Promoting multiword expressions in A* TAG parsing

Jakub Waszczuk and **Agata Savary**
Université François-Rabelais Tours,
3 place Jean-Jaurès,
41000 Blois, France
first.last@univ-tours.fr

Yannick Parmentier
LIFO - Université d'Orléans,
6, rue Léonard de Vinci,
45067 Orléans, France
first.last@univ-orleans.fr

Abstract

Multiword expressions (MWEs) are pervasive in natural languages and often have both idiomatic and compositional readings, which leads to high syntactic ambiguity. We show that for some MWE types idiomatic readings are usually the correct ones. We propose a heuristic for an A* parser for Tree Adjoining Grammars which benefits from this knowledge by promoting MWE-oriented analyses. This strategy leads to a substantial reduction in the parsing search space in case of true positive MWE occurrences, while avoiding parsing failures in case of false positives.

1 Introduction

Multiword expressions (MWEs), e.g. *by and large*, *red tape*, and *to pull one's socks up* 'to correct one's work or behavior', are linguistic objects containing two or more words and showing idiosyncratic behavior at different levels. Notably, their meaning is often not deducible from the meanings of their components and from their syntactic structure in a fully compositional way. Thus, interpretation-oriented NLP tasks, such as semantic calculus or translation, call for MWE-dedicated procedures. Syntactic parsing often underlies such tasks, and the crucial issue is at which point the MWE identification should take place: before (Nivre and Nilsson, 2004), after (Constant et al., 2012) or during parsing (Wehrli et al., 2010; Green et al., 2013; Candito and Constant, 2014; Nasr et al., 2015; Constant and Nivre, 2016). The last, joint, approach proves the most efficient due to at least two reasons. Firstly, some MWEs coincide with word combinations that cross phrase boundaries, which is hard to detect prior to parsing, as in example (1). Secondly, while most MWEs have both an idiomatic and a compositional reading, as in examples (2)–(3), the former occurs much more frequently than the latter for large classes of MWEs. In Sec. 6 we show that, indeed, the *idiomaticity rate*, i.e. the ratio of occurrences with idiomatic reading to all occurrences in a corpus, exceeds 0.95 for verbal MWEs and compounds. This suggests that promoting MWE-oriented analyses in parsing might lead to rapidly achieved correct parses. (Wehrli, 2014) shows that, indeed, the quality of symbolic parsing significantly increases if an occurrence of a MWE is admitted as soon as the necessary syntactic constraints are fulfilled. Our goal is to apply a similar strategy, i.e. to systematically promote MWE-oriented interpretations, while parsing with Tree Adjoining Grammars (TAGs).

(1) **After all** the preparations we finally left.
(2) After being criticized, she **pulled her socks up**.
(3) When the kid shivered with cold, **she pulled its socks up**.
(4) **Acid rains** in Ghana are equally grim.

Consider the sentence in example (4). At least two competing analyses are syntactically valid for the first 4 words: *rains* is (a) a verb with the subject *acid*, or (b) the head noun of a nominal phrase. In the latter case, the nominal phrase has either (i) a compositional reading (*acid* is a regular nominal modifier) or (ii) an idiomatic one (*acids rains* is an NN compound).

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers,
pages 429–439, Osaka, Japan, December 11-17 2016.

Our objective is to propose a parsing strategy which would promote analysis (b) and reading (ii). More precisely, the parser should only provide grammar-compliant MWE-oriented analyses each time they are feasible. Thus, we wish to both avoid the parsing failure for (1), and rapidly achieve the correct syntactic parses of (2)–(4), due to imposing their idiomatic interpretations. In this way, the parser's search space is reduced, with virtually no loss of correct parses, and with rare errors at the level of MWE identification, as in (3). The rate of such errors is the complement of the idiomaticity rate of the text to be parsed (here: 0.05).

Note that promoting the most probably correct analysis, whether containing MWEs or not, is the goal of probabilistic parsers in general. Thus, instead of designing a custom parsing architecture for promoting MWEs, it would be more adequate to simply train a general-purpose parser on a treebank containing MWE annotations. This solution is however hindered by data insufficiency. Firstly, many languages still lack large-size treebanks. Secondly, very few treebanks contain a full-fledged range of MWE annotations, even for English (Rosén et al., 2015). Thirdly, MWEs are subject to sparseness problems even more than single words: most existing MWEs occur never or rarely in MWE-annotated corpora (Czerepowicka and Savary, 2015), let alone treebanks. Here, we partly cope with these problems by an Earley-style A* parser using a MWE-oriented heuristic, which takes advantage of a potential occurrence of MWEs in a sentence. While it is designed to systematically promote MWEs regardless of their probabilities, the parser could be very well used with a weighted TAG and the weights assigned to individual elementary trees could be estimated on the basis of training data.

In Sec. 2 we remind basic facts about TAGs. In Sec. 3 we explain the MWE-promoting strategy in TAG parsing. In Sec. 4 we describe the parsing algorithm on a running example and we formalize its heuristics in Sec. 5. In Sec. 6 we show experimental results on a Polish TAG grammar extracted from a treebank. The choice of Polish is due to the fact that high-quality MWE resources compatible with the treebank are available for this language. In Sec. 7 we compare our approach with related work. Finally, we conclude and comment on future work.

2 Tree Adjoining Grammars

A TAG (Joshi et al., 1975) is a tree-rewriting system defined as a tuple $\langle \Sigma, N, I, A, S \rangle$, where Σ (resp. N) is a set of terminal (resp. non-terminal) symbols, I and A are sets of elementary trees (ETs), and $S \in N$ is the axiom. Trees in I are called initial trees (ITs), their internal and leaf nodes are labeled with symbols in N and in $\Sigma \cup N$, respectively. Their non-terminal leaf nodes are called substitution nodes and marked with $\downarrow$. Trees in A are called auxiliary trees (ATs) and are similar to trees in I except that they contain a leaf node (called a foot and marked with $\star$) whose label is the same as the one of the root. Consider the toy TAG in Fig. 2 covering three competing interpretations for *acid rains* in example (4). Notably, tree t_5 represents its idiomatic reading. We have $I = \{t_1, t_3, t_4, t_5, t_6\}$ and $A = \{t_2\}$.

