ParTAGe, (yet another) parser for TAGs

Jakub Waszczuk

February 21, 2018
TreeGraSP Meeting #1

Plan

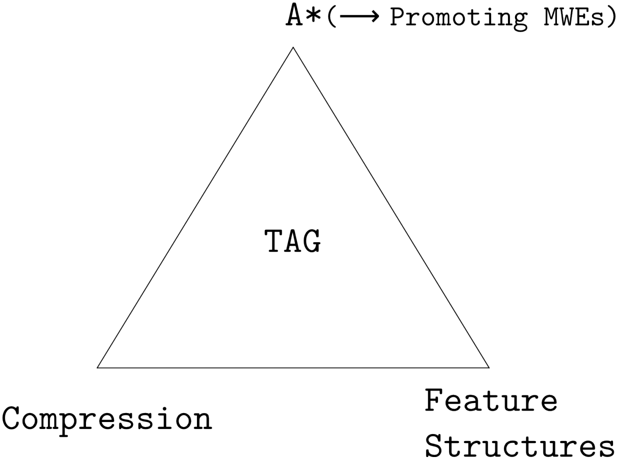
Background

ParTAGe

- Developed as a part of my PhD thesis at the University of Tours (advisors: Agata Savary and Yannick Parmentier)

- Why? TAGs are convenient for modeling MWEs and their idiosycracies
- Designed as a PhD thesis playground rather than an industrial-strength parser

ParTAGe



Plan

Introduction

TAG A* Parsing

Promoting MWEs

Grammar Compression

Feature Structures

Future Work

A* Parsing

Why A* parsing for TAGs?

- Parsing time complexity: polynomial in the sentence length and linear in the grammar size ($\mathcal{O}(n^6 * |G|)$) [Gardent et al., 2014]
  - Too costly for practical NLP applications
- A* parsing: speed up via reduction of the parsing search space
  - The first derivation found is the most probable one
- LTAGs: $\mathcal{O}(n^6) \rightarrow \mathcal{O}(n^2)$, under favorable circumstances

Motivating work

A* CCG parsing [Lewis and Steedman, 2014]

- The weight of a derivation = the sum of the weights of the participating CCG categories
- Weights estimated on a per-sentence basis
- The result quick and accurate (on par with SOA CCG parsers)

Can we apply this idea to TAGs?

- MWEs represented in TAG as multi-anchored elementary trees [Abeillé and Schabes, 1989]
- [Lewis and Steedman, 2014]: no support for multi-anchored units

Handling MWEs in A* TAG parsing [Waszczuk et al., 2016b]

Statistical characterization

Weight of a derivation is a sum of the weights of the participating elementary trees (ETs)

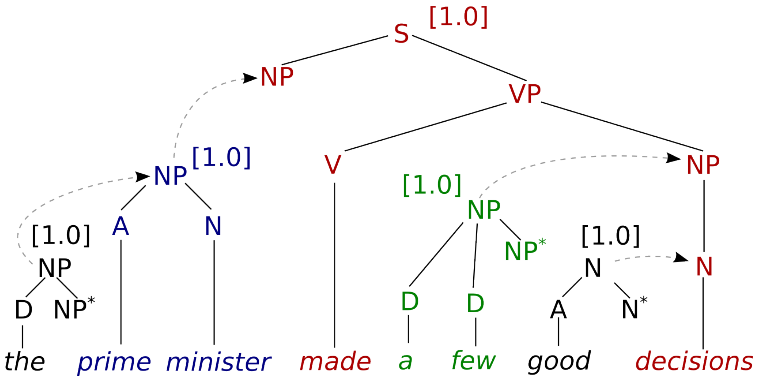


Figure: The weight of the MWE-based derivation = 5.

