Multiword Expressions in Dependency Parsing

Joakim Nivre

Uppsala University
Linguistics and Philology

Overview of the Course

1. Introduction to dependency grammar and dependency parsing
2. Graph-based and transition-based dependency parsing
3. Multiword expressions in dependency parsing
4. Practical lab session (MaltParser)

Plan for this Lecture

- Multiword expressions in dependency parsing
  - Linguistic representations
  - Parsing techniques
  - Empirical studies
- Universal Dependencies
  - General principles
  - Multiword expressions

Linguistic Representations

- How do we represent MWEs in dependency trees?
- Do we need to modify the definition of a dependency tree?
- What about different classes of MWEs?
  - Fixed: by and large, in spite of
  - Semi-fixed: part(s) of speech, kick(s/ed) the bucket
  - Flexible: put off, look for, take a photo
- What about discontiguous MWEs?

The Spanning Tree Assumption

- Basic assumption in (current) dependency parsing:
  - Dependency graph for $x = w_1, \ldots, w_n$ is a spanning tree in G_x
  - Every token is a node in the dependency tree (spanning)
  - Every node (except the root) has one incoming arc (tree)
- Possible variations:
  - Give up the tree assumption – allow general graphs
  - Give up the spanning assumption – tokens $\neq$ nodes

Tokens and Nodes

Token	Node	Example
1	1	Ordinary word tokens

Tokens and Nodes

Token	Node	Example
1	1	Ordinary word tokens
1	>1	Clitics, contractions

Tokens and Nodes

Token	Node	Example
1	1	Ordinary word tokens
1	>1	Clitics, contractions
>1	1	Multiword expressions

Tokens and Nodes

Token	Node	Example
1	1	Ordinary word tokens
1	>1	Clitics, contractions
>1	1	Multiword expressions
1	0	Punctuation?

Tokens and Nodes

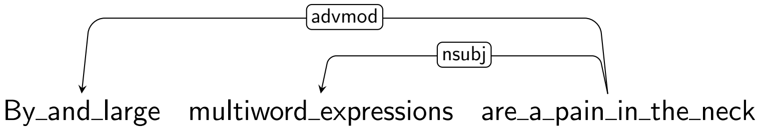
Token	Node	Example
1	1	Ordinary word tokens
1	>1	Clitics, contractions
>1	1	Multiword expressions
1	0	Punctuation?
0	1	Ellipsis?

Tokens and Nodes

Token	Node	Example
1	1	Ordinary word tokens
1	>1	Clitics, contractions
>1	1	Multiword expressions
1	0	Punctuation?
0	1	Ellipsis?

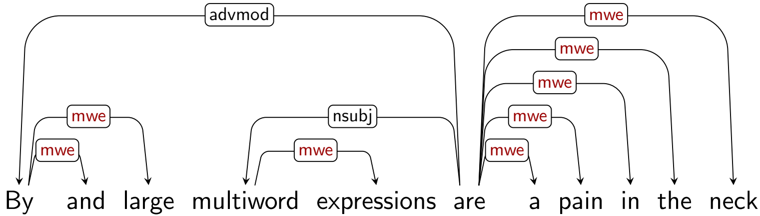
This requires a new type of dependency parser!

MWEs as Special Tokens



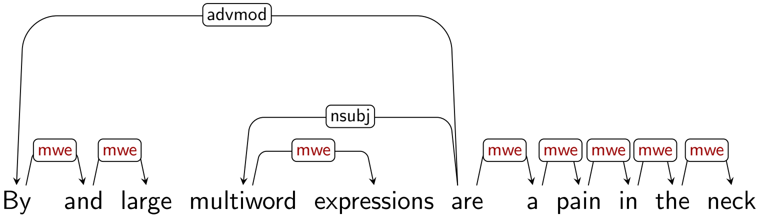
- Simplifies parsing if MWEs can be identified prior to parsing
- Limited to contiguous MWEs and awkward for flexible MWEs
- Common in treebanks (about half of the CoNLL-X data sets)
- What about part-of-speech tags?