ETs are combined to derive new trees using *substitution* and *adjunction*. Substitution consists in replacing a leaf with an ET whose root is labeled with the same non-terminal (cf. the dotted arrow in Fig. 1). Adjunction consists in inserting an AT t inside any tree t' provided that the root/foot label of t is the same as the label of the insertion point in t' (cf. the dashed arrows in Fig. 1). The result of a TAG derivation is twofold: a *derived tree*, and a *derivation tree*. The former represents the syntactic tree resulting from tree rewriting. The latter shows which ETs have been combined and how, as shown in Fig. 1(b). The derived tree of a sentence containing a syntactically regular MWE is identical to the one with its compositional reading, but their derivation trees differ. Thus, in the context of joint syntactic parsing and MWE identification (cf. Sec. 1), the derived and the derivation trees can be seen as the results of the former and of the latter task, respectively.

A TAG whose every ET contains at least one terminal leaf is called an LTAG (lexicalized TAG). The reason why we are particularly interested in LTAGs is that we consider MWEs a central challenge in NLP, and LTAGs show several advantages with respect to them (Abeillé and Schabes, 1989). Firstly, each MWE, together with the lexical and morphosyntactic constraints that it imposes, can be represented as a unique ET. Unification constraints on feature structures attached to tree nodes allow one to naturally express dependencies between arguments at different depths in the ETs (e.g. the subject-possessive agreement in *to pull one's socks up*). This is not the case for most other grammatical formalism, which handle long-distance dependencies by feature percolation. 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 (e.g. required but non-lexicalized arguments) are directly visible in elementary trees and are handled in parsing mostly by substitution. Discontinuities coming from insertion of adjuncts (e.g. *a bunch of NP*, *a whole bunch of NP*) are invisible in elementary trees but are handled by adjunction.

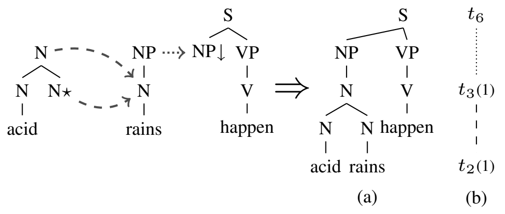


Figure 1: Tree rewriting in TAG resulting in a derived tree (a), and a derivation tree (b).

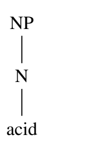
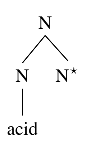
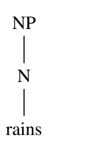
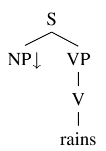
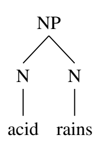
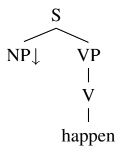
(t_1)	(t_2)	(t_3)	(t_4)	(t_5)	(t_6)
					
NP → N$_0$ N$_0$ → acid	N → N$_1$ N★ N$_1$ → acid	NP → N$_2$ N$_2$ → rains	S → NP VP$_3$ VP$_3$ → V$_4$ V$_4$ → rains	NP → N$_5$ N$_6$ N$_5$ → acid N$_6$ → rains	S → NP VP$_7$ VP$_7$ → V$_8$ V$_8$ → happen

Figure 2: A toy TAG grammar converted into flat rules

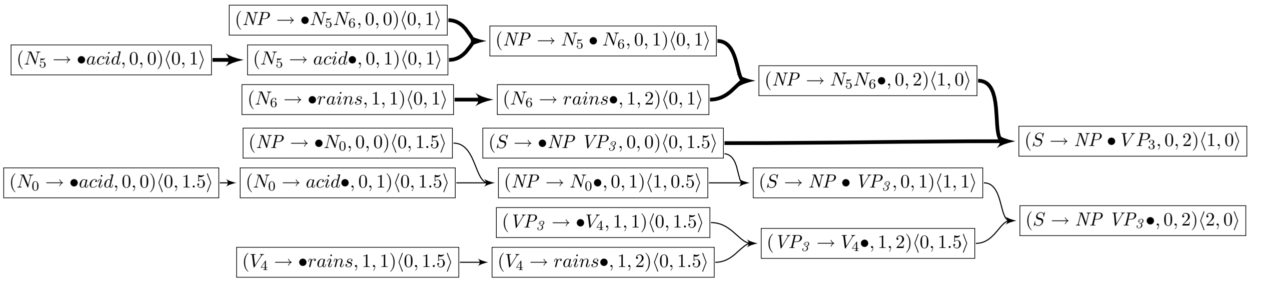

$(NP \to \bullet N_5 N_6, 0, 0)\langle 0, 1\rangle$
$(N_5 \to \bullet acid, 0, 0)\langle 0, 1\rangle$
$(N_5 \to acid\bullet, 0, 1)\langle 0, 1\rangle$
$(NP \to N_5 \bullet N_6, 0, 1)\langle 0, 1\rangle$
$(N_6 \to \bullet rains, 1, 1)\langle 0, 1\rangle$
$(N_6 \to rains\bullet, 1, 2)\langle 0, 1\rangle$
$(NP \to N_5 N_6\bullet, 0, 2)\langle 1, 0\rangle$
$(NP \to \bullet N_0, 0, 0)\langle 0, 1.5\rangle$
$(S \to \bullet NP\ VP_3, 0, 0)\langle 0, 1.5\rangle$
$(S \to NP \bullet VP_3, 0, 2)\langle 1, 0\rangle$
$(N_0 \to \bullet acid, 0, 0)\langle 0, 1.5\rangle$
$(N_0 \to acid\bullet, 0, 1)\langle 0, 1.5\rangle$
$(NP \to N_0\bullet, 0, 1)\langle 1, 0.5\rangle$
$(S \to NP \bullet VP_3, 0, 1)\langle 1, 1\rangle$
$(VP_3 \to \bullet V_4, 1, 1)\langle 0, 1.5\rangle$
$(VP_3 \to V_4\bullet, 1, 2)\langle 0, 1.5\rangle$
$(S \to NP\ VP_3\bullet, 0, 2)\langle 2, 0\rangle$
$(V_4 \to \bullet rains, 1, 1)\langle 0, 1.5\rangle$
$(V_4 \to rains\bullet, 1, 2)\langle 0, 1.5\rangle$

Figure 3: Hypergraph representing the chart parsing of the substring *acid rains* with ETs t_1, t_4 and t_5 from Fig. 2. The lowest-cost path representing the idiomatic interpretation is highlighted in bold.