Weight of a compositional analysis

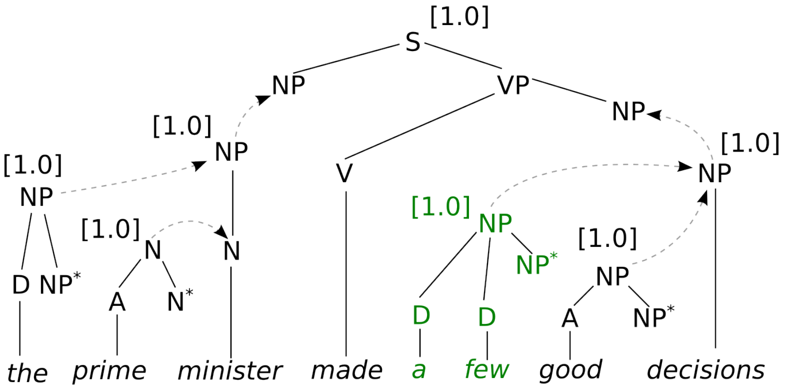


Figure: The weight of the compositional derivation = 7, it is thus **less probable** than the MWE-based derivation.

Weighted inference rules

AX:	$\dfrac{}{\mathbf{0} : (N \to \bullet \alpha, (i,i))}$	$i \in \{0, \ldots, n-1\}$ $N \to \alpha$ is a rule
SC:	$\dfrac{\mathbf{w} : (N \to \alpha \bullet M \beta, (i,j,k,l))}{\mathbf{w} : (N \to \alpha M \bullet \beta, (i,j,k,l+1))}$	$\ell(M) = s_{l+1}$
DE:	$\dfrac{\mathbf{w} : (N \to \alpha \bullet, (i,j,k,l))}{\mathbf{w} + \mathbf{w_N} \cdot [root(\mathbf{N})] : (N, (i,j,k,l))}$	w_N is the weight of the corresponding ET
PS:	$\dfrac{\mathbf{w_1} : (N \to \alpha \bullet M \beta, (i,j,k,l)) \quad \mathbf{w_2} : (M, (l,j',k',l'))}{\mathbf{w_1} + \mathbf{w_2} : (N \to \alpha M \bullet \beta, (i, j \oplus j', k \oplus k', l'))}$	
SU:	$\dfrac{\mathbf{w_1} : (N \to \alpha \bullet M \beta, (i,j,k,l)) \quad \mathbf{w_2} : (R, (l,l'))}{\mathbf{w_1} + \mathbf{w_2} : (N \to \alpha M \bullet \beta, (i,j,k,l'))}$	$\neg foot(M) \wedge \ell(M) = \ell(R)$ $root(R)$
FA:	$\dfrac{\mathbf{w_1} : (N \to \alpha \bullet F \beta, (i,l)) \quad \mathbf{w_2} : (M, (l,j',k',l'))}{\mathbf{w_1} : (N \to \alpha F \bullet \beta, (i,l,l',l'))}$	$foot(F) \wedge \ell(M) = \ell(F)$ $root(M) \implies (j',k') = (-,-)$
RA:	$\dfrac{\mathbf{w_1} : (R, (i,j,k,l)) \quad \mathbf{w_2} : (M, (j,j',k',k))}{\mathbf{w_1} + \mathbf{w_2} : (M, (i,j',k',l))}$	$root(R) \wedge \ell(R) = \ell(M)$ $root(M) \implies (j',k') = (-,-)$

Table: Weighted inference rules of an Earley-style, bottom-up TAG parser [Alonso et al., 1999] (N, M, R, F are ET nodes, $\ell(N)$ is the (non-)terminal with which N is decorated, and α, β are sequences of nodes)

A* TAG parsing

A* heuristic

A* parsing algorithm requires a **heuristic** – a lower-bound estimate on the cost of parsing the remaining part of the sentence

