. Second, arguments of verbal elements (i.e. verbs, participles and infinitives) are also somewhat more difficult to find when there are at least two verbal elements within the clause, which is especially true for adverbial participles and infinitives. In Figure 6, the differences between the two trees are shown. The noun *pecsenyéjükkel* (roast-3PLPOSS-INS) "with their thing" is linked to the adverbial participle in the correct analysis, but it connects to the main verb in the other. Third, identifying the root node of the sentence may also be problematic for this setting. As Farkas and Bohnet (2012) reported that preconversion can achieve better results for finding the root node in English, this seems to be a language-specific issue and it represents an interesting difference between English and Hungarian. Nevertheless, training on constituency trees has a beneficial effect on finding multiword named entities. Hence, it can be concluded that although the evaluation scores are similar, the errors the two systems make differ from each other.

6 Discussion and Conclusions

Here, we compared dependency analyses of Hungarian obtained in different ways. It was revealed that although the accuracy scores are similar to each other, each system makes different types of errors. On the other hand, there are some specific linguistic phenomena that seem to be difficult for dependency parsing generally as they were among the most frequent sources of errors in each case (e.g. coordination, multiple modifiers and the attachment of conjunctions and adverbs).

Converting constituency trees into dependency trees enabled us to experiment with a silver standard dependency corpus as well. Our results empirically showed that better results can be achieved on the gold standard corpus, hence manual annotation of dependency trees is desirable. However, when there is no access to manually annotated dependency data, converting the output of a constituency parser into dependency format or training the dependency parser on converted data may also be viable: similar to English, both solutions result in competitive scores but the errors the systems make differ from each other.

In the future, we would like to investigate how the advantages of constituency and dependency representations may be further exploited in parsing Hungarian and we also plan to carry out some uptraining experiments with both types of parsers.

Acknowledgements

This work was supported in part by the European Union and the European Social Fund through the project FuturICT.hu (grant no.: TÁMOP-4.2.2.C-11/1/KONV-2012-0013).

References

Itziar Aduriz, Maria Jesus Aranzabe, Jose Maria Arriola, Aitziber Atutxa, A. Diaz de Ilarraza, Aitzpea Garmendia, and Maite Oronoz. 2003. Construction of a Basque dependency treebank. In *Proceedings of the 2nd Workshop on Treebanks and Linguistic Theories (TLT)*, pages 201–204, Växjö, Sweden.

Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 89–97.

Dóra Csendes, János Csirik, Tibor Gyimóthy, and András Kocsor. 2005. The Szeged TreeBank. In Václav Matousek, Pavel Mautner, and Tomás Pavelka, editors, *Proceedings of the 8th International Conference on Text, Speech and Dialogue, TSD 2005*, Lecture Notes in Computer Science, pages 123–132, Berlin / Heidelberg, September. Springer.

Katalin É. Kiss. 2002. *The Syntax of Hungarian*. Cambridge University Press, Cambridge.

Richárd Farkas and Bernd Bohnet. 2012. Stacking of dependency and phrase structure parsers. In *Proceedings of COLING 2012*, pages 849–866, Mumbai, India, December. The COLING 2012 Organizing Committee.

Richárd Farkas, Veronika Vincze, and Helmut Schmid. 2012. Dependency Parsing of Hungarian: Baseline Results and Challenges. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 55–65, Avignon, France, April. Association for Computational Linguistics.

Slav Petrov and Ryan McDonald. 2012. Overview of the 2012 shared task on parsing the web. Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL).

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 433–440.

Slav Petrov, Pi-Chuan Chang, Michael Ringgaard, and Hiyan Alshawi. 2010. Uptraining for accurate deterministic question parsing. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 705–713, Cambridge, MA, October. Association for Computational Linguistics.

Slav Petrov. 2010. Products of random latent variable grammars. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27, Los Angeles, California, June. Association for Computational Linguistics.

Emily Pitler. 2012. Conjunction representation and ease of domain adaptation. Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL).

Djamé Seddah, Reut Tsarfaty, Sandra Kübler, Marie Candito, Jinho D. Choi, Richárd Farkas, Jennifer Foster, Iakes Goenaga, Koldo Gojenola Galletebeitia, Yoav Goldberg, Spence Green, Nizar Habash, Marco Kuhlmann, Wolfgang Maier, Yuval Marton, Joakim Nivre, Adam Przepiórkowski, Ryan Roth, Wolfgang Seeker, Yannick Versley, Veronika Vincze, Marcin Woliński, and Alina Wróblewska. 2013. Overview of the SPMRL 2013 shared task: A cross-framework evaluation of parsing morphologically rich languages. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 146–182, Seattle, Washington, USA, October. Association for Computational Linguistics.

Wolfgang Seeker, Richárd Farkas, Bernd Bohnet, Helmut Schmid, and Jonas Kuhn. 2012. Data-driven dependency parsing with empty heads. In *Proceedings of COLING 2012: Posters*, pages 1081–1090, Mumbai, India, December. The COLING 2012 Organizing Committee.

Veronika Vincze, Dóra Szauter, Attila Almási, György Móra, Zoltán Alexin, and János Csirik. 2010. Hungarian Dependency Treebank. In *Proceedings of LREC 2010*, Valletta, Malta, May. ELRA.