3 Promoting MWEs in weighted TAG parsing

The fact that MWEs are represented in LTAGs as ETs allows us to propose a very simple and yet powerful strategy of promoting them in parsing. As seen in Sec. 2, parsing with an LTAG consists in combining ETs via substitution or adjunction. We define the weight of a full parse as the sum of the weights of the participating ETs. Note that the more sentence words belong to MWEs, and the longer are those MWEs, the less ETs are needed to cover the sentence. Suppose, for instance, that the sequence *acid rains* in Fig. 1 is covered by its idiomatic interpretation represented by tree t_5 from Fig. 2, instead of being handled by adjunction. In this case parsing *acid rains happen* produces the same derived tree as before but the derivation tree is smaller: it involves 2 ETs instead of 3.

This simple observation underlies our idea of promoting MWE-oriented analyses. Namely, suppose the input LTAG trivially weighted, i.e., each ET having weight 1. Then, finding analyses containing the maximum number of MWEs boils down to achieving the lowest-weight parses. Our objective is to find them more rapidly than other parses, which can be achieved by an A* algorithm using a MWE-driven heuristics, as described in the following sections. See also Sec. 8 for considerations on how this solution might generalize to non-trivially weighted grammars, notably with weights estimated on the basis of treebanks.

4 Weighted parsing with a flattened TAG

In (Waszczuk et al., 2016) we presented a TAG parsing architecture based notably on grammar flattening, subtree sharing and finite-state-based compression. Here, we sketch a simplified version of this architecture, and explain how it implements parsing as an A* graph traversal algorithm. Then in Sec. 5 we define the heuristic implementing the MWE promoting strategy, which – to the best of our knowledge – is totally novel.

Consider again the LTAG in Fig. 2. For the sake of presentation and compression (cf. Sec. 6), we represent TAG ETs as sets of flat production rules (Alonso et al., 1999) with indexed non-terminals.[1] For instance, the two N non-terminals in t_5 receive different indexes so as to avoid spurious analyses like $[[rains]_N[acid]_N]_{NP}$. A rule headed by the root of an ET (e.g., $S \rightarrow NP\ VP_3$) is called a *top rule*. The other rules are called *inside rules*.

Suppose that only the first two words of sentence (4) are to be parsed with a grammar subset limited to t_1, t_4 and t_5. With a flattened grammar representation, TAG parsing comes close to CFG parsing (even if dedicated inference rules are needed for adjunction, which is neglected in this paper). Like for CFG, an Earley-style parsing process for TAGs defined within a deductive framework (Shieber et al., 1995), involving an *agenda* (queue of weighted items) and a *chart*, can be represented as a hypergraph (Klein and Manning, 2001), more precisely a B-graph (Gallo et al., 1993), whose nodes are items of the chart and of the agenda, and whose hyperarcs represent applications of inference rules, as shown in Fig. 3. Each item $\mathcal{I} = (r, k, l)$ contains a dotted rule r and the span (k, l) over which the symbols to the left of the dot have been parsed.[2] For instance, the hyperarc leading from $(N_5 \rightarrow \bullet acid, 0, 0)$ to $(N_5 \rightarrow acid\bullet, 0, 1)$ means that the terminal $acid$ has been scanned from position 0 to 1. The latter item can then be combined with $(NP \rightarrow \bullet N_5 N_6, 0, 0)$ to yield $(NP \rightarrow N_5 \bullet N_6, 0, 1)$, etc. $\mathcal{I}$ and r are called *passive*, if the dot occurs at the end of r, and *active* otherwise. A sentence s has been parsed if a target item has been reached (spanning over the whole sentence, with a passive top rule headed by S).

The specificity of such a hypergraph lies in the fact that it is dynamically generated as the parsing process goes on. The main objectives include the generation of the smallest possible portion of this hypergraph, while including all the requested parses. In our case those are all optimal parses[3], in the sense of the MWE-promoting strategy.

Each derivation traversing $\mathcal{I} = (r, k, l)$ and resulting in a full parse tree T can be divided into two parts: (i) $\mathcal{I}$'s *inside derivation*, i.e., the part of the derivation corresponding to a (possibly partial) subtree of T rooted at r' head and spanning over (k, l), (ii) $\mathcal{I}$'s *outside derivation*, the part of the derivation corresponding to a partial tree obtained from T but excluding $\mathcal{I}$'s *inside derivation*. The weights of $\mathcal{I}$'s best inside and outside derivations are denoted by $\beta(\mathcal{I})$ and $\alpha(\mathcal{I})$. They are calculated according to the strategy described in Sec. 3, i.e. as numbers of ETs involved.

In symbolic CFG parsing, and in deductive parsing in general, the sentence parsability problem boils down to target node B-reachability in the (gradually constructed) hypergraph, and can be solved e.g. by a depth-first search generalized to hypegraphs. In probabilistic CFG parsing, parse trees and hypergraph B-paths are scored, and discovering the best parse is equivalent to finding the shortest B-path, which can be done by Dijkstra's algorithm generalized to hypergraphs (Gallo et al., 1993). The search space of this basic algorithm can be reduced in the A* algorithm (Klein and Manning, 2003), by introducing a heuristic which estimates the distance of each node to a target node. Namely, each $\mathcal{I}$ is assigned two values: $\beta(\mathcal{I})$ and $h(\mathcal{I})$, the latter being an estimation of $\alpha(\mathcal{I})$. The parsing items are popped from agenda in increasing order of $\beta(\mathcal{I}) + h(\mathcal{I})$. The heuristic used to calculate $h(\mathcal{I})$ should be *admissible*, i.e. should never overestimate ($h(\mathcal{I}) \leq \alpha(\mathcal{I})$). Additionally, if the heuristic is *monotonic* (i.e. $\beta(\mathcal{I}) + h(\mathcal{I})$ never increases), then an item is never re-introduced into the agenda once is has been popped, and the algorithm runs faster.

