Constraint Grammar-based conversion of Dependency Treebanks

Eckhard Bick
University of Southern Denmark
Campusvej 55, DK-5230 Odense M, Denmark
eckhard.bick@gmail.com

Abstract

This paper presents a new method for the conversion of one style of dependency treebanks into another, using contextual, Constraint Grammar-based transformation rules for both structural changes (attachment) and changes in syntactic-functional tags (edge labels). In particular, we address the conversion of traditional syntactic dependency annotation into the semantically motivated dependency annotation used in the Universal Dependencies (UD) Framework, evaluating this task for the Portuguese Floresta Sintá(c)tica treebank. Finally, we examine the effect of the UD converter on a rule-based dependency parser for English (EngGram). Exploiting the ensuing comparability and using the existing UD Web treebank as a gold standard, we discuss the parser's performance and the validity of UD-mediated evaluation.

1 Introduction

Dependency parsers have become a standard module in language technology program pipelines, providing structural information for higher-level tasks such as Information Extraction (Gamallo & Garcia 2012) and Machine Translation (Xu et al. 2009). Dependency links are computationally easy to process because they are token-based, but they also provide a syntactic bridge for the assignment or approximation of semantic relations. In order to facilitate such a semantic interpretation of dependency trees, some descriptive conventions within dependency grammar have moved from syntactically motivated attachment to direct links between content words, regarding function words (prepositions, auxiliaries, subordinating conjunctions) as dependents - and never heads - of content words (nouns, verbs, adjectives). Such semantic dependencies are used, for instance, to link semantic roles in the tecto-grammatical layer of the Prague Dependency treebank (Böhmová 2013), and they are also an important design feature in Universal Dependencies (McDonald et al. 2013), a new standard for dependency annotation designed to facilitate the exchange of tools and data across languages.

In addition, being a descriptive rather than a procedural standard, the Universal Dependencies (UD) framework not only makes it easier to use a given tool with input from different languages, but also to use different tools for the same language in a comparable fashion, making the output interpretable across paradigms, and allowing higher-level applications to work independently of the dependency technology used. In order for this setup to work, however, interoperability is important, and the output from existing parsers (or, in machine learning, their input from training treebanks) has to be converted into the new formalism. Syntactic conversion tasks are not a new issue: For instance, many dependency treebanks are converted versions of constituent treebanks, usually employing hand-written rules (e.g. tregex patterns, Marneffe et al. 2006). In this paper, we describe a method for the conversion of syntactic Constraint Grammar (CG) dependencies, using the same type of rules (i.e.

D S Sharma, R Sangal and A K Singh. Proc. of the 13th Intl. Conference on Natural Language Processing, pages 109–114,
Varanasi, India. December 2016.

CG) for the conversion as are used in the parser itself. This way, all contextual information can be integrated seamlessly, and unlike simple regular expressions, a CG conversion rule can make use of complex contextual constraints and relational information, such as propagated dependency links. Also, topological constraints (n-gram context or unbounded left- or right-searches) and non-topological constraints (dependencies) can be addressed at the same time, or even in the same rule. Since CG rules are run in modular batches and allow the use of environment variables or input-driven flags, language-specific conversion needs can be addressed in a flexible way within the same grammar..

2 CG Dependency rules

Constraint Grammar-rules are linguistically designed rules expressing linguistic truths in a contextual and procedural fashion. The open source CG3 formalism (Bick & Didriksen 2015), for instance, allows the grammarian to assign dependency relations based on POS context, syntactic function etc., and will even allow reference to other, already-assigned dependencies.

```
rule (a) SETPARENT (DET)
         TO (*1 N BARRIER NON-ADJ)

rule (b) SETPARENT (<mv>)
         TO (p <aux> + VFIN LINK p (*))
```

Thus, rule (a) is meant for "virgin" input without dependencies, and will attach a determiner (DET) to a noun (N) to the right (*1) with nothing but adjectives (NON-ADJ) in between. Rule (b), on the other hand, is an example of a format conversion rule, raising main verb attachment from the syntactic, finite verb auxiliary head (p=parent) to the latter's own head, whatever its type (*). With regard to punctuation, "virgin" rules were needed rather than conversion, because many parsers simply attach punctuation to either the top node or the preceding token. UD-style coordination, on the other hand, was achieved in a straight-forward fashion, since input treebank data followed the "Melczuk" tradition of sequential coordination, with a "Melczuk" flag[1] for live parses.