MWEs as Dummy Dependency Structures



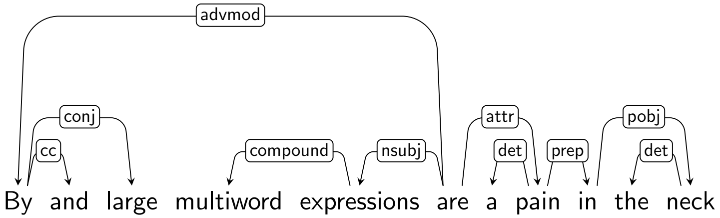
- Canonical structure without syntactic significance
- Special labels distinguish from real dependencies
- Part-of-speech tags may or may not reflect MWE-hood

MWEs as Dummy Dependency Structures



- Canonical structure without syntactic significance
- Special labels distinguish from real dependencies
- Part-of-speech tags may or may not reflect MWE-hood

MWEs as Real Dependency Structures



- Dependency structure reflects real internal structure
- Special labels may be used for subtypes (for example, LVCs)

So what representations should we use?

- Different types of MWEs require different representations
- At one end of the spectrum: by and large
  - No point in representing internal syntactic structure
  - Equivalent to a single node in dependency structure
  - Special token or dummy dependencies?
- At the other end: take a photo
  - Needs internal structure to allow modification and inflection
  - Real dependencies, special labels?
- What about everything in between?

Parsing Techniques

- Three main approaches:
  - Pre-processing – analyze MWEs before parsing
  - Post-processing – analyze MWEs after parsing
  - Joint processing – analyze MWEs during parsing
- Key question:
  - Does MWE identification help parsing or vice versa or both?
  - The answer may be different for different types of MWEs!

Techniques and Representations

	Pre	Joint	Post
Special tokens	yes	no	yes

Techniques and Representations

	Pre	Joint	Post
Special tokens	yes	no	yes
Dummy dependencies	?	yes	?

Techniques and Representations

	Pre	Joint	Post
Special tokens	yes	no	yes
Dummy dependencies	?	yes	?
Real dependencies	no	yes	yes

Techniques and Representations

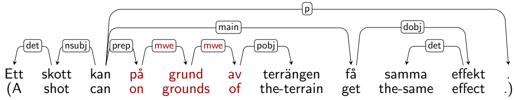
	Pre	Joint	Post
Special tokens	yes	no	yes
Dummy dependencies	?	yes	?
Real dependencies	no	yes	yes

If different types of MWEs require different representations, they may require different processing techniques as well!

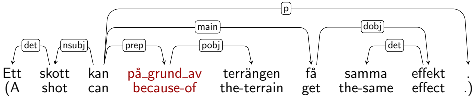
An Early Study [Nivre and Nilsson 2004]

- Swedish treebank with (limited) MWE annotation:
  - Function words like in spite of, at large
  - Names like Carl XVI Gustaf, Swedish Academy
  - Numerical expressions like $2 + 2 = 4$

1. Joint processing with dummy dependencies:



2. Preprocessing with special tokens (gold input):



Results

	MWE	Other
Joint	71.1	80.7
Preprocessing	–	81.6

- Perfect MWE recognition improves parsing accuracy (slightly)
- Typical effects of failing to recognize MWEs:
  - Unusual part-of-speech patterns leads to distorted structure (vad beträffar = as regards)
  - Different attachment preferences for MWEs and compositional phrases (i regel = as a rule)

Results

	MWE	Other
Joint	71.1	80.7
Preprocessing	–	81.6

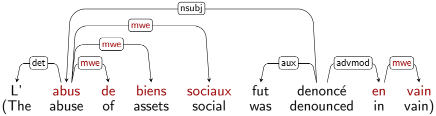
- Perfect MWE recognition improves parsing accuracy (slightly)
- Typical effects of failing to recognize MWEs:
  - Unusual part-of-speech patterns leads to distorted structure (vad beträffar = as regards)
  - Different attachment preferences for MWEs and compositional phrases (i regel = as a rule)

Similar results observed later for Turkish [Eryiğit et al. 2011]

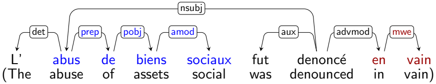
Regular and Irregular MWEs

[Candito and Constant 2014]