We apply the A* algorithm in a slightly adapted version in that we do not search for one but for all optimal parses, i.e. those containing grammar-compliant idiomatic interpretations. Thus, we do not quit when the first target item has been reached, but only when we are sure that no more optimal derivations can be found. As long as $\mathcal{I}$ stays on the agenda, $\beta(\mathcal{I})$ has to be recalculated each time a new hyperarc with head node $\mathcal{I}$ is added. Once $\mathcal{I}$ moves to the chart, $\beta(\mathcal{I})$ remains constant. In Fig. 3, the couple $\langle\beta(\mathcal{I}), h(\mathcal{I})\rangle$ decorates each node. Note that in case of parsing with a flattened TAG, only an ET t, not its

[1] Our proposal applies, however, to other LTAG representations as well.

[2] For simplicity, we ignore the fact that an item's span can include a gap accounting for adjunction.

[3] In probabilistic CFG parsing, the 1-best parse (Klein and Manning, 2003) or k-best parses (Pauls and Klein, 2009) are usually considered.

individual flat rules, is assigned a weight. Therefore, t's weight contributes to $\beta(\mathcal{I})$ only when t has been fully parsed, and it contributes to $h(\mathcal{I})$ otherwise. For instance, going from items $(NP \rightarrow N_5 \bullet N_6, 0, 1)$ and $(N_6 \rightarrow rains\bullet, 1, 2)$ to $(NP \rightarrow N_5 N_6\bullet, 0, 2)$ we have completed parsing the top rule of t_5, thus the weight of this ET (1) is added to W_1. However, item $(N_6 \rightarrow rains\bullet, 1, 2)$ is decorated with $\langle 0, 1\rangle$, since no ET has been fully parsed so far but we are parsing tree t_5 (with weight 1), whose terminals fully cover the intended span $(0, 2)$.

5 MWE-driven heuristic

The proper choice of the heuristic is crucial for the performance of the A* algorithm. We propose a heuristic $h(\mathcal{I})$ specifically designed to handle MWEs and, more generally, ETs with multiple anchors, which allows to use the A* parsing algorithm with MWE-aware weighted TAG grammars. In case weight 1 is assigned to all ETs, the heuristic closely models the strategy of promoting MWEs described in Sec. 3. Namely, it admits that if a given MWE has a chance to occur in the part of the sentence that remains to be parsed (i.e., in its outside derivation), then this MWE probably occurs. More precisely, the yet unparsed portion of the sentence can be divided into two parts: (i) the terminals yet to be covered by the tree that we are currently parsing, (ii) the remaining terminals. The heuristic consists in considering each terminal s_i from (ii) separately and assuming that it will be parsed with the ET containing s_i within the longest possible MWE.

Formally, let $\mathcal{S} = s_1 s_2 \ldots s_{|\mathcal{S}|}$ be the input sentence and $Pos(\mathcal{S})$ the set of positions between its words, ranging from 0 to $|\mathcal{S}|$. Since the same word can occur more than once in a sentence or a tree, we manipulate multisets of words. For a set X, a multiset over X is a set of pairs $\{(x, m(x)) : x \in X\}$, where $m(x) \in \mathbb{N}^+$ is called the multiplicity of x. We extend set notations and operators to multisets. For instance, $\{(a, 2), (b, 1)\}$ is noted as $\{a, a, b\}_{ms}$, and we have $\{a, b\}_{ms} \cup \{a\}_{ms} = \{a, a, b\}_{ms}$, $\{a, a, b\}_{ms} \setminus \{a, b\}_{ms} = \{a\}_{ms}$, $\{a, b\}_{ms} \subseteq \{a, a, b\}_{ms}$, $\{a, a, b\}_{ms} \nsubseteq \{a, b\}_{ms}$, $|\{a, a, b\}_{ms}| = 3$, etc. For any set X, let $\mathcal{M}(X)$ be the set of all multisets over X. Let $Rest(\mathcal{I})$ denote a multiset of words in the input sentence $\mathcal{S}$ outside of $\mathcal{I}$'s span, i.e., $Rest(\mathcal{I}) = \{s_1, \ldots, s_k, s_{l+1}, \ldots, s_{|\mathcal{S}|}\}_{ms}$.[4] Let $tree(r)$ be the ET from which r stems, and $W(t) \in [0, \infty)$ the weight of the ET t. For instance, in Fig. 2 and 3, for $r = N_5 \rightarrow acid\bullet$ we have $tree(r) = t_5$ and $W(t_i) = 1$ for $i = 1, \ldots, 6$.

Let $sub(t) \in \mathcal{M}(\Sigma)$ be the multiset of terminals in tree t. For instance, $sub(t_5) = \{acid, rains\}_{ms}$. For each word w, let $minw(w)$ denote the minimal weight of scanning w by an ET, i.e., the minimum proportion of w among all terminals of a single ET. More precisely,

$$minw(w) = \min_{t:(w,i) \in sub(t)} \frac{W(t)}{|sub(t)|}. \tag{5}$$

For instance, the proportion of *acid* in the terminals of t_1, t_2 and t_5 is, 1, 1 and 0.5, respectively, so $minw(acid) = 0.5$. Similarly $minw(rains) = 0.5$.[5] Thus, with all ET weights equal to 1, the longer a MWE, the lower are the $minw$ values of its components.

Let $sub(r), super(r) \in \mathcal{M}(\Sigma)$ be the multisets of terminals occurring in $tree(r)$ inside and outside of the subtree rooted at r's head, respectively. For instance, $sub(N_5 \rightarrow acid\bullet) = \{acid\}_{ms}$ and $super(N_5 \rightarrow acid\bullet) = \{rains\}_{ms}$. Note that for any top rule r, $super(r) = \emptyset_{ms}$.

Let $suff(r)$ be the set of passive non-top rules headed by the symbols in r' body after the dot. For instance, $suff(NP \rightarrow N_5 \bullet N_6) = \{N_6 \rightarrow rains\bullet\}$ and $suff(S \rightarrow \bullet NP\, VP_3) = \{VP_3 \rightarrow V_4\bullet\}$. Note that if r is passive, $suff(r) = \emptyset$.