All in all, our conversion grammar contains 79 attachment rules in its general section. Rule order is important, and sometimes several steps are needed for one change, as in *"he wondered if David would be at the meeting"* (Fig. 1 and 2), where an object clause function has to be raised from an auxiliary to its main verb, then - if the latter is a copula - to the subject complement, and if this is a pp, yet another level to the pp's semantic head.

```
He [he] <masc> PERS 3S NOM @SUBJ>  #1->2
wondered [wonder] <mv> V IMPF @FS-STA  #2->0
if [if] <clb> KS @SUB  #3->5
David [David] <hum> PROP S NOM @SUBJ>  #4->5
would [will] <aux> V IMPF @FS-<ACC  #5->2
be [be] <mv> V INF @ICL-AUX<  #6->5
at [at] PRP @<SA  #7->6
the [the] <def> ART S/P @>N  #8->9
meeting [meeting] <occ> <def> N S NOM @P<  #9->7
```

Fig. 1: CG dependency annotation[2]

1	He	he	PRON	2	nsubj
2	wondered	wonder	VERB	0	root
3	if	if	SCONJ	9	mark
4	David	David	PROPN	9	nsubj
5	would	will	AUX	9	aux
6	be	be	VERB	9	cop
7	at	at	ADP	9	case
8	the	the	DET	9	det
9	meeting	meeting	NOUN	2	ccomp
10	.	.	PU	2	punct

Fig. 2: UD annotation[3]

[1] Coordination annotation following Melczuk attaches the second and all furhter conjuncts onto the first, e.g. attaching only the first of several coordinated direct objects to the verb, and treating the first object as the head of the others.

[2] In CG, a token has space-separated tag fields, such as word form, [lemma], <secondary tags>, POS & MORPHOLOGY, @SYNTACTIC_FUNCTION, #self-id->daughter-id. Tags used in Fig. 1 are: PERS=personal pronoun, MASC=male, 3S=third person singular, NOM=nominative, @SUBJ=subject, V=verb, IMPF=past tense, <mv>=main verb, KS=subordinating conjunction, @SUB=subordinator, PROP=proper noun, S=singular, <hum>=human, <aux>=auxiliary, @FS-<ACC=accusative [direct object] subclause, INF=infinitive, @ICL-AUX<=complement of auxiliary, PRP=preposition, @<SA=left-attached valency-bound adverbial, ART=article, <def>=definite, S/P=singular/plural, N=noun, @P<=argument of preposition

[3] Fig. 2 shows UD in CoNLL notation, here with the following TAB-separated fields: ID-nr., token, lemma, POS, head-id, edge label. There is also a field for fine-grained POS which in UD is filled with feature-attribute pairs. These are generated

Fig. 1 and 2 illustrate the substantial differences between a traditional dependency scheme, like EngGram's, and the UD convention. Thus, while the daughter of "wondered" (id 2) in Fig. 1 is an internally structured object subclause represented by its finite verb (id 5), it is a c-complement noun (9 meeting) in Fig. 2, with a shallow row of daughters, where the distinction between subordinator, subject, auxiliary, copula and preposition only resides in the so-called edge labels (mark, nsubj, aux, cop and - for prepositions - case), without any internal structure.

3 Function tag normalisation

Apart form the dependency structure itself, format conversion into the UD standard also involves the adaptation of syntactic function tags (or *edge labels*). In our scenario, this amounts to the conversion of one cross-language tag set (in our test scenario, the VISL[4] tag set) into another (UD), with a potential of being largely language-independent. Correspondences are not 1-to-1, however, with differences in granularity. Therefore, contextual rules are needed for this task, too. Rule (c), for instance, substitutes the existing edge label of an argument-of-preposition (@P<) with another label ($1 variable), harvested from the copula head of that preposition, implementing the UD principle "semantically" transparent.

```
rule (c) SUBSTITUTE (/¥.*/r) (VSTR:$[5]1)
         TARGET @P<
         (*-1 PRP LINK 0 @<SC OR @<SA)
         (p COPULA LINK 0 (/\(¥.*?\)$/r)) ;
```

In addition, some edge labels in the UD scheme are not purely syntactic, with conversion rules having to draw on morphological or semantic features from the input annotation, as for modifier edge labels that are named after the modifying POS, rather than its syntactic function with relation to the head. Thus, the VISL scheme distinguishes between free adverbials (ADVL), bound adverbials (SA), prepositional arguments (PIV) adnominal (>N, N<) and adject (>A, A<) modifiers, all of which will either be *nmod*, *amod* or *advmod* in the UD scheme, depending on whether the dependent is a noun, adjective or adverb[6]. Our general UD-converter contains about 90 edge label rules, with optional additions for the normalization of POS and morphological features (mostly local rules) and English treebank-specific rules. Because our method performs tag conversion not by means of a simple replacement table, but through the use of context-dependent rule of (almost) arbitrary complexity, it is not limited to VISL-style tags and can handle complicated many-to-many tag conversion where the necessary category information is implicit only, and distributed over different constituents or across various levels of annotation.