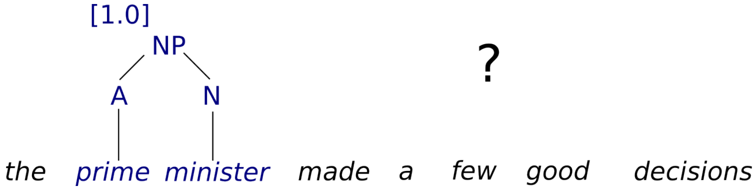
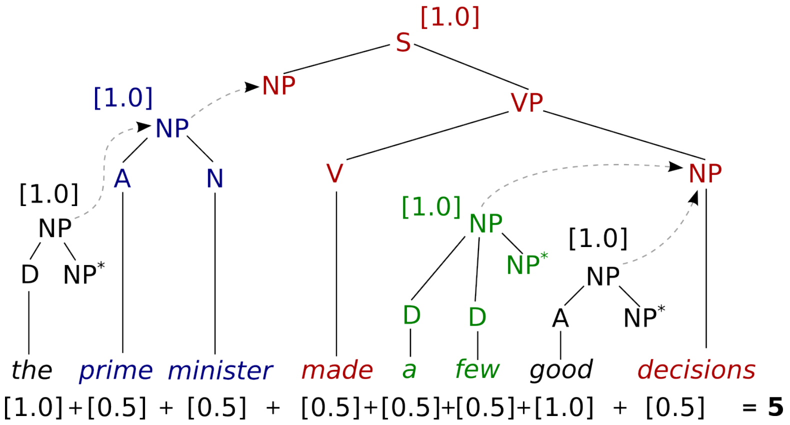


Figure: A hypothetical parsing configuration considered by the parser: *prime minister* analyzed as a MWE, the cost of parsing the remaing part of the sentence to be determined.

MWE-driven heuristic

Projecting weights on words

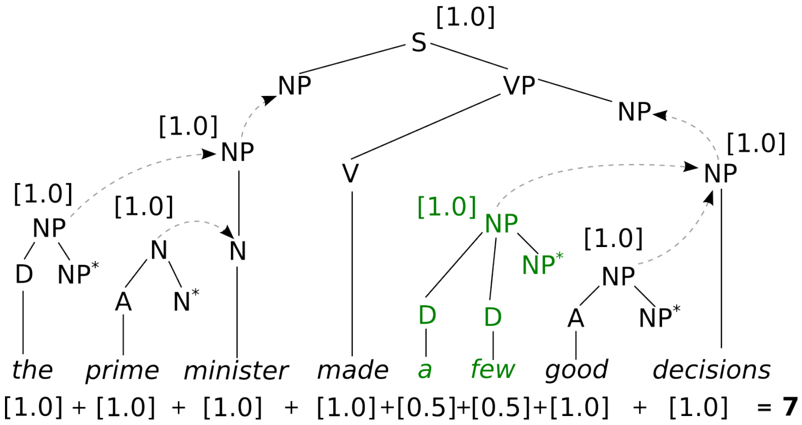
- Weights of ETs are projected on words
- Weight of a derivation = sum of the weights projected on words



MWE-driven heuristic

Projecting weights on words

- Weights of ETs are projected on words
- Weight of a derivation = sum of the weights projected on words



MWE-driven heuristic

Hypotheses

The **lowest possible weight** will be projected over each of the remaining words.