Finally, let $Req(\mathcal{I})$ be the multiset of words required by the yet unparsed part of the current tree, i.e.,

$$Req(\mathcal{I}) = super(r) \cup \bigcup_{p \in suff(r)} sub(p). \tag{6}$$

[4] In case of adjunction $\mathcal{I}$'s span includes two additional indices denoting the gap, and the words within the gap also belong to $Rest(\mathcal{I})$.

[5] Variants of the $minw(w)$ definition include distributing the weights of individual terminals in an ET proportionally to their frequencies in the corpus. Our experiments did not show any advantage of such a distribution over the uniform one.

For instance in Fig.3, for item $\mathcal{I} = (NP \to N_5 \bullet N_6, 0, 1)$ we have $super(NP \to N_5 \bullet N_6) = \emptyset_{ms}$, $sub(N_6 \to rains\bullet) = \{rains\}_{ms}$, and $Req(\mathcal{I}) = \{rains\}_{ms}$.

For any item $\mathcal{I} = (r, k, l)$ we define a primary heuristic $h_0(\mathcal{I})$ as in equation (7).

$$h_0(\mathcal{I}) = \begin{cases} \infty, \text{ if } Req(\mathcal{I}) \not\subseteq Rest(\mathcal{I}) \\ \sum\limits_{(s,i) \in Rest(\mathcal{I}) \setminus Req(\mathcal{I})} minw(s) \times i, \text{ otherwise} \end{cases} \tag{7}$$

Then the estimation for the weight of $\mathcal{I}$'s best outside derivation, i.e. $\alpha(\mathcal{I})$, is given by equation (8).

$$h(\mathcal{I}) = \begin{cases} h_0(\mathcal{I}), \text{ if } \mathcal{I} \text{ is a top-rule passive item} \\ W(tree(r)) + h_0(\mathcal{I}), \text{ otherwise} \end{cases} \tag{8}$$

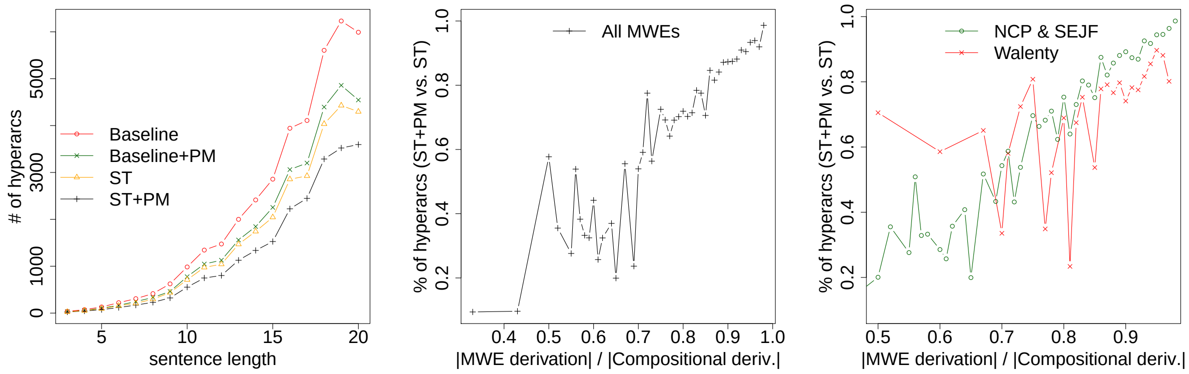
For instance, in the top-rule passive item $(NP \to N_0\bullet, 0, 1)$ we have finished parsing t_1 ($\beta(\mathcal{I}) = 1$) and we still have to consume *rains*, which implies a weight at least equal to $h(\mathcal{I}) = minw(rains) = 0.5$. In the inside-rule passive item $\mathcal{I} = (N_5 \to acid\bullet, 0, 1)$ we have $Rest(\mathcal{I}) = \{rains\}_{ms}$, $Req(N_5 \to acid\bullet) = \{rains\}_{ms}$, thus $h(\mathcal{I}) = W(t_5) = 1$. Finally, in the active item $\mathcal{I} = (NP \to N_5 \bullet N_6, 0, 1)$ we have $Rest(\mathcal{I}) = \{rains\}_{ms}$, $super(NP \to N_5 \bullet N_6) = \emptyset_{mt}$, and $Req(\mathcal{I}) = \{rains\}_{ms}$, thus $h(\mathcal{I}) = W(t_5) = 1$.

With this heuristic, and weight 1 assigned to individual ETs, the derivations containing MWEs are often reached before the paths towards compositional ones are even followed. For instance the item $(N_0 \to acid\bullet, 0, 1)$ has the estimated cost 1.5, and it will be created later than $(S \to NP\,VP_3\bullet, 0, 2)$. Thus, the hyperpath (highlighted in bold) assuming the idiomatic reading of *acid rains*, will be followed before the path assuming that *rains* is a verb.

For a given item the heuristic assumes that each remaining word w from the input sentence (with the exception of the words required by the rule underlying the item) will be scanned with the lowest possible cost, i.e. $minw(w)$ – see Eq. (5). The heuristic never over-estimates the cost of parsing the remaining part of the sentence and is thus *admissible*. All but one inference rules of the parser are also *monotonic*, in the sense that the estimation, stemming from the application of an inference rule, of the total weight $\beta(\mathcal{I}) + h(\mathcal{I})$ of an item $\mathcal{I}$ is greater or equal to the total weight, $\beta(\mathcal{I}') + h(\mathcal{I}')$, of any premise item $\mathcal{I}'$ of this rule. The sole exception concerns the inference rule – called *foot adjoin (FA)*, see (Waszczuk et al., 2016) – responsible for recognizing the so-called *gaps* over which adjoining could be performed. This is related to the fact that the weight of the item inferred with FA does not depend on the $\beta(\mathcal{I}')$ weight of its premise item $\mathcal{I}' = (r, k, l)$, where item $\mathcal{I}'$ provides an evidence that adjunction could possibly take place over span (k, l). Nonetheless, the algorithm guarantees that when item $\mathcal{I}$ is popped from the agenda, one of the hyperarcs representing an optimal derivation of $\mathcal{I}$ is already inferred, and thus the $\beta(\mathcal{I})$ value is correctly calculated.