4 Alignment-driven changes

Attachment and label conversion are enough to make an existing parser produce output compatible with UD guidelines, but for the sake of interoperability and evaluation, tokenization can be very important, too, as well as treebank-specific handling of the internal dependencies and edge labels of complex names and multi-word expressions (MWE). Thus, in order to make converted EngGram output compatible with the UD English Web Treebank (Silveira et al. 2014), we had to add another grammar module, handling MWEs such as names, compounds and complex function words. Among other adaptations, we introduced a new rule type designed to assign separate tags and attachments to input MWEs. Thus, (d) addresses 3-part proper nouns (with '=' as separation marker), creating 3 separate tokens, <NER1-3>, with word and lemma forms taken from regular expression variables in the target MWE. In the example, * indicates the part that inherits the original POS and function, while 1->3, 2->3 indicate rightward internal attachment[7] and c->p means that the last part inherits incoming (c, child) and outgoing (p, parent) dependencies from the MWE.

by our converter, but left out in the illustration for clarity.

[4] http://beta.visl.sdu.dk/tagset_cg_general.pdf

[5]http://beta.visl.sdu.dk/treebanks.html

[6]This reflects different parser designs of function-first (CG) vs. form-first (statistical parsing), where attachments are based on either syntactic function or POS, respectively

[7]This head-last name part attachment is treebank-specific and in conflict with UD guidelines that ask for head-first attachment: http://universaldependencies.github.io/docs/u/dep/name.html

rule (d) SPLITCOHORT:threepart
("<$1>"v "$1"v <NER1> PROP @>N ¥compound 1->3 "<$2>"v "$2"v <NER2> PROP @>N ¥compound 2->3 "<$3>"v "$3"v <NER3> * c->p) TARGET ("<([^=]+?)=([^=]+?)=([^=]+?)>"r PROP) (NOT 0 (<e-word>)) ;

5 Evaluation

Though our method allows the formulation of conversion rules for any kind of dependency treebank, we chose UD conversion of two specific treebanks for testing - the Danish Arboretum treebank[8] (423.000 tokens) and the Portuguese Floresta treebank[9] (210.000 tokens, Afonso et al. 2002), both using the afore-mentioned VISL annotation style[10]. In this setting, conversion speed was about 25.000 tokens/sec on a single 4-core machine with a Linux OS. In a live parsing pipeline using rule-based parsers[11] this amounts to only a slight increase in CPU time.

5.1 Qualitative evaluaion: Floresta treebank

In the treebank conversion task, dependency arcs were changed for 52% of tokens for Danish, and 51% for Portuguese, reflecting the essential difference between "traditional" syntactic heads and UD's semantic heads. Especially affected were pp's, verb chains and predicatives. Specific UD edge labels could be assigned with a very high coverage (99.7%) for both treebanks.

In order to validate our claim that a format converter based on CG rules can be very accurately tailored to a given target annotation convention such as Universal Dependencies, we compared our own conversion of the Portuguese Floresta treebank (FlorestaUD) with the one published at the UD website for the CoNLL version of Floresta (HamleDT[12]), also based on automatic conversion (using Treex[13] and Interset[14]). Since Portuguese was added to the UD website *after* we developed our converter, our rules reflect the general annotation guidelines of the UD project, and are not based on Portuguese examples from the UD website, and any differences can thus be used to illustrate how well - or not - the two versions match the UD target guidelines[15] at http://universaldependencies.org/u/overview/syntax.html and http://universaldependencies.org/u/dep/.

