English Dependency Grammar

Tomas BY
Fachbereich Informatik, Universität Hamburg
Vogt-Kölln Straße 30
22527 Hamburg,
Germany,
tomas@nats.informatik.uni-hamburg.de

Abstract

There is no large and freely accessible dependency grammar for English publically available. We present a step in the direction of providing that, using a parser that produces dependency syntax trees with a grammar based on constraints. How this system models verbs phrases, conjunctions, and relative clauses is described in some detail.

Introduction

The most comprehensive dependency-based formalisation of English syntax is surely Mel'čuk and Pertsov (1987). Hudson (1990) discusses informally the dependency oriented modelling of a large part of English, and Creswell and Rambow (2003) contains a representative set of syntactic constructions in dependency format, but with less discussion.

We describe here an implementation of a parser for English using the 'Weighted Constraint Dependency Grammar' formalism (Schroeder *et al.*, 2000), which produces dependency syntax trees.

Section 1 introduces the basic concepts in dependency grammar, section 2 contains a short description of our rule format, and sections 3–7 discusses some major syntactic elements in English and how we model them. Finally, section 8 compares our approach to the formalism used in Mel'čuk and Pertsov (1987).

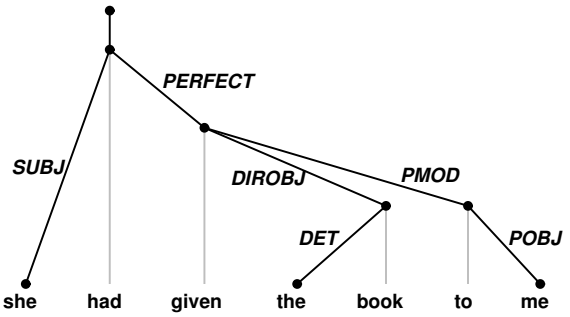
1 Heads and relation types

A dependency grammar assigns tree representations to sentences, and the form of these trees is determined by which word is considered the head of any particular phrase, and which types of links between words are available.

The criteria that have been suggested for selecting the head include that its meaning should be a hypernym of what the whole phrase refers to (as in 'book' being a hypernym of 'big book;' Hudson, 1990, p. 106), that it is the word which has the major influence on the possibilities of the phrase as a whole to combine with other words (Mel'čuk and Pertsov, 1987, p. 69; Hudson, 1990, pp. 106–7), and also that it is the word that determines the pattern of subordination within the phrase (Hudson, 1990, p. 107).

In some cases, like prepositional phrases, these criteria seem to agree, and all the cited authors consider the preposition the head of the phrase. But for noun phrases, the first criterion suggests the main noun as the head[1] while the second criterion favors the determiner.[2] Similarly, in a verb phrase, criterion three (and one perhaps) point towards the main verb being the head[3] but criterion two towards the finite verb.[4]

The tree below shows our positions. Head of the sentence is the finite verb; the determiner modifies the noun in the noun phrase, and the preposition is the head of its phrase.



As can also be seen in this example, the dependency links have labels, and the complete set of these are shown in table 1 together with the corresponding link types in Mel'čuk and Pertsov (1987, pp. 88–9), Hudson (1990, pp. 189, 233), and Creswell and Rambow (2003).

[1] As do Mel'čuk and Pertsov (1987, pp. 371–8) and Creswell and Rambow (2003).

[2] Like Hudson (1990, p. 272).

[3] Creswell and Rambow (2003).

[4] Mel'čuk and Pertsov (1987, pp. 120–1); Hudson (1990, p. 219).