6 Experimental results

We evaluated our parsing strategy with Składnica, a Polish treebank with over 9,000 manually disambiguated constituency trees (Świdziński and Woliński, 2010). As it contains no MWE annotations, we produced them automatically, by projecting 3 existing MWE resources: (i) the named entity (NE) layer of the National Corpus of Polish (NCP) (Savary et al., 2010) (only the multiword NEs were taken into account), (ii) SEJF, an extensional lexicon of Polish nominal, adjectival and adverbial MWEs (Czerepowicka and Savary, 2015), (iii) Walenty, a Polish valence dictionary (Przepiórkowski et al., 2014) with over 8,000 verbal MWEs. The mapping for (i) was straightforward and did not require manual validation, since Składnica is a subcorpus of the NCP, whose NE annotation and adjudication were performed manually. The mapping for (ii) and (iii), followed by a manual validation, consisted in searching for syntactic nodes satisfying all lexical constraints and part of syntactic constraints of a MWE entry. The required lexical nodes were to be contiguous for (ii) but not for (iii). As a result, 2026 idiomatic occurrences (1303 from NCP-NE, 368 from SEJF and 355 from Walenty) and 40 compositional ones (22 for SEJF and 18 for Walenty) were identified, which implies the idiomaticity rate about 0.95 (0.95 for Walenty and 0.94 for SEJF).



(a)

(b)

(c)

Figure 4: (a) Average number of hyperarcs explored depending on the parsing strategy (for clarity using only sentence of length < 20), (b) Average % of hyperarcs explored with the PM+ST strategy, using the ST strategy as a reference, and (c) Average % of hyperarcs explored depending on the type of MWEs.

A TAG grammar with 28652 lexicalized elementary trees was then extracted from the MWE-marked treebank, similarly to (Krasnowska, 2013) or (Chen and Shanker, 2004). Each treebank subtree marked for a MWE yielded: (i) a MWE-dedicated ET containing all paths leading to the lexical (co-)anchors, (ii) ETs covering the compositional interpretations. Various compression techniques can be applied to a flattened TAG (Waszczuk et al., 2016). We used a representation in which common subtrees and prefixes of flat rules are shared.

We assess our parser's efficiency in terms of the size of its parsing hypergraph. We believe it to be a more objective measure to compare different parsing strategies than the absolute parsing time, since each hypergraph edge corresponds to an application of an inference rule, i.e. to a basic parsing step (as in theoretical complexity considerations).[6] Conversely, the parsing time is highly dependent on the low level implementation details.[7]

The baseline hypergraph is the one generated with the full grammar, when no MWE-promoting strategy is used and all grammar-compliant parses are generated for each sentence. The MWE-promoting (PM) hypergraph, compared to this baseline, includes mainly the optimal parses (the algorithm ensures that, in PM, all optimal parses are achieved, but some sub-optimal parses may also be reached, since heuristic h is an imperfect estimation of α), i.e. those in which the maximum number of words belongs to potential MWEs.

The experiment was carried out on the same dataset from which the grammar was extracted. Therefore, for each sentence, the baseline hypergraph contained both its gold (i.e., conforming to Składnica) parse (derived tree) and its gold MWE identification (derivation tree). The PM hypergraphs, in turn, contained the correct parses for virtually 100% of the sentences,[8] and correct MWE identification for around 95% of them (due to the idiomaticity rate equal to 0.95). Thus, the parsing efficiency gain due to the PM strategy occurred with no loss of accuracy.

The PM strategy is comparable to supertagging (ST), i.e. pre-selecting, for each sentence, a subset of ETs which have good chances to be used in the derivation, in order to reduce the parsing search space. We experimented with a simple form of ST, which restricts the grammar to ETs whose terminals occur in the given sentence. Namely, we examined the ST hypergraph containing all parses for each sentence, and the one when ST was combined with PM (where mainly optimal parses were achieved).

Fig. 4a shows the absolute sizes of the hypergraphs for these 4 strategies in function of the sentence

[6] The overhead related to computing the values of the heuristic is at most linear in the size of the sentence, and may be much lower with efficient low-level optimizations.

[7] In an optimized implementation, TAG parsing time is proportional to the number of hyperarcs, as reported by (Waszczuk et al., 2016).

[8] A sanity check showed that for 54 sentences the gold parse was not found, mainly due to some abbreviation- and letter-case-related specificities, as well as to missing MWE annotations in Składnica.

length. The PM strategy brings enhancement regardless of whether supertagging is used or not. The supertagging alone outperforms, on average, the baseline MWEs-promoting strategy. Since the combination of ST and PM strategies proves the most efficient, we restrict further experiments to this version.

Note that Fig. 4a does not fully reflect the potential advantages of the PM strategy, whose behavior does not directly depend on the length of the parsed sentence, but rather on the number and the size of the MWEs potentially occurring in it. These 2 values can be together represented as the ratio of the size of the MWE-based derivation tree to the size of the corresponding compositional derivation tree (i.e. the one assuming no MWE occurrence). Expectedly, as shown in Fig. 4b, the lower this ratio (i.e. the more words in the sentence belong to MWEs, and the longer are these MWEs), the more significant the hypergraph size reductions. Moreover, the resulting graph suggests that the hypergraph size reductions are linear with respect to this ratio. Note that the vertical axis now shows the proportional gain in the hypergraph size due to the ST+PM strategy with respect to the ST strategy alone.

Finally, we investigated the behavior of the PM+ST strategy for two types of MWEs independently: verbal MWEs from Walenty and compounds from NCP and SEJF. As shown in Fig. 4c, verbal MWEs, while less frequent, prove to be better in reducing ambiguity for sentences with low number of potential MWEs. It is hard to ascertain this claim for sentences with lower gold derivation size ratio. While compounds seem to outperform verbal MWEs in this case, sentences with verbal MWEs for which this ratio is low are also very short in our dataset (of length 5, on average, for the 20 sentences with the lowest ratio), and thus exhibit low syntactic ambiguity.