For the inspected sentences (914 tokens), our CG conversion and the HamleDT conversion differed in 13.6% of dependency arcs and 9.8% of edge labels:

UD guidelines conflicts	dependency arcs	edge labels
differences CG/Hamle	13.6 %	9.8 %
Hamle UD conflicts	12.1 %	6.2 %
CG UD conflicts	0.2 %	0.5 %
both in conflict (treebank errors)	0.7 %	0.3 %
both compatible (unclear/undecided)	0.5 %	1.2 %

Table 1: Conflicts with UD guidelines

As can be seen from Table 1, our CG-based conversion achieved a satisfactory match with UD guidelines, with almost no conflicts for dependency arcs, and 10-times fewer conflicts for edge labels than in the HamleDT version. A breakdown of conflict types revealed that the discrepancy was largest for punctuation, accounting for 47% of HamleDT"s dependency arc conflicts. Since the original Floresta treebank attaches all punctuation to the root node (0), while UD guidelines ask for true syntactic attachments (e.g. highest node in a subordinated unit for paired punctuation), this is an area where conversion actually *adds* information, and using complex contextual rules - such as CG rules - becomes an obvious advantage.

[8] Available through the ELRA Catalogue of Language Resources (catalog.elra.info)
[9] Available through the Linguateca project website (http://www.linguateca.pt/floresta/)
[10] The annotation style is described at http://visl.sdu.dk/treebanks.html#VISL_dependency_trees
[11] such as the ones listed on visl.sdu.dk/constraint_grammar_languages.html
[12] http://ufal.mff.cuni.cz/hamledt
[13] http://ufal.mff.cuni.cz/treex
[14] http://ufal.mff.cuni.cz/interset
[15] Using the same annotation convention and thus making treebanks comparable across languages is the very core idea of UD, and while language-specific additions are possible, they only make sense for features not shared with the majority of languages, and none such additions are documented in the Portuguese section of the UD website.

Another systematic area of conflict were verb phrases (vp's): The UD scheme, in accordance with its semantics-over-syntax approach, sees auxiliaries as dependents of main verbs, but unlike our CG rules, HamleDT conversion seems to have no such effect on the Floresta treebank (which has syntactic dependency and auxiliaries as heads), causing on average two edge label discrepancies and two attachment discrepancies for each vp. Also, as a consequence of this conversion failure, HamleDT does not seem to be able to "see through" auxiliaries in connection with the otherwise UD-mandated copula switch[16], where both subject and copula verb become dependents of subject complements.

Edge label conflicts are fewer, but the HamleDT conversion appears to have more problems than the CG-based conversion, in particular where the change is not local/POS-based, but contextually motivated, as in the distinction between name relations and appositions, or the distinction between quantifying numerals (nummod) and others (e.g. year names or dates).

As a final topic of notorious difficulty in dependency annotations, we checked coordination ellipsis (e.g. 'he bought a hat for his wife, and a book for his daughter'), where UD suggests a 'remnant' edge label for the coordinated small clause, with dependency arcs between equivalent functions. This structure, while difficult for a live CG parse, could be correctly produced by our rules on the basis of Floresta treebank labels[17] and shallow verb attachment.

5.2 Quantitative evaluation: CG Parsing

Obviously, comparability of tools and data is a major motivation for UD conversion, so we tried to put this hypothesis to the test by going beyond an evaluation of just the conversion method, comparing live, *UD-converted* EngGram output against the English test section of the UD Web Treebank[18]. While UD-conversion did make a direct comparison possible, we also encountered a long list of unexpected problems even in the face of converted labels and attachments, caused in particular by conflicts between the publically available UD Web Treebank and official UD guidelines (e.g. name heads, punctuation attachment). Any such difference will look like a performance error in the evaluated system, while in reality it is a consistency error in the treebank. These problems were furhter aggravated by some lexicon-based "idiosyncratic" tokenization in both the UD treebank (e.g. some hyphenated words are split, some aren't) and the input parsers (that used closed-class MWEs). Forcing the latter to accept the tokenization of the former with the help of an additional preprocessor improved alignment, but at the price of potentially detrimental changes in rule performance, for instance where a contextual reference to a given MWE cannot be instantiated, because it has been split. Performance figures naturally reflect all of these issues on top of EngGram and conversion accuracy as such. In addition, the "as-is" run on force-tokenized raw test also includes errors from the morphosyntactic stage of EngGram, propagating into the dependency stage. Thus, providing the dependency stage with hand-corrected morphosyntactic input improved performance, providing a cleaner picture of structural and categorial conversion efficiency.

UD English Web Treebank test data	label failure[19]	UAS (dep)	LS (label)	LAS (both)
as is	0.3%	80.9	86.6	75.7
hand-corrected morphosyntactic input	0.2%	86.2	90.6	81.9

Table 2: Performance of Conversion Grammar

[16] Other, minor copula differences, albeit possibly intended ones, were that HamleDT extended the copula switch to clausal predicatives and that it seemed to derive copula status from the existence of a predicative argument, ending up with at least one extra copula ('ficar' - 'become'), while our own conversion implemented the general UD guidelines, with only one copula foreseen ('be'), and no switch for clausal predicatives..