- French dependency treebank with dummy MWE dependencies:



- Alternative representations for regular MWEs:



- PoS patterns used to distinguish regular and irregular MWEs

Processing Models

	Irregular	Regular
Joint	Parser	Parser
Joint-Reg	Pre	Parser
Joint-Irreg	Parser	Post
Pipeline	Pre	Post

- Pre = MWEs pre-recognized and merged to single tokens
- Post = MWEs recognized after parsing

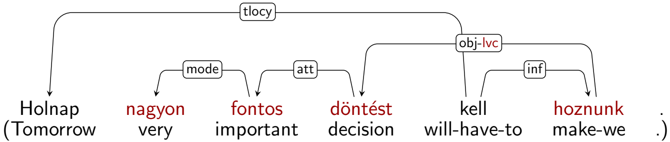
Results

	Dummy		Real	
	MWE	Overall	MWE	Overall
Joint	73.5	84.5	81.4	86.9
Joint-Reg	73.3	84.2	80.4	86.6
Joint-Irreg	75.4	84.4	82.1	87.0
Pipeline	74.4	83.9	80.4	86.5

- Real dependencies better than dummy dependencies
- Irregular MWEs benefit most from joint processing
- Regular MWEs better identified after parsing?

Light Verb Constructions [Vincze et al. 2013]

- Hungarian dependency treebank with LVC annotation:



- Can a parser learn to identify light verb constructions?
- How is overall parsing accuracy affected?

Results

	LVC	Overall
Parser plain	–	90.6
Parser LVC	75.6	90.4
Post dictionary	21.3	–
Post C4.5	74.5	–

- Parser improves LVC identification with a marginal drop in overall labeled attachment score
- Parser significantly better than post-classifier on discontiguous LVCs ($64.0 > 60.0$)

Conclusion

- We have only scratched the surface ...
- Complex interaction between several factors:
  - MWE types
  - Linguistic representations
  - Processing techniques
- Tentative conclusions:
  - MWE identification can benefit from syntactic context
  - Regular MWEs should be assigned regular syntactic structure

Universal Dependencies

- Background:
  - Treebank annotation schemes vary across languages
  - Hard to compare results across languages [Nivre et al. 2007]
  - Hard to evaluate cross-lingual learning [McDonald et al. 2013]
  - Hard to build multilingual systems
- Universal Dependencies (http://universaldependencies.github.io/docs/):
  - Stanford universal dependencies [de Marneffe et al. 2014]
  - Google universal part-of-speech tags [Petrov et al. 2012]
  - Interset morphologial features [Zeman 2008]

Universal Dependencies

- Background:
  - Treebank annotation schemes vary across languages
  - Hard to compare results across languages [Nivre et al. 2007]
  - Hard to evaluate cross-lingual learning [McDonald et al. 2013]
  - Hard to build multilingual systems
- Universal Dependencies (http://universaldependencies.github.io/docs/):
  - Stanford universal dependencies [de Marneffe et al. 2014]
  - Google universal part-of-speech tags [Petrov et al. 2012]
  - Interset morphologial features [Zeman 2008]

First guidelines released Oct 1, 2014

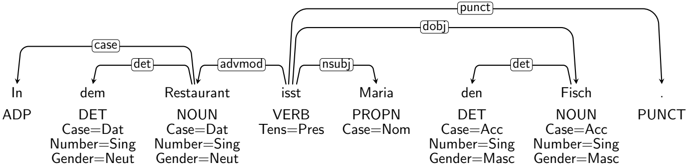
Universal Dependencies

- Background:
  - Treebank annotation schemes vary across languages
  - Hard to compare results across languages [Nivre et al. 2007]
  - Hard to evaluate cross-lingual learning [McDonald et al. 2013]
  - Hard to build multilingual systems
- Universal Dependencies (http://universaldependencies.github.io/docs/):
  - Stanford universal dependencies [de Marneffe et al. 2014]
  - Google universal part-of-speech tags [Petrov et al. 2012]
  - Interset morphologial features [Zeman 2008]