7 Related work

While A* algorithms have been widely used for AI inference problems where a lightest derivation is to be found (Felzenszwalb and McAllester, 2007), this is to our knowledge the first attempt at using them within the context of MWE parsing with TAG. This work was inspired by Lewis and Steedman (2014) who applied A* to parsing with another strongly lexicalized grammar formalism, namely CCG. Unlike in this work, our grammar rules are not constrained to have a single lexical item, hence they can explicitly represent MWEs. This calls for a more elaborate heuristic, since a not yet parsed terminal can either be consumable by the currently parsed tree or not, as is the case with *rains* in item $(NP \rightarrow N_5 \bullet N_6, 0, 1)$ as opposed to $(NP \rightarrow N_0 \bullet, 0, 1)$ in Fig. 3. Distinguishing these two cases leads to a more precise weight estimation.

Angelov and Ljunglöf (2014) proposed to apply A* top-down parsing to parallel multiple context-free grammars, a formalism strictly more expressive than TAGs. In their approach weights are assigned to production rules and the grammar is not assumed to be strongly lexicalized, which complicates the design of an efficient heuristic. Their evaluation showed that a non-admissible heuristic can be orders of magnitude faster than the admissible version, at the expense of parsing quality.

Other ways of dealing with MWEs in the context of TAG would involve pre- or post-processing. A post-processing step would consist in identifying MWE interpretations in derivation structures (potentially with an additional processing cost). Regarding pre-processing, current state-of-the-art techniques are related to probabilistic supertagging (Bangalore and Joshi, 1999), as opposed to the simple symbolic supertagging applied in Sec.6. While labeling the words of a sentence with candidate ETs, one may either keep for each word the most probable ET, or all ETs whose probabilities are above a given threshold. Large MWE annotations are needed to train such supertaggers. Probabilistic treatment of contiguous MWEs has been applied to Tree-Substitution Grammar with encouraging results (Green et al., 2013). The main drawback of such probabilistic pre-processing is the fact that it can prevent the parser from finding the right derivations in case when the supertagging was wrong. This situation is avoided in A* parsing which, while requiring that candidate ETs be annotated with the corresponding probabilities, performs a filtering of unlike ET candidates on the fly.

An alternative to probabilistic supertagging has been proposed by Boullier (2003). There, an approximated CFG grammar is computed from an input TAG, and used to parse the input sentence so as to decide which ETs should be selected for TAG parsing. This approach has been enhanced by Gardent et al. (2014) to take word order into account. We consider such a supertagging technique an interesting candidate for future work. One could indeed not only select ETs that are compatible with the sentence to parse but also distinguish ETs for literal interpretations from ETs for MWEs. Like non-statistical supertagging, using an A* algorithm has the advantage to process MWEs while keeping ambiguity as long as possible to avoid dismissing valid interpretations.

Relatively few works have explicitly addressed the idiomaticity rate of MWEs. (Savary et al., 2012) perform a straightforward matching of a Polish economic MWE lexicon, containing extensional descriptions of morpho-syntactic variants, against a corpus and obtain only 0.12%–0.21% of false positives. (El Maarouf and Oakes, 2015) examine 10 verbal MWEs in the British National Corpus and find out that the idiomaticity measure for half of them exceeds 0.95, and for 9 most frequent of them is above 0.676.

8 Conclusions and future work

We have presented a novel LTAG parsing architecture in which parses potentially containing MWEs are given higher priorities so as to be achieved faster than the competing compositional analyses. The underlying A* algorithm uses a distance estimation heuristic based on the number of terminal nodes in elementary trees. The results obtained with a Polish TAG grammar show that this strategy can considerably reduce the number of parsing items to be explored in order to generate a subset of parses very likely to contain the correct parse. The tests used a grammar extracted from a MWE-annotated treebank but the method also applies to hand-crafted grammars.

Future work includes possible enhancements of the A* heuristic. It currently does not require that, if an ET is used to scan an input terminal, then all the other terminals of this ET also have to be present. It does not require either that the terminals need to be scanned in the appropriate order. Taking such constraints into account might enhance both the parsing quality and speed. Note also that the heuristic ignores ETs which contain no lexical anchors, so it is mainly adapted to strongly lexicalized TAGs. Relieving this constraint, while preserving MWE promotion, would be worth consideration.

Another perspective is to evaluate the computational overhead of the MWE-based heuristic, as opposed to identifying MWEs in a post-parsing step. Also, a fine-grained estimation of the idiomaticity rate of different types of MWEs might give us hints as to which of them should best be identified before, during or after parsing. With such data at hand, it should be possible to construct a multi-stage MWE-aware parsing architecture, tunable for optimum trade-off between accuracy and speed.

Even with MWE lexicon mapping on a treebank, as shown in Sec. 6, sufficiently large MWE-annotated treebanks are hard to obtain, and if they do exist, they are still concerned by MWE sparseness. In the long run, we aim at a hybrid parsing architecture in which a MWE-driven parser is fed with a probabilistic TAG grammar combined with MWE lexicons. We believe that such an extension of our solution to a hybrid setting is possible due to two factors. Firstly, the heuristic described in Sec. 5 generalizes to any weighted TAG with non-negative weights assigned to individual ETs. Secondly, systematically promoting MWE-oriented analyses in probabilistic parsing can be achieved even if MWEs are underrepresented in the training corpus. Namely, MWE-oriented ETs could stem from a syntactic MWE lexicon, such as Walenty (Przepiórkowski et al., 2014), while their weights could be calculated from the weights of the ETs corresponding to their compositional analyses. Alternatively, the weights could be represented as lexicographically ordered pairs, consisting of (i) the number of ETs participating in the underlying derivations, and (ii) the actual weights stemming from the weighted grammar.

Finally, integrating feature structures and unification within this parsing framework might lead to faster pruning of spurious analyses, and enable a more precise MWE identification, especially for inflectionally rich languages like Polish.

Acknowledgments

This work has been supported by the European Framework Programme Horizon 2020 via the PARSEME[9] European COST Action (IC1207), as well as by the French National Research Agency (ANR) via the PARSEME-FR[10] project (ANR-14-CERA-0001).

[9] www.parseme.eu

[10] http://parsemefr.lif.univ-mrs.fr

References

Anne Abeillé and Yves Schabes. 1989. Parsing idioms in lexicalized tags. In Harold L. Somers and Mary McGee Wood, editors, *Proceedings of the 4th Conference of the European Chapter of the ACL, EACL'89, Manchester*, pages 1–9. The Association for Computer Linguistics.