Dependency label	'Surface-syntactic relation' (Mel'čuk and Pertsov, 1987)	'Grammatical relation' (Hudson, 1990)	'Surface-syntactic role' (Creswell and Rambow, 2003)
MODAL	Auxiliary	Incomplete complement	Adj
PERFECT	Auxiliary	Incomplete complement	Adj
PROGRESSIVE	Auxiliary	Incomplete complement	Adj
PASSIVE	Auxiliary	Incomplete complement	Adj
PARTICLE	Phrasal-junctive	Particle	Adj
SUBJ	Predciative	Subject	Subj
OSUBJ[5]	Predicative	Subject	Subj
—	Agentive		
DIROBJ	1st/2nd/3rd/4th completive	Object	Obj
INDOBJ	1st/2nd/3rd/4th completive	Indirect object	Obj2
PMOD	1st/2nd/3rd/4th completive	Oblique	Pobj/Pobj2
PMOD	(?)	Adjunct-complement	
—	Absolute-predicative		
—	Subjective-copredicative	Incomplete complement	
—	Pron.-subj.-copredicative	Incomplete complement (?)	
—	Objective-copredicative	Incomplete complement (?)	
—	—	Visitor[6]	
DET	Determinative	Complement	Adj
QMOD	Quantitative	Complement	
GMOD	Possessive	Complement	
NUMBER	Numeral-junctive		
POBJ	Prepositional	Complement (?)	Obj
AMOD	Modificative	Adjunct	Adj
AMOD	Descriptive-modificative	Adjunct (?)	Adj
—	Comparative		Adj
AMOD	Adverbial	Adjunct	Adj
AMOD	Modificative-adverbial	Adjunct (?)	Adj
AMOD	Appositive-adverbial	Adjunct (?)	Adj
AMOD	Attributive-adverbial	Adjunct (?)	Adj
AMOD	Compositive		
AMOD	Elective		
APPOSITION	Appositive		Adj
APPOSITION	Descriptive-appositive		Adj
AMOD	Attributive		
AMOD	Descriptive-attributive		
—	Binary-junctive		
—	Sequential		
—	Parenthetical		
AMOD	Adjunctive		
AMOD	Restrictive		
—	Colligative		
—	Expletive		
CONJ	Subordinate-conjunctional		
CONJ	Coordinate-conjunctional		
CONJ	Predicative-conjunctional		
CONJ	Completive-conjunctional		
CONJ	Absolute-conjunctional		
CONJ	Coordinative		

Table 1: Syntactic relation types

[5]This label differs from the normal SUBJ in that it is object case, as in 'He saw *her* leave.'

[6]This relation links an extracted word with the verb before which it occurs, *e.g.* the first two words in 'What do you think he said?' (Hudson, 1990, p. 192).

For efficiency reasons, the WCDG system does not allow any other nodes in the dependency graph than one for each of the original words in the sentence. In Mel'čuk and Pertsov (1987, pp. 48, 57–8) such extra nodes are allowed theoretically, but not actually used in the formalisation of English (Mel'čuk and Pertsov, 1987, pp. 85–6, 502–3). Creswell and Rambow (2003) use them for a variety of constructions: verb phrase ellipsis, control verbs, missing subjects in subordinate clauses, and conjunctions with more than two conjuncts.

2 Constraints

The formalism we are using is described in Foth *et al.* (2003) and is based on constraints on the possible combinations of word categories and dependency links.[7] One of the basic constraints, which forbids direct cycles in the syntax tree,[8] is the following.

```
{X:SYNTAX,Y:SYNTAX} : no_loops : 0 :
  X^id=Y@id -> X@id!=Y^id;
```

The first line contains a variable declaration (two edges `X` and `Y`), a constraint name, and a numerical weight. On the second line above is the actual constraint as a quasi-logical formula. The operators '`@`' and '`^`' refer to the nodes that the edge goes from and to, respectively.

3 Verb group structure

The basis of our formalisation of English verb groups is the observation by Quirk *et al.* (1985, pp. 151–3) that their structure is made up of the following four components.

MODAL Modal auxiliary followed by a base form.
PERFECT The aux. 'have' followed by past part.
PROGRESSIVE The aux. 'be' followed by pres. part.
PASSIVE The auxiliary 'be' followed by past part.

When two or more of these occur in the same phrase, they can only come in the order above. The components will also overlap partially, so that the final participle in one component and the initial auxiliary in the following component are realised by the same word in the phrase.

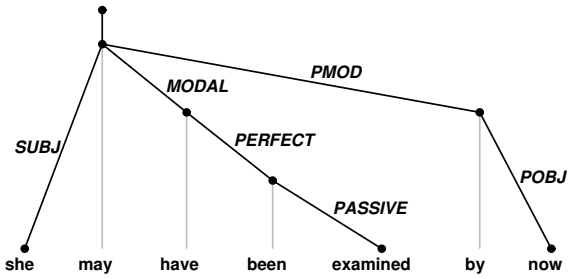
Two groups of constraints are used to formalise the structure of verb groups. The first concerns pairwise combinations of words, and restricts, for each of the verb group link types, the part-of-speech categories of the two words.

```
{X:SYNTAX} : verb_group_modal : 0 :
  X.label=MODAL -> X@form=base & X^aux=modal;
{X:SYNTAX} : verb_group_perfect : 0 :
  X.label=PERFECT -> X@form=past_part & X^aux=have;
{X:SYNTAX} : verb_group_progressive : 0 :
  X.label=PROGRESSIVE -> X@form=pres_part & X^aux=be;
{X:SYNTAX} : verb_group_passive : 0 :
  X.label=PASSIVE -> X@form=past_part & X^aux=be;
```