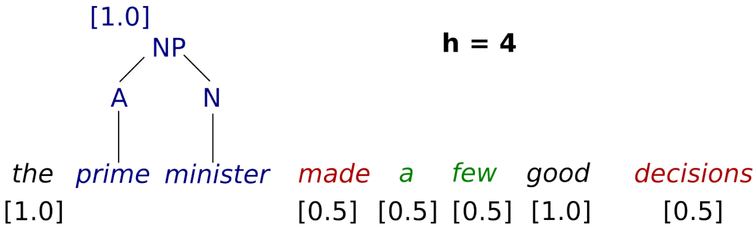
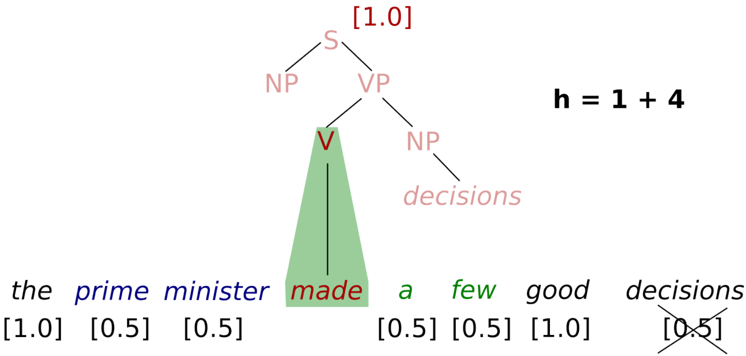


Figure: The parser recognized *prime minister* as a MWE. It still needs to parse the remaning words: *the* and *made a few good decisions*.

MWE-driven heuristic

Accounting for the tree t being matched

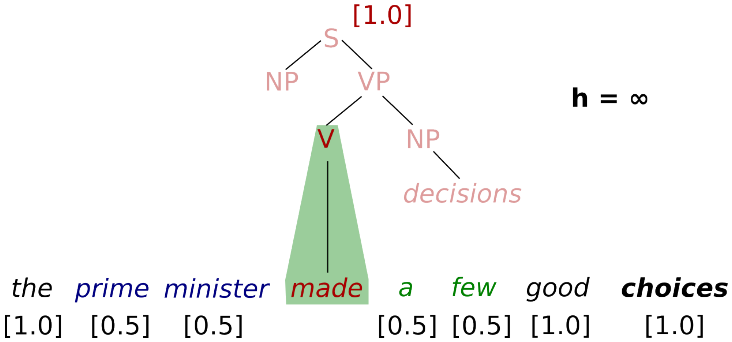
The heuristic accounts fot the weight of t, and ignores the terminals outside the current span and still required to fully match t.



MWE-driven heuristic

Dead-end detection for the tree t being matched

The heuristic returns ∞ when the terminals still required to fully match t are not present in the remaining part of the sentence.



MWE-driven heuristic

Properties

Admissible but **not** monotonic (due to gap-related predictions)

- Seems to be correct anyway [Nederhof, 2003]

Monotonic version

Trace via inference rules the weights projected over the gap

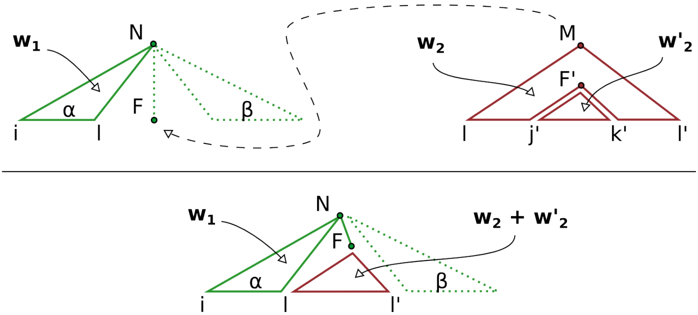


Figure: Graphical representation of the FA rule

Plan

Motivating work

Promoting collocations [Wehrli et al., 2010]

- Promoting strong collocations (in particular, MWEs): an effective way of dealing with syntactic ambiguity

How to obtain such behavior in $A^\star$ TAG parsing?