Miguel Alonso, David Cabrero, Eric Villemonte de la Clergerie, and Manuel Vilares Ferro. 1999. Tabular algorithms for TAG parsing. In *EACL 1999*, pages 150–157.

Krasimir Angelov and Peter Ljunglöf. 2014. Fast statistical parsing with parallel multiple context-free grammars. In *EACL*, volume 14, pages 368–376.

Srinivas Bangalore and Aravind K. Joshi. 1999. Supertagging: An Approach to Almost Parsing. *Comput. Linguist.*, 25(2):237–265, June.

Pierre Boullier. 2003. Supertagging: a Non-Statistical Parsing-Based Approach. In *Proceedings of the 8th International Workshop on Parsing Technologies (IWPT 03)*, pages 55–65, Nancy, France.

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, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 743–753.

John Chen and Vijay K. Shanker. 2004. New developments in parsing technology. chapter Automated Extraction of Tags from the Penn Treebank, pages 73–89. Kluwer Academic Publishers, Norwell, MA, USA.

Matthieu Constant and Joakim Nivre. 2016. A transition-based system for joint lexical and syntactic analysis. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 161–171, Berlin, Germany, August. Association for Computational Linguistics.

Matthieu Constant, Anthony Sigogne, and Patrick Watrin. 2012. Discriminative strategies to integrate multiword expression recognition and parsing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 204–212, Stroudsburg, PA, USA. Association for Computational Linguistics.

Monika Czerepowicka and Agata Savary. 2015. SEJF - a Grammatical Lexicon of Polish Multi-Word Expression. In *Proceedings of Language and Technology Conference (LTC'15), Poznań, Poland*. Wydawnictwo Poznańskie.

Ismail El Maarouf and Michael Oakes. 2015. Statistical Measures for Characterising MWEs. In *IC1207 COST PARSEME 5th general meeting*.

Pedro Felzenszwalb and David McAllester. 2007. The generalized A* architecture. *Journal of Artificial Intelligence Research*, 29:153–190.

Giorgio Gallo, Giustino Longo, Stefano Pallottino, and Sang Nguyen. 1993. Directed hypergraphs and applications. *Discrete Appl. Math.*, 42(2-3):177–201, April.

Claire Gardent, Yannick Parmentier, Guy Perrier, and Sylvain Schmitz. 2014. Lexical Disambiguation in LTAG using Left Context. In Zygmunt Vetulani and Joseph Mariani, 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.

Spence Green, Marie-Catherine de Marneffe, and Christopher D. Manning. 2013. Parsing Models for Identifying Multiword Expressions. *Computational Linguistics*, 39(1):195–227.

Aravind K. Joshi, Leon S. Levy, and Masako Takahashi. 1975. Tree adjunct grammars. *Journal of the Computer and System Sciences*, 10:136–163.

Dan Klein and Christopher D. Manning. 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.

Dan Klein and Christopher D. Manning. 2003. A* parsing: Fast exact viterbi parse selection. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*.

Katarzyna Krasnowska. 2013. Towards a polish LTAG grammar. In Mieczyslaw A. Klopotek, Jacek Koronacki, Malgorzata Marciniak, Agnieszka Mykowiecka, and Slawomir T. Wierzchon, editors, *Language Processing and Intelligent Information Systems - 20th International Conference, IIS 2013, Warsaw, Poland, June 17-18, 2013. Proceedings*, volume 7912 of *Lecture Notes in Computer Science*, pages 16–21. Springer.

Mike Lewis and Mark Steedman. 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.

Alexis Nasr, Carlos Ramisch, José Deulofeu, and Valli André. 2015. Joint Dependency Parsing and Multiword Expression Tokenisation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL'15)*.

Joakim Nivre and Jens Nilsson. 2004. Multiword units in syntactic parsing. In *Proceedings of MEMURA 2004 – Methodologies and Evaluation of Multiword Units in Real-World Applications, Workshop at LREC 2004, May 25, 2004, Lisbon, Portugal*, pages 39–46, Lisbon, Portugal, May.

Adam Pauls and Dan Klein. 2009. K-best a* parsing. In Keh-Yih Su, Jian Su, and Janyce Wiebe, editors, *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*, pages 958–966. The Association for Computer Linguistics.

Adam Przepiórkowski, Elżbieta Hajnicz, Agnieszka Patejuk, and Marcin Woliński. 2014. Extended phraseological information in a valence dictionary for NLP applications. In *Proceedings of the Workshop on Lexical and Grammatical Resources for Language Processing (LG-LP 2014)*, pages 83–91, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Victoria Rosén, Gyri Smørdal Losnegaard, Koenraad De Smedt, Eduard Bejček, Agata Savary, Adam Przepiórkowski, Petya Osenova, and Verginica Barbu Mitetelu. 2015. A survey of multiword expressions in treebanks. In *Proceedings of the 14th International Workshop on Treebanks & Linguistic Theories conference*, Warsaw, Poland, December.

Agata Savary, Jakub Waszczuk, and Adam Przepiórkowski. 2010. Towards the annotation of named entities in the national corpus of polish. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may. European Language Resources Association (ELRA).

Agata Savary, Bartosz Zaborowski, Aleksandra Krawczyk-Wieczorek, and Filip Makowiecki. 2012. SEJFEK - a Lexicon and a Shallow Grammar of Polish Economic Multi-Word Units. In *Proceedings of the 3rd Workshop on Cognitive Aspects of the Lexicon*, pages 195–214, Mumbai, India, December. The COLING 2012 Organizing Committee.

Stuart M Shieber, Yves Schabes, and Fernando CN Pereira. 1995. Principles and implementation of deductive parsing. *The Journal of logic programming*, 24(1):3–36.

Jakub Waszczuk, Agata Savary, and Yannick Parmentier. 2016. 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, July.

Eric Wehrli, Violeta Seretan, and Luka Nerima. 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, August. Association for Computational Linguistics.

Eric Wehrli. 2014. The relevance of collocations for parsing. In *Proceedings of the 10th Workshop on Multiword Expressions (MWE)*, pages 26–32, Gothenburg, Sweden, April. Association for Computational Linguistics.

Marek Świdziński and Marcin Woliński. 2010. Towards a bank of constituent parse trees for Polish. In Petr Sojka, Aleš Horák, Ivan Kopeček, and Karel Pala, 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.