The other group constrains the possible longer sequences, by listing those pairwise combinations of link labels that are not allowed.

```
{X:SYNTAX\Y:SYNTAX} : verb_group_same : 0 :
  Y.label = X.label -> false;
{X:SYNTAX\Y:SYNTAX} : verb_group_perf_mod : 0 :
  Y.label=PERFECT & X.label=MODAL -> false;
{X:SYNTAX\Y:SYNTAX} : verb_group_prog_mod : 0 :
  Y.label=PROGRESSIVE & X.label=MODAL -> false;
{X:SYNTAX\Y:SYNTAX} : verb_group_prog_perf : 0 :
  Y.label=PROGRESSIVE & X.label=PERFECT -> false;
{X:SYNTAX\Y:SYNTAX} : verb_group_pas_mod : 0 :
  Y.label=PASSIVE & X.label=MODAL -> false;
{X:SYNTAX\Y:SYNTAX} : verb_group_pas_perf : 0 :
  Y.label=PASSIVE & X.label=PERFECT -> false;
{X:SYNTAX\Y:SYNTAX} : verb_group_pas_prog : 0 :
  Y.label=PASSIVE & X.label=PROGRESSIVE -> false;
```

The example below shows the analysis of 'may have been examined.'



As shown here, the auxiliary verbs form a chain of dependencies from the main verb up to the finite verb, to which the subject links, while the objects link to the main verb. Both Mel'čuk and Pertsov (1987, pp. 120–1) and Hudson (1990, pp. 219, 239–44) use a similar structure but with only one relation type,[9] so the different tenses/aspects are not distinguished in the syntax tree. Creswell and Rambow (2003) makes the main verb the root of the tree, with the auxiliaries linked below, like modifiers. This has the consequence that, for complex tenses, the finite verb and the subject, which have number agreement, can be far apart in the tree.

[7] For technical reasons, constraints can involve at most three nodes (words).

[8] *C.f.* the assumption of antisymmetry in Mel'čuk and Pertsov (1987, p. 54).

[9] 'Auxiliary' and 'Incomplete complement,' respectively, see table 1.

4 Verb frames

The types and number of the complements of verbs are encoded using the classification in Quirk *et al.* (1985, pp. 1171, 1220. 1232), shown in table 2, and the relation names we use (see the first column in table 1) are the standard grammatical terms. Hudson (1990) and Creswell and Rambow (2003) use a similar approach while Mel'čuk and Pertsov (1987, pp. 93–9) differs slightly by using the same relations (1st/2nd/3rd/4th completive), in some cases,[10] also for what would normally be considered adjuncts. The question of how to encode the verb frames in the lexicon is not discussed by the other authors cited here.

—	Intransitive verb
Copular verb (SVC & SVA)	
A1	Adjectival subject complement
A2	Nominal subject complement
A3	Adverbial complementation
Monotransitive verb (SVO)	
B1	Noun phrase as object (with passive)
B2	Noun phrase as object (without pass.)
B3	*That*-clause as object
B4	*Wh*-clause as object
B5	*Wh*-infinitive as object
B6	*To*-infinitive (no subject) as object
B7	*-ing* clause (with no subject) as object
B8	*To*-infinitive (with subject) as object
B9	*-ing* clause (with subject) as object
Complex transitive v. (SVOC & SVOA)	
C1	Adjectival object complement
C2	Nominal object complement
C3	Object + adverbial
C4	Object + *to*-infinitive
C5	Object + bare infinitive
C6	Object + *-ing* clause
C7	Object + *-ed* clause
Ditransitive verb (SVOO)	
D1	Noun phrases as objects
D2	With prepositional object
D3	Indirect object + *that*-clause
D4	Indirect object + *wh*-clause
D5	Indirect object + *wh*-infinitive clause
D6	Indirect object + *to*-infinitive
Adjective	
E1	Prepositional phrase
E2	*that*-clause
E3	*wh*-clause
E4	*than*-clause
E5	*to*-infinitive clause
E6	*-ing* participle clause

Table 2: Frame types

Since there are no 'internal' or non-terminal nodes in the dependency syntax trees, the verb complementation constraints typically refer to the head of the complement only, for example '`noun`' in the case of a noun phrase complement, as in the example below[11] which says that a verb of type A2 needs a noun (phrase) as direct object.

```
{X:SYNTAX} : dirobj_noun : verb_frame : 0 :
  X.label=DIROBJ & X^frame=a2 -> X@cat=noun;
```