[17] The Floresta treebank uses ordinary function labels for the constituents in coordination ellipsis - the same ones that would have been used in the presence of a - repeated - verb.

[18] This treebank uses the CoNLL format (Buchholz et al. 2006), for which EngGram has an export option.

[19] Cases where no rule could assign a specific UD edge label, resulting in the underspecified 'dep'.

The labelled attachment score (LAS) for dependency (81.9) matches the average in-domain system performance for English in the CoNLL 2007 shared task (80.95, Nivre et al. 2007), where the tagged input to the dependency parsers also had been hand-corrected[20]. In other words, our UD-conversion makes it possible to compare the output of a rule-based system (EngGram) to machine-learning (ML) parsers that also use the UD scheme. The converted EngGram output does fall short of CoNLL top-performance *in-domain* (89.61), but on the other hand LAS is similar to the best *cross-domain* CoNLL result (81.06), which arguably is a fairer comparison, because we are using an existing rule-based system without domain specificity as input for our UD conversion grammar. For such a system, everything is - so to say - cross-domain.

6 Perspectives

Although our method was employed and evaluated as a conversion extension for CG parsers and CG treebanks, the same type of conversion rules should in principle work for non-CG input, too, as long as dependencies and tags are expressed in a compatible fashion. Thus, similar rules could be used for linguistically transparent genre tuning of existing dependency parsers, or for adding depth and additional annotation layers to existing treebanks. Examples of the latter are missing punctuation attachment (as in the Floresta Treebank), secondary dependencies for relative pronouns and small clauses, or discourse-spanning long-distance dependencies.

References

Susana Afonso, Eckhard Bick, Renato Haber & Diana Santos. 2002. Floresta sintá(c)tica: a treebank for Portuguese, In Proceedings of LREC'2002, Las Palmas. pp. 1698-1703, Paris: ELRA

Eckhard Bick & Tino Didriksen. 2015. CG-3 - Beyond Classical Constraint Grammar. In: Beáta Megyesi: Proceedings of NODALIDA 2015, May 11-13, 2015, Vilnius, Lithuania. pp. 31-39. Linköping: LiU Electronic Press

Eckhard Bick. 2003. Arboretum, a Hybrid Treebank for Danish, in: Joakim Nivre & Erhard Hinrich (eds.), *Proceedings of TLT 2003 (2nd Workshop on Treebanks and Linguistic Theory, Växjö, November 14-15, 2003)*, pp.9-20. Växjö University Press

Alena Böhmová, Jan Hajić, Eva Hajićová and Barbora Hladká. 2003. The Prague Dependency Treebank, A Three-Level Annotation Scenario. In: Anne Abeillé (ed.): Treebanks - Building and Using Parsed Corpora. pp 103-127. Springer

Sabine Buchholz & Erwin Marsi. 2006. CoNLL-X Shared Task on Multilingual Dependency Parsing. Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X), pages 149–164, New York City. Association for Computational Linguistics. http://www.aclweb.org/anthology/W06-2920.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal dependency annotation for multilingual parsing. In Proceedings of ACL 2013

Pablo Gamallo, Marcos Garcia & Santiago Fernández-Lanza. 2012. Dependency-Based Open Information Extraction. In: Proceedings of the Joint Workshop on Unsupervised and Semi-Supervised Learning in NLP. pp. 10-18. ACL

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In: Proceedings of LREC, pp. 449-454

J. Nivre, J. Hall, S. Kübler, R. McDonald, J. Nilsson, S. Riedel, and D. Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In: Proceedings of the CoNLL Shared Task of EMNLP-CoNLL 2007, pp. 915-932

Natalia Silveira, Timothy Dozat, Marie-Catherinde de Marneffe, Samuel Bowman, Miriam Connor, John Bauer & Christopher D. Manning. 2014. A Gold Standard Dependency Corpus for English. In: Proceedings of LREC 2014.

Peng Xu, Jaeho Kang, Michael Ringgaard and Franz Och. Using a Dependency Parser to Improve SMT for Subject-object-verb-languages. In: Proceedings of NAACL '09. pp 245-253. ACL

[20]In the CoNLL task, linguist-revised treebanks were used to provide tagged input and dependency gold standards.