Towards a MWE-driven A* parsing with LTAGs [WG2,WG3]

Jakub Waszczuk, Agata Savary

Towards a MWE-driven A* parsing with LTAGs [WG2,WG3]

Jakub Waszczuk, Agata Savary
Université François Rabelais Tours, Laboratoire d'informatique, France
first.last@univ-tours.fr

Natural language parsing is known to potentially produce a high number of syntactic interpretations for a sentence. Some of them may contain multiword expressions (MWEs) and achieving them faster than compositional alternatives proved efficient in symbolic parsing (see below). We propose to apply this strategy to symbolic LTAG (Lexicalized Tree Adjoining Grammar) parsing using an architecture adaptable to probabilistic parsing.

We are particularly interested in LTAGs because, according to (Abeillé and Schabes 1989), they show several advantages with respect to parsing MWEs. Firstly, unification constraints on feature structures attached to tree nodes allow one to naturally express dependencies between arguments at different depths in the elementary trees (as in NP_0 *vider DET sac* 'to express one's secret thoughts', where the determiner *DET* embedded in the direct object must agree in person and number with the subject NP_0). Secondly, the so-called *extended domain of locality* offers a natural framework for representing two different kinds of discontinuities. Namely, discontinuities coming from the internal structure of a MWE are directly visible in elementary trees and are handled in parsing mostly by substitution. Discontinuities coming from insertion of modifiers (e.g. *a bunch of NP*, *a whole bunch of NP*) are invisible in elementary trees but are handled in parsing by adjunction.

Consider the sentence in example (1).

(1) **Acid rains** in Ghana are equally grim.

When it is being scanned by a left-to-right parser, two competing interpretations are syntactically valid for the first 4 words. One of them considers *rains* as a verb whose subject is *acid* while, according to the other, *rains* is the head noun of the NN compound *acid rains*. Our objective is to propose a parsing strategy which would promote the latter interpretation due the fact that it contains a known MWE. More precisely, the parser should: (i) trivially, admit only grammar-compliant analyses of a sentence, (ii) achieve MWE-oriented interpretations more rapidly than potential compositional interpretations, (iii) eliminate no grammar-compliant interpretations.

Note that all these conditions could rather easily be met for sentence (1) in a pre-processing-based approach in which potential MWEs are identified prior to parsing and conflated into word-with-spaces tokens. Such an approach might however lead to a parsing failure in the case of sentence (2) if the two initial tokens are wrongly merged into a nominal compound in the pre-parsing step. In order to avoid errors of this kind, MWE identification and parsing should be performed jointly.

(2) **Hunger strikes** the civilians since 2001.

Seminal works, such as (Finkel and Manning 2009, Green *et al.* 2011, 2013, Constant *et al.* 2013), show that the results of probabilistic MWE identification and/or parsing are improved when both tasks are performed simultaneously. (Wehrli *et al.* 2010) point out that such an improvement (also within further parsing-based applications, e.g. machine translation) occurs in symbolic parsing (here: in a Chomskian grammar-based approach) when the knowledge about a potential occurrence of MWEs guides the parsing process.

Our goal is to apply a similar strategy to the one in (Wehrli *et al.* 2010), i.e. to systematically promote MWE-oriented interpretations, within LTAG parsing[1] We additionally wish to design the parser architecture in such a way that corpus-based probabilities about MWE contexts can be

[1]The parsing algorithm should of course abstract away from the way the input LTAG grammar was obtained (manually crafted, generated from a metagrammar, or learned from a treebank).

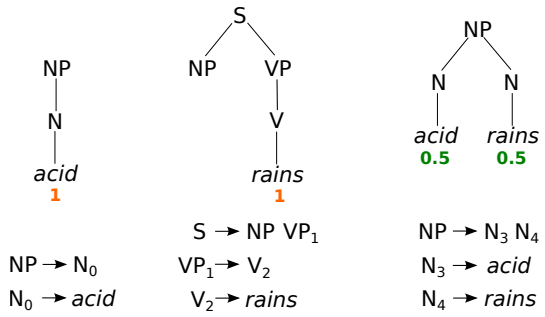

Figure 1: A toy LTAG grammar and its conversion into flat rules

easily injected into it as soon as they are available (we have performed no experiments to obtain them yet)

Note that promoting MWEs will of course be inaccurate for sentence (2). However: (i) the correct interpretation will not be discarded (it will simply be followed later than the MWE-oriented one), (ii) (Wehrli *et al.* 2010) shows that giving high priority to certain types of MWEs in parsing is a good strategy on average.