- Assign the weight 1 to each ET in the grammar

Experimental Evaluation [Waszczuk et al., 2016b]

Evaluation protocol

- **Dataset**: the Składnica treebank [Świdziński and Woliński, 2010] annotated with MWEs [Savary and Waszczuk, 2017]
- **Grammar**: MWE-aware, extracted from Składnica
- **Preprocessing**: lexical selection + compression
- **Evaluation**: run the A* parser and measure the search-space-size reductions stemming from promoting MWEs

Results

- Virtually 100% correct syntactic analysis (w.r.t baseline)
- Around 95% correct MWE identification
- Search-space reductions of 18.1% on average and up to 90.6%

Plan

Grammar compression

Dual grammar representation [Waszczuk et al., 2016a]

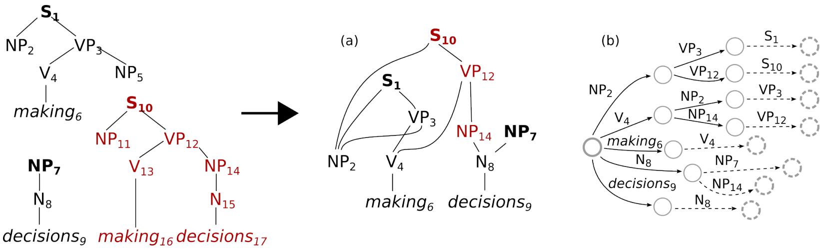


Figure: (a) subtree sharing [Schabes and C. Waters, 1995] (b) FSA-based representation of dotted rules [Nederhof, 1998]

Consequences for parsing

+ Chart items get conflated $\Longrightarrow$ computation gets smaller
+ Can be applied to symbolic parsing (no weights)

Combining grammar compression with A*

Challenges

- Under compression, a chart item can correspond to traversals of many different ETs
- The values of the heuristic can still be computed in $\mathcal{O}(1)$ time[1]
- Applying inference rules in $\mathcal{O}(1)$ not yet achieved in the current implementation

[1] At least as long as no dead-end detection is performed

Plan

Parsing with feature structures in ParTAGe

Principles

+ Unification (i) explicitely handled in the inference rules and (ii) performed on-the-fly (rather than in post-processing [Parmentier et al., 2008, Koller, 2017])
  + Should allow for better integration with $A^\star$
+ Support for generic unification-like computations over derivation trees (optional top/bottom FS distinction, flat or nested FSs, adjunction constraints, etc.)
+ Composes smoothly with compression (subtree sharing and prefix-tree representation of dotted rules)
− Not integrated with $A^\star$ yet

Implementation

- A version which supports XMG-generated TAGs with flat FSs is available at `https://github.com/kawu/partage4xmg`

Parsing with feature structures in ParTAGe

Example

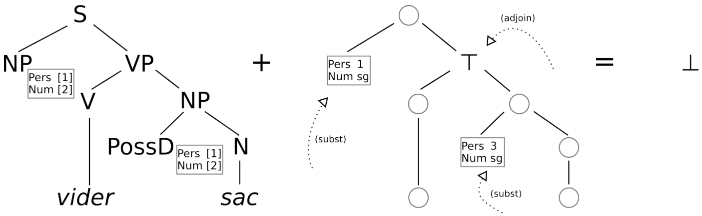


Figure: A graphical representation of a bottom-up unification computation given an ET decorated with FSs and unification variables (on the left) and a tree of FSs originating from adjunctions and substitutions (on the right).

Parsing with feature structures in ParTAGe

Disadvantages

- Unification performed on entire ETs, rather than at the moment of substitution/adjunction
- Trees over FSs are stored in chart items (complexity issue)
- No sharing of common FS parts

Plan

Future work

- Repeat the experimental evaluation of promoting MWEs with a truly weighted grammar
- Relax the assumption of the independence between ETs [Resnik, 1992, Yoshikawa et al., 2017]
- Find the right balance between A*, compression, and FSs
  $\Rightarrow$ develop the corresponding unified implementation

Thank you!

References I

Abeillé, A. and Schabes, Y. (1989).
Parsing Idioms in Lexicalized TAGs.
In *Proceedings of the Fourth Conference on European Chapter of the Association for Computational Linguistics*, EACL '89, pages 1–9, Stroudsburg, PA, USA. Association for Computational Linguistics.