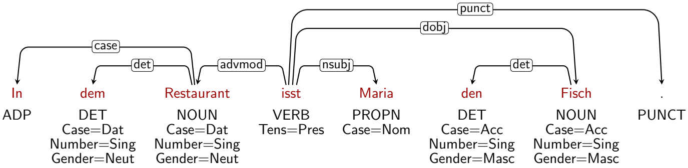
First guidelines released Oct 1, 2014

First 10 treebanks released Jan 15, 2015

Universal Dependencies

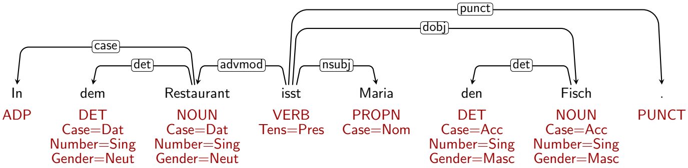


Universal Dependencies



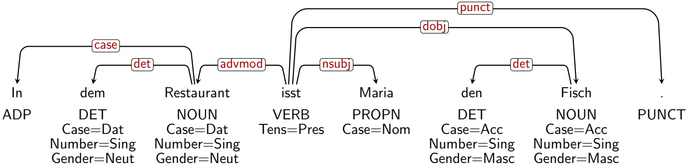
- Syntactic words – explicit splitting of clitics and contractions

Universal Dependencies



- Syntactic words – explicit splitting of clitics and contractions
- Universal part-of-speech tags + morphological features

Universal Dependencies

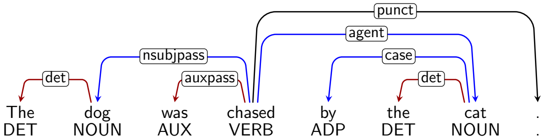


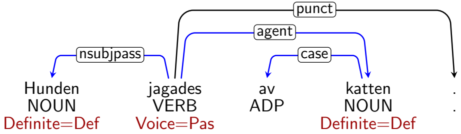
- Syntactic words – explicit splitting of clitics and contractions
- Universal part-of-speech tags + morphological features
- Dependency tree + augmented dependencies (not shown)

Guiding Principles

- Maximize parallelism
  - Don't annotate the same thing in different ways
  - Don't make different things look the same
- But don't overdo it
  - Don't annotate things that are not there
  - Languages select from a universal pool of categories
  - Allow language-specific extensions

Dependency Structure





- Keeping content words as heads promotes parallelism
- Function words often correlate with morphology

Dependency Relations [de Marneffe et al. 2014]

Core dependents of clausal predicates

Nominal dep	*Predicate dep*	
nsubj	csubj	
nsubjpass	csubjpass	
dobj	ccomp	xcomp
iobj		

Non-core dependents of clausal predicates

Nominal dep	*Predicate dep*	*Modifier word*
nmod	advcl	advmod
		neg

Special clausal dependents

Nominal dep	*Auxiliary*	*Other*
vocative	aux	mark
discourse	auxpass	punct
expl	cop	

Noun dependents

Nominal dep	*Predicate dep*	*Modifier word*
nummod	acl	amod
appos		det
nmod		neg

Compounding and unanalyzed

compound	mwe	goeswith
name	foreign	

Coordination

conj	cc	punct

Case-marking, prepositions, possessive

case

Loose joining relations

list	parataxis	remnant
dislocated		reparandum

Other

Sentence head	*Unspecified dependency*
root	dep

- Taxonomy of 42 universal grammatical relations, broadly supported across many languages in language typology
- Language specific subtypes can be added

Morphology

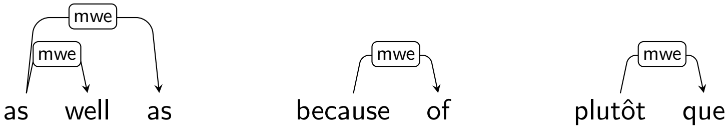
Open class words	Closed class words	Other
ADJ	ADP	PUNCT
ADV	AUX	SYM
INTJ	CONJ	X
NOUN	DET	
PROPN	NUM	
VERB	PART	
	PRON	
	SCONJ	