LTAG with weighted terminals Our parser relies on a particular LTAG grammar representation in which each elementary LTAG tree is converted into a set of flat production rules[2], similarly to (Alonso *et al.* 1999). Fig. 1 illustrates this conversion on a set of 3 elementary trees[3]. Note that the non-terminal *N* occurring 3 times in this grammar is represented by 3 different non-terminals N_0, N_3 and N_4 in the target rules.[4] This distinction is necessary in order to prevent non-compatible subtree combinations. For instance, we should not admit an NN-compound *rains acid* (which would be admitted if the two *N* terminals from the 3^{rd} tree were not distinguished in the resulting production rules).

We admit a version of the grammar in which each elementary tree has the same weight (equal to 1) i.e. the same probability of being used in parsing a sentence. This weight is then distributed equally over all terminal nodes occurring in the tree. Here, the terminal nodes *acid* and *rains* have weight 1 in each of the 1^{st} two trees, while they have weight 0.5 in the 3^{rd} tree.

Parsing as a hypergraph We propose an Early-style parsing algorithm for LTAGs inspired by (Klein and Manning 2001). The parsing process is represented here as a hypergraph (Gallo *et al.* 1993) whose nodes are parsing chart states, and whose hyperarcs represent applications of inference rules, i.e. combinations of previous chart states resulting in new states. The appendix shows a fragment of the hypergraph created while parsing the two initial words of sentence (1) with the grammar from Fig. 1. For instance, the hyperarc leading from the initial state $(N_3 \rightarrow \bullet acid, 0, 0)$ to state $(N_3 \rightarrow acid\bullet, 0, 1)$ indicates that the terminal $acid$ has been recognized over the sentence span from position 0 to 1. The latter state can then be combined with state $(NP \rightarrow \bullet N_3 N_4, 0, 0)$ yielding a new state $(NP \rightarrow N_3 \bullet N_4, 0, 1)$, and so on. The whole sentence is successfully parsed if a state has been reached whose underlying rule has the S symbol in its head and the dot at the end of its body, and whose span goes from 0 to the length of the sentence.

Note that some hyperarcs, namely those corresponding to scanning a symbol from the input, are weighted with the values stemming from the corresponding terminal nodes in the grammar. For instance the hyperarc from $(N_0 \rightarrow \bullet acid, 0, 0)$ to $(N_0 \rightarrow acid\bullet, 0, 1)$ has weight 1 since its underlying rule $N_0 \rightarrow acid$ stems from the 1^{st} tree in Fig. 1, while the hyperarc from $(N_3 \rightarrow \bullet acid, 0, 0)$ to $(N_3 \rightarrow acid\bullet, 0, 1)$ has weight 0.5 since its rule stems from the 3^{rd} tree. The cost of a parse is then defined as the sum of weights of all traversed hyperarcs. Here, the hyperpath (highlighted in bold), corresponding to the idiomatic interpretation of *acid rains*, has cost 1, while the interpretation assuming that *rains* is a verb has cost 2. Thus, promoting MWE-oriented interpretations boils down to finding minimum-cost hyperpaths in the parsing hypergraph.

Recall that we also wish to find such interpretations earlier than compositional alternatives. We think that this problems could be solved by

[2]The proposals from the following section apply, though, also to the standard LTAG grammar format.

[3]For the sake of simplicity we only present initial trees and ignore auxiliary trees in this abstract. Our algorithm, however, does take auxiliary trees as well as the adjunction operation into account.