Alonso, M., Cabrero, D., de la Clergerie, E. V., and Ferro, M. V. (1999).
Tabular algorithms for TAG parsing.
In *EACL 1999*, pages 150–157.

Gardent, C., Parmentier, Y., Perrier, G., and Schmitz, S. (2014).
Lexical Disambiguation in LTAG using Left Context.
In Vetulani, Z. and Mariani, J., editors, *Human Language Technology. Challenges for Computer Science and Linguistics. 5th Language and Technology Conference, LTC 2011, Poznan, Poland, November 25-27, 2011, Revised Selected Papers*, volume 8387, pages 67–79. Springer.

References II

Koller, A. (2017).

A feature structure algebra for FTAG.

In *13th International Workshop on Tree-Adjoining Grammar and Related Formalisms*, 13th International Workshop on Tree-Adjoining Grammar and Related Formalisms, Umeå, Sweden.

Lewis, M. and Steedman, M. (2014).

A* CCG Parsing with a Supertag-factored Model.

In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 990–1000. Association for Computational Linguistics.

Nederhof, M.-J. (1998).

An alternative lr algorithm for tags.

In *COLING 1998 Volume 2: The 17th International Conference on Computational Linguistics*.

Nederhof, M.-J. (2003).

Weighted Deductive Parsing and Knuth's Algorithm.

Comput. Linguist., 29(1):135–143.

References III

Parmentier, Y., Kallmeyer, L., Lichte, T., Maier, W., and Dellert, J. (2008).

TuLiPA: A Syntax-Semantics Parsing Environment for Mildly Context-Sensitive Formalisms.

In *9th International Workshop on Tree-Adjoining Grammar and Related Formalisms (TAG+9)*, pages 121–128, Tübingen, Germany.

Resnik, P. (1992).

Probabilistic tree-adjoining grammar as a framework for statistical natural language processing.

In *COLING 1992 Volume 2: The 15th International Conference on Computational Linguistics*.

Savary, A. and Waszczuk, J. (2017).

Projecting multiword expression resources on a polish treebank.

In *Proceedings of the 6th Workshop on Balto-Slavic Natural Language Processing*, pages 20–26, Valencia, Spain. Association for Computational Linguistics.

References IV

Schabes, Y. and C. Waters, R. (1995).
Tree insertion grammar: A cubic-time, parsable formalism that lexicalizes context-free grammar without changing the trees produced.
Computational Linguistics, Volume 21, Number 4, December 1995.

Waszczuk, J., Savary, A., and Parmentier, Y. (2016a).
Enhancing practical TAG parsing efficiency by capturing redundancy.
In *21st International Conference on Implementation and Application of Automata (CIAA 2016)*, Proceedings of the 21st International Conference on Implementation and Application of Automata (CIAA 2016), Séoul, South Korea.

Waszczuk, J., Savary, A., and Parmentier, Y. (2016b).
Promoting multiword expressions in A* TAG parsing.
In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 429–439.

References V

Wehrli, E., Seretan, V., and Nerima, L. (2010).
Sentence analysis and collocation identification.
In *Proceedings of the Workshop on Multiword Expressions: from Theory to Applications (MWE 2010)*, pages 27–35, Beijing, China. Association for Computational Linguistics.

Yoshikawa, M., Noji, H., and Matsumoto, Y. (2017).
A* CCG parsing with a supertag and dependency factored model.
CoRR, abs/1704.06936.

Świdziński, M. and Woliński, M. (2010).
Towards a Bank of Constituent Parse Trees for Polish.
In Sojka, P., Horák, A., Kopeček, I., and Pala, K., editors, *Text, Speech and Dialogue: 13th International Conference, TSD 2010, Brno, Czech Republic*, volume 6231 of *Lecture Notes in Artificial Intelligence*, pages 197–204, Heidelberg. Springer-Verlag.