- Taxonomy of 17 universal part-of-speech tags, based on the Google Universal Tagset [Petrov et al. 2012]
- Standardized inventory of morphological features, based on the Interset system [Zeman 2008]

MWEs in Universal Dependencies

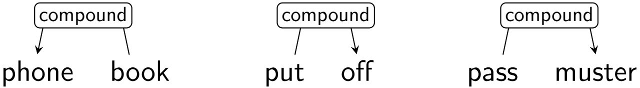
- UD does not allow merged tokens: in spite of $\not\to$ in_spite_of
- MWEs have to be encoded with (dummy or real) dependencies
- Three relations currently used:
  - mwe: fixed grammaticized expressions
  - compound: lexical compounds of any category
  - name: multiword proper names

The mwe relation



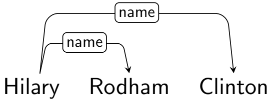
- Used for fixed grammaticized expressions that behave like function words or short adverbials
- Annotated in a flat, head-initial structure, where all words in the expression modify the first one using the mwe label

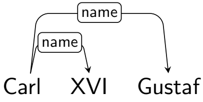
The compound relation



- Used for lexical compounds, including nominal compounds and particle verbs
- Annotated to reflect headness properties

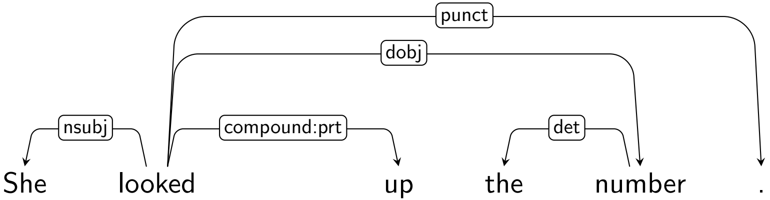
The name relation





- Used for proper nouns constituted of multiple nominal elements, but not for for phrasal or clausal names (The King of Sweden, Gone with the Wind)
- Annotated in a flat, head-initial structure, where all words in the name modify the first one using the name label

Language-Specific Subtypes



- We can define language-specific subtypes of universal relations
- This holds for MWE-type relations as well
- Examples in the first release:
  - compound:prt used for verb particles in several lanugages
  - nsubj:lvc, dobj:lvc used for LVCs in Hungarian

Want to Build a Better Mousetrap?

- Universal Dependencies is an open and evolving standard
  - Version 1 of the guidelines released Oct 1, 2014
  - Will be kept stable for at least a year
  - We need your help to improve the next version
  - We also need data from more languages

Coming Up Next

1. Introduction to dependency grammar and dependency parsing
2. Graph-based and transition-based dependency parsing
3. Multiword expressions in dependency parsing
4. Practical lab session (MaltParser)
   - Choose a language from the first UD release
   - Train and evaluate a dependency parser
   - Analyze parsing performance with respect to MWEs

References and Further Reading

- Marie Candito and Matthieu Constant. 2014. Strategies for contiguous multiword expression analysis and dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 743–753.
- Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. 2014. Universal Stanford Dependencies: A cross-linguistic typology. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, pages 4585–4592.
- Gülşen Eryiğit, Tugay Ilbay, and Ozan Arkan Can. 2011. Multiword expressions in statistical dependency parsing. In *Proceedings of the Second Workshop on Statistical Parsing of Morphologically Rich Languages*, pages 45–55.
- 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 the 51st Annual Meeting of the Association for Computational Linguistics. Volume 2: Short Papers*, pages 92–97.

- Joakim Nivre and Jens Nilsson. 2004.
  Multiword units in syntactic parsing. In *Proceedings of the Workshop on Methodologies and Evaluation of Multiword Units in Real-World Applications (LREC)*, pages 39–46.
- Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007.
  The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task 2007*, pages 915–932.
- Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012.
  A universal part-of-speech tagset. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*.
- Veronika Vincze, János Zsibrita, and István Nagy T. 2013.
  Dependency parsing for identifying hungarian light verb constructions. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 207–215.
- Daniel Zeman. 2008.
  Reusable tagset conversion using tagset drivers. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*, pages 213–218.