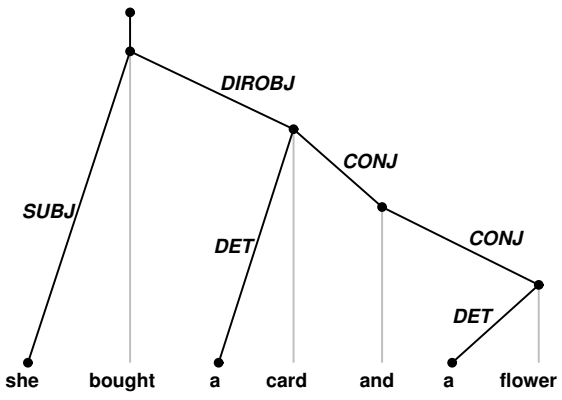
Sometimes, however, it is necessary to have further conditions on the complement. The following expression[11] says that a B8-verb needs a '`to`' (-infinitive phrase) with a subject. The expression '`has(X@id,OSUBJ)`' means that there is a `OSUBJ` link up to '`X@id`,' which is the lower node of the `DIROBJ`.

```
{X:SYNTAX} : dirobj_ingcs_subj : 0 :
  X.label=DIROBJ & X^frame=b8 ->
    X@cat=to & has(X@id,OSUBJ);
```

In total, there are 51 verb sub-categorisation constraints in our grammar.

5 Conjunctions

Conjunctions are probably among those syntactic constructions in English for which a dependency representation is least natural, and Hudson (1990, pp. 97–8 & chap. 14) advocates the use of phrase structure to model them. We, like Mel'čuk and Pertsov (1987, pp. 64–5, 153–6) and Creswell and Rambow (2003), make the left-most conjunct the head of the construction with the conjunction word and the other conjunct modifying it.



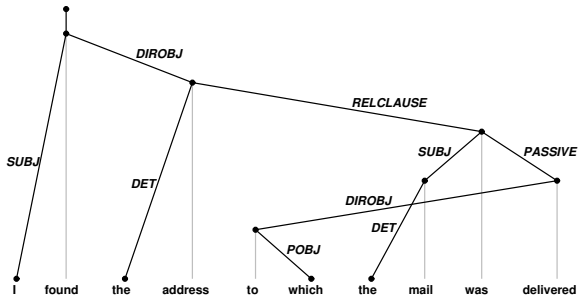
Our grammar does not include punctuation in the syntax tree, so when there is more than one conjunct, and the later pairs are separated by commas, the `CONJ` relation links the two conjuncts directly.

[10] *E.g.* the verb '`rent`' takes five arguments (including the subject): who, what, to whom, for how much, and for how long (Mel'čuk and Pertsov, 1987, p. 94).

[11] Both these constraints are simplified versions of those we use, which employ function macros.

6 Relative clauses

In relative clauses we choose to link the head of the subordinate clause to the noun it modifies, same as Creswell and Rambow (2003). This, unfortunately, violates the 'projectivity' principle (Hays, 1964, p. 519; Mel'čuk and Pertsov, 1987, pp. 183–6; Hudson, 1990, pp. 114–5), traditionally considered important in dependency grammar.



Another obvious problem here is that there is no direct link between the relative pronoun (which) and the modified noun (address).[12]

Both (Mel'čuk and Pertsov, 1987, pp. 130, 363–4) and (Hudson, 1990, pp. 383–403) use multiple links between the modifiee and the subordinate clause to model these constructions, an avenue not open to us since we restrict ourselves to only a single tree with no cycles.

7 Preferences

The WCDG constraints have weights which are real numbers between 0 and 1, indicating the severity of the constraint's failing. This ranges from not allowed to fail (0) to no penalty for failing (1).

A natural application for this feature is encoding preferences, as in the example below. An apposition must attach to either a proper or a common noun, and we set a higher penalty (*i.e.* lower weight) on the former, indicating that this is the preferred option.

```
{X:SYNTAX} : apposition_to : 0 :
  X.label=APPOSITION ->
    ( X^cat=pnoun | X^cat=cnoun );
{X:SYNTAX} : apposition_to_pnoun : 0.1 :
  X.label=APPOSITION -> X^cat=pnoun;
{X:SYNTAX} : apposition_to_cnoun : 0.5 :
  X.label=APPOSITION -> X^cat=cnoun;
```

Through other constraints, the grammar also prefers shorter '`APPOSITION`' links, *i.e.* the less there is between the modified noun and the (head of the) apposition the better.