[4]Here, we do not present the conversion process in details. It includes, in fact, a compression stage based on common subtree sharing, and representing flat rules via a finite-state automaton.

an A*-style algorithm, similarly to (Lewis and Steedman 2014) for CCG parsing. The A* algorithm is based on a heuristic which estimates the distance that separates a given node from the target node. This distance estimation must never overestimate. We propose an estimation function h based precisely on the potential occurrence of MWEs in the part of the sentence that remains to be parsed. It assumes that each remaining word will be scanned with a grammar terminal containing the lowest possible weight, thus providing a lower bound on the remaining parsing cost. For example, the value of $h(N_0 \rightarrow acid\bullet, 0, 1)$ is 0.5 because the remaining part (assuming that *acid rains* is all that there is to parse), *rains*, cannot be scanned cheaper than 0.5. The total estimated cost of this state is thus equal to 1.5, therefore it will not be visited before state $(S \rightarrow NP \bullet VP_1, 0, 2)$ – which represents the optimal-cost interpretation of *acid rains* – is reached.

Note that the more terminals a grammar tree contains the lower the weights assigned to these terminals. Thus, this strategy truly promotes MWE-oriented interpretations.

Formally, remaining cost estimation for state (q, i, j) depends only on its span (i, j):

$h(q, i, j) = \sum_{k \in \{1,\ldots,i\} \cup \{j+1,\ldots,|s|\}} w(k)$

$w(k) = \min\{weight(r, l) : r \in \mathcal{F}(\mathcal{G}), l \in \{1, \ldots, |r|\}, r_k = s_l\}$

where s is the input sentence, s_i is its i-th word (starting from 1), $\mathcal{G}$ is a TAG, $\mathcal{F}(\mathcal{G})$ is $\mathcal{G}$ converted to the set of flat rules, $|r|$ is the length of r's body, r_i its i-th body element, and $weight(r, l)$ is the weight assigned to the l-th body element of r.

The perspectives of this work include proving the correctness of our MWE-based heuristics in A*, and providing experimental results of the parser. In the long run, weights assigned to grammar trees might be enhanced with probabilities acquired from a corpus, which would result in a probabilistic MWE-prone parser for LTAGs.

References

Abeillé, A. and Schabes, Y. (1989). Parsing idioms in lexicalized tags. In H. L. Somers and M. M. Wood, eds., *Proceedings of the 4th Conference of the European Chapter of the ACL, EACL'89, Manchester*, pp. 1–9. The Association for Computer Linguistics.

Alonso, M. A., Cabrero, D., de la Clergerie, E. V., and Ferro, M. V. (1999). Tabular algorithms for TAG parsing. In *EACL 1999, 9th Conference of the European Chapter of the Association for Computational Linguistics, June 8-12, 1999, University of Bergen, Bergen, Norway*, pp. 150–157. The Association for Computer Linguistics.

Constant, M., Roux, J. L., and Sigogne, A. (2013). Combining compound recognition and PCFG-LA parsing with word lattices and conditional random fields. *ACM Trans. Speech Lang. Process.*, **10**(3), 8:1–8:24.

Finkel, J. R. and Manning, C. D. (2009). Joint Parsing and Named Entity Recognition. In *HLT-NAACL*, pp. 326–334. The Association for Computational Linguistics.

Gallo, G., Longo, G., Pallottino, S., and Nguyen, S. (1993). Directed hypergraphs and applications. *Discrete Appl. Math.*, **42**(2-3), 177–201.

Green, S., de Marneffe, M.-C., Bauer, J., and Manning, C. D. (2011). Multiword Expression Identification with Tree Substitution Grammars: A Parsing tour de force with French. In *EMNLP*, pp. 725–735. ACL.

Green, S., de Marneffe, M.-C., and Manning, C. D. (2013). Parsing Models for Identifying Multiword Expressions. *Computational Linguistics*, **39**(1), 195–227.

Klein, D. and Manning, C. D. (2001). Parsing and hypergraphs. In *Proceedings of the Seventh International Workshop on Parsing Technologies (IWPT-2001), 17-19 October 2001, Beijing, China*. Tsinghua University Press.

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)*, pp. 990–1000. Association for Computational Linguistics.

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)*, pp. 27–35, Beijing, China. Association for Computational Linguistics.

Appendix A

Chart parsing of the substring *acid rains* represented as a hypergraph