[12] *C.f.* 'relative antecedent' (Hudson, 1990, p. 389).

8 Constraints and 'syntagms'

The WCDG constraints are similar to the 'syntagms' in Mel'čuk and Pertsov (1987) in that both formalisms express the possible combinations of words[13] and syntax relations.[14] Among the differences is that the syntagms are much richer in information, and, perhaps partly as a consequence, that that system has never been fully implemented. As an example, syntagm 27.1 (Mel'čuk and Pertsov, 1987, p. 360) says that a preposition can take a noun phrase complement.

Preposition governing a nominal phrase (27.1)

27.1.

X (Prep, **not** postpos!)
prepositional ⇔ X+ ... +Y **not** nom, **not** poss
Y ΔNP

1) Y ≠ THAT[4] [**The chair on that I am now sitting*... vs. *The chair that I am now sitting on*: see Syntagm 27.10];
2) **either** ... ⊅ Z(Prep), **or** ... ⊃ hphn Z(Prep) hphn [see (10)];
3) **if** W —(Conj,coord)→ X, **then** X comma Ỹ **is possible** [see (8) and (9)]

The interesting part here is the left side: that a 'Prepositional' link goes from a preposition (Prep) X to the head of a noun phrase subtree (NPΔ)[15] Y, not in the genitive case (**not** poss), with the surface order X, followed optionally by something else, and then Y (X + ... + Y).

Saying this in the English WCDG grammar takes, at least, the following three constraints.

```
{X:SYNTAX} : pobj_to : 0 :
  X.label=POBJ -> X^cat=prep;
{X:SYNTAX} : pobj_from : 0 :
  X.label=POBJ ->
    ( X@cat=noun
    & ( exists(X@case) -> X@case!=genitive ) );
{X:SYNTAX} : pobj_direction : 0 :
  X.label=POBJ -> X\;
```

In the last of these three constraints, '`X\`' means that the link `X` goes to the left. To avoid parsing mistakes we might also want to say that a preposition needs a complement, and that there can only be one per preposition.

```
{X:SYNTAX} : pobj_required : 0 :
  X@cat=prep -> has(X@id,POBJ);
{X:SYNTAX,Y:SYNTAX} : pobj_unique : 0 :
  X.label=POBJ & Y.label=POBJ ->
    (X^id=Y^id -> X@id=Y@id);
```

In addition, there are some more constraints with preferences (see section 7 above), for better parsing accuracy, and in total we use eleven constraints having to do with prepositions.

[13] Parts-of-speech in our case, and 'wordforms' in Mel'čuk and Pertsov (1987, p. 165).

[14] These are listed in table 1.

[15] Mel'čuk and Pertsov (1987, pp. 485–7)

Conclusions

We describe a comprehensive, implemented, dependency grammar of English, using 14 part-of-speech categories, 26 relation types, and 168 constraints on the combinations of these. It is work-in-progress and, in particular, the necessary lexicon for evaluating against a large corpus (the Penn Treebank) has not been finished.

Acknowledgements

Kilian Foth has given much help with the grammar development, and Wolfgang Menzel commented on a draft version of the paper.

References

Creswell, C. and Rambow, O.: 2003, English Dependency Treebank Coding Manual, `http://www.cis.upenn.edu/~creswell/dependency/`.

Foth, K., Hamerich, S., Schröder, I., Schulz, M. and By, T.: 2003, *[X]CDG User guide*, Fachbereich Informatik, Universität Hamburg.

Hays, D. G.: 1964, Dependency Theory: A Formalism and Some Observations, *Language* **40**(4), 511–525.

Hudson, R.: 1990, *English Word Grammar*, Basil Blackwell Ltd.

Mel'čuk, I. A. and Pertsov, N. V.: 1987, *Surface Syntax of English*, John Benjamins.

Quirk, R., Greenbaum, S., Leech, G. and Svartvik, J.: 1985, *A Comprehensive Grammar of the English Language*, Longman, London.

Rambow, O., Cresswell, C., Szekely, R., Tarber, H. and Walker, M.: 2002, A Dependency Treebank for English, *Third International Conference on Language Resources and Evaluation*, pp. 857–863.

Schroeder, I., Menzel, W., Foth, K. and Schulz, M.: 2000, Modeling dependency grammar with restricted constraints, *Traitement automatique des langues*.